FG-OrIU: Towards Better Forgetting via Feature-Gradient Orthogonality for Incremental Unlearning

Qian Feng[1], **JiaHang Tu**[1], **Mintong Kang**[2], **Hanbin Zhao**[1]*, **Chao Zhang**[1], **Hui Qian**[1]
[1]College of Computer Science and Technology, Zhejiang University
[2]Department of Computer Science University of Illinois at Urbana-Champaign

Abstract

*Incremental unlearning (IU) is critical for pre-trained models to comply with sequential data deletion requests, yet existing methods primarily suppress parameters or confuse knowledge without explicit constraints on both feature and gradient level, resulting in superficial forgetting where residual information remains recoverable. This incomplete forgetting risks security breaches and disrupts retention balance, especially in IU scenarios. We propose FG-OrIU (**F**eature-**G**radient **Or**thogonality for **I**ncremental **U**nlearning), the first framework unifying orthogonal constraints on both features and gradients level to achieve deep forgetting, where the forgetting effect is irreversible. FG-OrIU decomposes feature spaces via Singular Value Decomposition (SVD), separating forgetting and remaining class features into distinct subspaces. It then enforces dual constraints: feature orthogonal projection on both forgetting and remaining classes, while gradient orthogonal projection prevents the reintroduction of forgotten knowledge and disruption to remaining classes during updates. Additionally, dynamic subspace adaptation merges newly forgetting subspaces and contracts remaining subspaces, ensuring a stable balance between removal and retention across sequential unlearning tasks. Extensive experiments demonstrate the effectiveness of our method.*

1. Introduction

In recent years, *pre-training with fine-tuning* [29, 73, 80] has dominated model development. However, large-scale datasets often contain harmful or private content, necessitating mechanisms to erase specific data traces. Machine Unlearning (MU) [4, 5, 7, 12, 22, 27, 31, 43, 52, 53, 72] addresses this by enabling pre-trained models (PTMs) to comply with deletion mandates like GDPR (General data protection regulation)'s "right to be forgotten" [46, 48]. While existing MU methods focus on static unlearning, forgetting a single or small set of classes in one task, real-world scenarios require Incremental Unlearning (IU) [8–10, 24, 26, 32, 35, 38, 44, 61, 83, 84]. Regulatory shifts (e.g., evolving OpenAI safety policies[1], EU AI Act[2]) and dynamic content restrictions necessitate sequential unlearning of multiple classes/concepts without costly retraining. IU need not only erase targets in a single step but also preserve retained performance and ensure irreversible forgetting across tasks, making it critical for sustainable model governance in evolving ethical and legal landscapes.

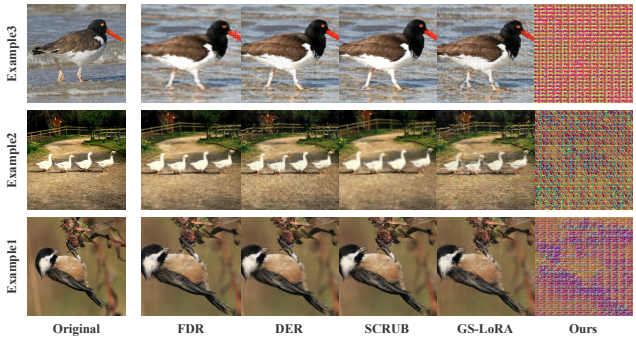


Figure 1. Here, we present three examples from forgetting classes, with the reconstructed images generated using Deep Image Prior (DIP) [59], where the last block's features from the unlearned model serve as the training target.

IU methods broadly address two scenarios: (1) concept erasure in generative models, where sensitive concepts (e.g., nudity, violence) are iteratively removed from diffusion models [11, 56], and (2) instant- or class-level unlearning in vision classifiers, where PTMs forget specific classes (e.g., face identity, medical data) while retaining unrelated ones [33, 43]. In this work, we focus on the latter, specifically targeting class-level IU in vision PTMs. Our goal is threefold: (a) to ensure that the model completely forgets the specified classes, (b) to robustly preserve the performance of remaining classes, and (c) to maintain the effectiveness of previous forgetting across sequential unlearning procedures.

*Corresponding author: Hanbin Zhao <zhaohanbin@zju.edu.cn>.

[1]https://openai.com/safety/
[2]https://artificialintelligenceact.eu/the-act/

Existing methods largely fall into two categories: *parameter suppression*, which selectively weakens target class-related parameters [18, 22, 74], and *knowledge confusion*, which distorts predictions via noisy labels or bad teacher distillation [12, 25, 36]. However, our empirical findings reveal that these methods lead to *superficial forgetting*, as the unlearned model still retains semantic information related to the forgetting classes, making the erased knowledge partially residual and recoverable. Specifically, reconstructing features from the last layer of the unlearned model using Deep Image Prior (DIP) [59] produces blurred yet recognizable images (as illustrated in Figure 1). Furthermore, fixing the backbone of the unlearned model and retraining only the classification head [10, 84], which recovers a portion of the original accuracy (as illustrated in Figure 3). The blurred image and the partial recovery of performance both indicate that **existing methods merely degrade the features of forgetting classes rather than completely eliminating them.** The degraded features still contain residual class-specific information, which leads to potential recovery of the forgetting effect. Besides, these degraded features not only enable recovery but also risk entanglement with the remaining class features [43]. Such entanglement destabilizes decision boundaries, degrading performance on the remaining classes, particularly in IU scenarios where sequential deletion requests amplify error propagation [24, 61]. To address these limitations and achieve *deep forgetting*, we argue that unlearning should explicitly constrain both feature representations and gradient updates. Since model updates rely on both forward computations (feature extraction) and backward optimization (gradient updates), it is critical to simultaneously suppress the features of forgetting classes and prevent their reintroduction during training updates. In Table 1, we summarize the differences between our methods and prior works.

To enforce explicit constraints on both feature representations and gradient updates, we draw inspiration from orthogonal mechanisms in related domains. For feature-level constraints, recent work in text-to-image models [56, 78] demonstrates that orthogonalizing sensitive token embeddings effectively removes targeted concepts from diffusion outputs, suggesting geometric isolation as a viable strategy for feature suppression. For gradient-level constraints, inspired by [79], which reduces the mutual influence of different tasks in multi-task learning, and [51, 86], which mitigates the forgetting of old tasks in continual learning, we leverage the insight that updating in directions orthogonal to old/other task subspaces helps alleviate catastrophic forgetting. While these orthogonal principles show promise in adjacent fields like generative model editing and multi-task/continual learning, their applicability to MU, particularly IU in vision PTMs, remains underexplored.

In this paper, we propose FG-OrIU (**F**eature-**G**radient **Or**thogonality for **I**ncremental **U**nlearning), a unified framework for IU in vision PTMs that enforces explicit constraints at both feature and gradient levels through three stages: **feature subspace decomposition, dual orthogonal projection, and dynamic subspace adaptation**. In **feature subspace decomposition**, we decompose the feature space of each layer into two subspaces via Singular Value Decomposition (SVD): the forgetting subspace $\mathcal{S}_f$, which captures the discriminative features of the classes to be removed, and the remaining subspace $\mathcal{S}_r$, which encodes the features of the remaining classes. In **dual orthogonal projection**, we freeze the backbone of PTM and introduce lightweight LoRA modules for efficient updates. Then, features of samples from forgetting classes are orthogonalized against $\mathcal{S}_f$ to eliminate correlations, while remaining classes features are aligned with $\mathcal{S}_r$ to minimize interference through minimizing the projection residual. Before backward optimization, gradients are first projected onto $\mathcal{S}_f$ to maximally perturb parameters critical for forgetting classes, then modified along the orthogonal complement of $\mathcal{S}_r$ to prevent contamination of retained features. In **dynamic subspace adaptation**, FOrIU dynamically expands $\mathcal{S}_f$ by merging new forgetting class subspaces and contracts $\mathcal{S}_r$ by removing overlapping features, ensuring cumulative forgetting without degrading retained performance.

Table 1. *Position, Complete, and Recover* refers to, where the forgetting occurs, whether the forgetting is thorough, and whether the forgetting effect can be restored, respectively.

	Position	*Complete*	*Recover*
superficial forgetting			
Prior works	Only Final Output	No	Partial
deep forgetting			
Ours	Feature&Gradient	Yes	No

Our main contributions are threefold:

- We reveal that existing IU methods achieve *superficial forgetting* rather than *deep forgetting*, as their partial feature suppression leaves residual information recoverable.
- We propose the first framework unifying forgetting on both feature and gradient level via dual orthogonal projection, ensuring irreversible forgetting while preserving remaining class robustness.
- Extensive experiments demonstrate superior performance in MU and IU scenarios across multiple benchmarks.

2. Related Work

2.1. Machine Unlearning

Machine unlearning (MU) aims to remove the influence of specific data from pre-trained models (PTMs) without retraining from scratch [71, 81]. Existing MU methods are broadly categorized into exact and approximate unlearning. Exact MU [5, 39] ensures statistical indistinguishability from models retrained on remaining data but is computationally prohibitive for large PTMs. Approximate MU re-

laxes this guarantee and is more practical. Current methods fall into two paradigms: (1) concept erasure in generative models [11, 56]; and (2) class-/sample-wise unlearning in classification models, which modifies pre-trained weights to forget target classes while preserving others [33, 43]. However, most studies focus on static unlearning, assuming a single-step forgetting task. Real-world applications, such as compliance with iterative AI safety policies (e.g., OpenAI's content guidelines) or evolving regulations like the EU AI Act, require incremental unlearning (IU) to process sequential deletion requests efficiently. Directly applying static methods to IU often leads to cumulative errors, model collapse, and incomplete forgetting [24, 61], underscoring the need for dedicated frameworks to ensure compliance in dynamic regulatory environments.

2.2. Incremental Unlearning

Existing MU methods, like *parameter suppression* [18, 22, 74] and *knowledge confusion* methods [10, 12, 24, 25, 36, 44, 84], lead to *superficial forgetting* due to the absence of direct constraints on the feature representation. In particular, the degraded features trap them in a dilemma between forgetting and remaining, which becomes even more severe when directly applied to IU. Meanwhile, some continual learning methods [6, 34], originally designed to retain old knowledge while dynamically learning new knowledge, also face inherent limitations in IU [44]. Previous work focusing on IU has primarily relied on toy models, e.g., linear models [14], or imposed heavy constraints, such as adding a class-specific synthetic signal in the pre-training stage [54], which is not feasible for PTMs. Moreover, while some focus on dynamically performing both continual learning and selective forgetting within learned knowledge [8, 35, 38], our approach differs in that we emphasize achieving *deep forgetting* when handling sequential unlearning tasks. That is, rather than alternating between learning and forgetting, the pre-trained model should completely forget unwanted knowledge while efficiently retaining the rest.

3. Problem Statement

Following [83, 84], we first formally define Machine Unlearning (MU), *i.e.*, static unlearning scenario. We then extend this definition to Incremental Unlearning (IU), *i.e.*, dynamic unlearning scenario, where deletion requests arrive sequentially.

Let $\mathcal{M}$ be a pre-trained model trained on the dataset $\mathbf{D}$. We denote the mapping relationship of the model as $f_{\mathcal{M}}(\mathcal{X}_{\mathbf{D}}) \to \mathcal{Y}_{\mathbf{D}}$, where $\mathcal{X}_{\mathbf{D}}$ and $\mathcal{Y}_{\mathbf{D}}$ represent the input set and output set, respectively. For MU, we partition $\mathbf{D}$ into $\mathbf{D_f}$ (forgetting classes) and $\mathbf{D_r}$ (remaining classes), under the constraint $|\mathbf{D_r}|+|\mathbf{D_f}| \ll |\mathbf{D}|$. The forgetting algorithm $\mathscr{F}$ produces $\mathcal{M}' = \mathscr{F}(\mathcal{M}, \mathbf{D_f}, \mathbf{D_r})$, altering the following mappings:

$$f_{\mathcal{M}'}(\mathcal{X}_{\mathbf{D_f}}) \not\longrightarrow \mathcal{Y}_{\mathbf{D_f}}, f_{\mathcal{M}'}(\mathcal{X}_{\mathbf{D_r}}) \longrightarrow \mathcal{Y}_{\mathbf{D_r}}. \quad (1)$$

Here, $\not\longrightarrow$ means the mapping relationship no longer holds.

For IU, let there are a sequence of unlearning task, $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_T\}$ and T is the number of unlearning tasks. And $\{\mathbf{D}_{f,t}, \mathbf{D}_{r,t}\}$ is the forgetting classes and remaining classes belonging to $\mathcal{T}_t$, where $\mathbf{D}_{r,t} = \mathbf{D} - \cup_{i=1}^{t}\mathbf{D}_{f,i}$. Starting from $\mathcal{M}_0 = \mathcal{M}$, the forgetting algorithm $\mathscr{F}$ iteratively processes till t-th unlearning task $\mathcal{T}_t$ to generate $\mathcal{M}_t = \mathscr{F}(\mathcal{M}_{t-1}, \mathbf{D}_{f,t}, \mathbf{D}_{r,t})$ which holds the followings:

$$f_{\mathcal{M}_t}(\mathcal{X}_{\mathbf{D}_{f,i}}) \not\longrightarrow \mathcal{Y}_{\mathbf{D}_{f,i}}, f_{\mathcal{M}_t}(\mathcal{X}_{\mathbf{D}_{r,j}}) \longrightarrow \mathcal{Y}_{\mathbf{D}_{r,j}}, \quad (2)$$

4. Methods

In this section, we introduce **FG-OrIU**, a comprehensive framework for Incremental Unlearning (IU) in pre-trained models (PTM). By imposing orthogonal constraints on both feature representations and gradient updates, FG-OrIU ensures thorough removal of unwanted knowledge while preserving performance on remaining classes. An overview of the framework is shown in Figure 2.

4.1. Feature Subspace Decomposition

PTMs acquire implicit knowledge from pre-training data, benefiting downstream tasks during inference. Unlearning aims to selectively remove the influence of forgetting classes from the PTM [10]. Unlike explicit knowledge in traditional Knowledge Bases (KBs) [49, 55], which can be directly edited, the knowledge in PTMs is implicit and intertwined across all parameters [10, 43, 88]. Effective unlearning requires precisely identifying feature spaces for forgetting and remaining. To achieve this, we decompose features into two subspaces: one capturing forgetting-class contributions and the other preserving remaining-class knowledge. Specifically, we perform singular value decomposition (SVD) on the feature matrix of forgetting classes and use the span of its dominant singular vectors to define the forgetting subspace. Similarly, we apply the same procedure to the remaining classes to construct its subspace. We then detail the process as follows.

Given a PTM, $\mathcal{M}$, we construct a representation matrix $\boldsymbol{R}_{f,i} = \left[\boldsymbol{x}_{1,i}, \ldots, \boldsymbol{x}_{n_{f_i},i}\right] \in \mathbb{R}^{n_{f_i} \times d}$ at each layer l, where columns correspond to representations of n_{f_i} samples from the remaining classes $\mathbf{D_{f,i}}$, and d is the output dimension[3]. SVD is applied to $\boldsymbol{R}_{f,i} = \boldsymbol{U}_{f,i}\boldsymbol{\Sigma}_{f,i}(\boldsymbol{V}_{f,i})^T$, followed by its k-rank approximation $(\boldsymbol{R}_{f,i})_k$ satisfying $||(\boldsymbol{R}_{f,i})_k||_F^2 \geq \epsilon||\boldsymbol{R}_{f,i}||_F^2$ for a threshold ϵ. The subspace $\mathcal{S}_{f,i} = \text{span}\{\boldsymbol{B}_{f,i}\}$ is defining using the first k vectors $\boldsymbol{B}_{f,i} = \{\boldsymbol{u}_{1,i}, \ldots, \boldsymbol{u}_{k,i}\}$ from $\boldsymbol{U}_{f,i}$, capturing directions with the highest singular values. Similarly, the feature matrix $\boldsymbol{R}_{r,i}$ for remaining classes $\mathbf{D_{r,i}}$ defines the subspace $\mathcal{S}_{r,i} = \text{span}\{\boldsymbol{B}_{r,i}\}$. The two sets of layer-wised feature subspace, $\{\mathcal{S}_{f,i}^1, \ldots, \mathcal{S}_{f,i}^l\}$ and $\{\mathcal{S}_{r,i}^1, \ldots, \mathcal{S}_{r,i}^l\}$, represent the core part for forgetting and remaining classes,

[3]Here, we omit the notation l for simplicity.

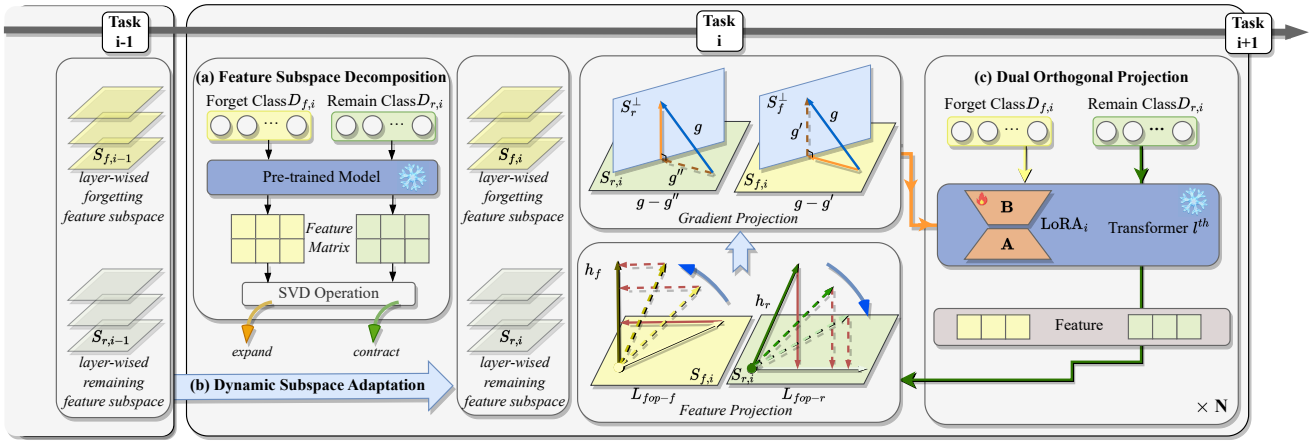


Figure 2. **Overview of FG-OrIU:** (a) For each unlearning task, we decompose the layer-wised- forgetting and remaining subspace separately (Sec. 4.1). (b) LoRA is inserted at each layer, with constraints applied at both the feature and gradient levels for forgetting and remaining (Sec. 4.2). (c) For new unlearning tasks, the forgetting subspace expands while the remaining contracts (Sec. 4.3).

respectively, isolating features to be erased while preserving those to retain. By explicitly decomposing feature subspaces across all layers, we strengthen supervision signals for parameter updates and mitigate the risk of gradient obfuscation [2] in deep networks, where gradient inaccuracies caused by layer-wise nonlinear transformations could otherwise weaken the unlearning effect. This multi-layer constraint ensures consistent suppression of forgetting class features throughout the model hierarchy.

4.2. Dual Orthogonal Projection

Prior to unlearning task $\mathcal{T}_i$, we freeze both the backbone and the classification head of the PTM and insert lightweight LoRA modules [30] at each layer. We then denote the model with LoRA inserted as $\mathcal{M}_\theta$, where θ represents the trainable parameters in LoRA. After unlearning, the updates in LoRA, θ^*, will be merged into the initial weight of the backbone, $\mathcal{M}_i \xrightarrow[\text{merge}]{\theta^*} \mathcal{M}_{i+1}$.

Building on the feature subspaces $\mathcal{S}_f$ and $\mathcal{S}_r$ defined in Section 4.1, we propose a dual orthogonal projection mechanism that operates through both forward feature alignment and backward gradient correction.

4.2.1. Feature Orthogonal Projection

Inspired by [56, 78], for forgetting classes, achieving thorough removal at the feature level requires that the features generated by the unlearned model, $\mathcal{M}_\theta$, deviate as much as possible from the feature subspace $\mathcal{S}_f$. Meanwhile, we aim to ensure that the unlearned model's feature subspace for remaining classes aligns with $\mathcal{S}_r$. The orthogonality constraint between $\mathcal{S}_f$ and the unlearned features serves as a critical mechanism for facilitating class-specific forgetting: under a fixed classification head, features orthogonal to the original subspace $\mathcal{S}_f$ induce maximal perturbations in class logits, as the inner product between orthogonal vectors diminishes to zero. This forces the unlearned model to suppress discriminative patterns associated with forgetting classes in its intermediate representations. To do so, we propose the following two regularization constraints. For forgetting classes, let $\boldsymbol{x_f}$ denote a batch of forgetting samples from $\mathbf{D_{f,i}}$, and $\boldsymbol{h}_f^l$ denote the l-th layer features gained with the unlearned model $\mathcal{M}_\theta$. For complete forgetting, we enforce:

$$\boldsymbol{h}_f^l \perp \mathcal{S}_{f,i}^l \quad \Rightarrow \quad \min ||\mathbf{P}_f^l \boldsymbol{h}_f^l||_F^2 \tag{3}$$

where $\mathbf{P}_f^l = \mathbf{B}_{f,i}^l (\mathbf{B}_{f,i}^l)^\top$ is the projection matrix for the forgetting subspace $\mathcal{S}_{f,i}^l$ and $||\cdot||_F^2$ is the F-norm. Eq.(3) ensures that the features of $\mathbf{D_{f,i}}$ gained from the unlearned model $\mathcal{M}_\theta$ lie outside the original feature subspace gained from $\mathcal{M}$, erasing the unwanted information. Thus, we define the *fop-f loss*:

$$\mathcal{L}_{fop-f} = \sum_{l=1}^{m} ||\mathbf{P}_f^l \boldsymbol{h}_f^l||_F^2. \tag{4}$$

where m denotes the total number of layers in the PTM. For remaining classes, we follow previous works and apply the standard cross-entropy loss by constraining the model's final output, as follows:

$$\mathcal{L}_{ce} = \mathcal{L}(\mathcal{X}_{\mathbf{D}_{r,1}}, \mathcal{Y}_{\mathbf{D}_{r,1}}), \tag{5}$$

where $\mathcal{L}$ denotes the cross-entropy loss function. In addition, we also propose applying explicit constraint at the feature level for the remaining classes to better preserve the desired information. Specifically, $\boldsymbol{x_r}$ denote a batch of remaining samples from $\mathbf{D_{r,i}}$, and and $\boldsymbol{h}_r^l$ denote the l-th layer features gained with the unlearned model $\mathcal{M}_\theta$. For better preservation, we maintain feature fidelity through:

$$\min ||\boldsymbol{h}_r^l - \mathbf{P}_r^l \boldsymbol{h}_r^l||_F^2 \tag{6}$$

where $\mathbf{P}_r^l = \mathbf{B}_{r,i}^l(\mathbf{B}_{r,i}^l)^\top$ is the projection matrix for the remaining feature subspace $\mathcal{S}_{r,i}^l$.

Eq.(6) ensures that the features of $\mathbf{D_{r,i}}$ gained from the unlearned model $\mathcal{M}_\theta$, align with the orignal feature subspace gained from $\mathcal{M}$, maintaining the better preservation. Therefore, We draw the following *fop-r* loss:

$$\mathcal{L}_{fop-r} = \sum_{l=1}^{m} ||\boldsymbol{h}_r^l - \mathbf{P}_r^l \boldsymbol{h}_r^l||_F^2. \tag{7}$$

Finally, the total loss is:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \lambda_1 \mathcal{L}_{fop-f} + \lambda_2 \mathcal{L}_{fop-r}, \tag{8}$$

where λ_1, λ_2 are the coefficients during training.

4.2.2. Gradient Orthogonal Projection

To prevent parameter updates from reintroducing forgotten knowledge while preserving retained features, we design a gradient correction mechanism inspired by gradient projection techniques in multi-task learning [79] and continual learning [51, 86]. Given the raw gradient $\boldsymbol{g_l}$ at layer l, we apply the following modifications:

$$\begin{aligned} \hat{\boldsymbol{g}}_\boldsymbol{l} &= \boldsymbol{g_l} - \boldsymbol{g}_\boldsymbol{l}' - \boldsymbol{g}_\boldsymbol{l}'' \\ &= \boldsymbol{g_l} - \underbrace{\boldsymbol{g_l}(\boldsymbol{I} - \mathbf{B}_{f,i}^l(\mathbf{B}_{f,i}^l)^\top)}_{(term1:gop-f)} - \underbrace{\boldsymbol{g_l}\mathbf{B}_{r,i}^l(\mathbf{B}_{r,i}^l)^\top}_{(term2:gop-r)}, \end{aligned} \tag{9}$$

Term (gop-f), $\boldsymbol{g}_\boldsymbol{l}'$: Removing the components of $\boldsymbol{g_l}$ that are orthogonal to $\mathcal{S}_{f,i}^l$ not only amplifies updates focusing on directions most critical for encoding forgetting class features, but also blocks parameter updates that could reintroduce removed knowledge.

Term (gop-r), $\boldsymbol{g}_\boldsymbol{l}''$: We aim to ensure that forgetting specific classes does not affect the remaining classes. Formally, for the unlearned model $f_{\mathcal{M}_\theta}$ and $\boldsymbol{x}_r$ from the remaining classes $\mathbf{D_{r,i}}$, the following equation holds: $f_{\mathcal{M}_\theta}(x_r) = f_{\mathcal{M}}(x_r)$. Specifically, we consider the input and output of layer l as follows:

$$\begin{aligned} \boldsymbol{W}_l\boldsymbol{x_r} + \Delta\theta_l\boldsymbol{x_r} &= \boldsymbol{W}_l\boldsymbol{x_r} \\ \Delta\theta_l\boldsymbol{x_r} &= \mathbf{0} \end{aligned} \tag{10}$$

where $\Delta\theta_l$ represents the change in the LoRA parameters inserted at layer l, which corresponds to its gradient. $\boldsymbol{W}_l$ is the parameter at layer l from the pre-trained model, and these parameters remain frozen during unlearning.

Eq.(10) indicates that if the update direction of the parameters, *i.e.*, the gradient of the trainable LoRA parameters, lies in the complementary space of $\mathcal{S}_{r,i}^l$, *i.e.*, $\boldsymbol{g_l} \perp \mathcal{S}_{r,i}^l$, then $\Delta\theta\boldsymbol{x_r} = \eta\boldsymbol{g_l}\boldsymbol{x_r} = \mathbf{0}$ (where η is the learning rate), ensuring that the unlearned model's performance on the remaining classes is not affected. Therefore, we further refine the gradient by removing its components along $\mathcal{S}_{r,i}^l$.

Finally, modified gradient $\hat{\boldsymbol{g}}_l$ serves the real gradient for parameter updates at the layer l.

4.3. Dynamic Subspace Adaptation

When encountering a new unlearning task $\mathcal{T}_{i+1}$, our method only needs to update the two decomposed feature subspaces. Since the forgetting classes $\mathbf{D_{f,i+1}}$ in $\mathcal{T}_{i+1}$ are a subset of the remaining classes from the previous task $\mathcal{T}_i$, *i.e.*, $\mathbf{D_{f,i+1}} \subseteq \mathbf{D_{r,i}}$, the feature subspaces can be efficiently adjusted. For simplicity, we omit the notion l of layers.

We first construct a representation matrix $\boldsymbol{R}_{f,i+1}$ using only the forgetting classes $\mathbf{D_{f,i+1}}$. Before applying SVD and k-rank approximation, we remove common bases already present in $\mathcal{S}_{f,i}$ to ensure newly added bases are unique and orthogonal to existing ones:

$$\hat{\boldsymbol{R}}_{f,i+1} = \boldsymbol{R}_{f,i+1} - \boldsymbol{B}_{f,i}\left(\boldsymbol{B}_{f,i}\right)^T\left(\boldsymbol{R}_{f,i+1}\right) \tag{11}$$

$$= \boldsymbol{R}_{f,i+1} - \boldsymbol{R}_{f,i+1}^{proj}. \tag{12}$$

Next, we perform SVD on $\hat{\boldsymbol{R}}_{f,i+1} = \hat{\boldsymbol{U}}_{f,i+1}\hat{\boldsymbol{\Sigma}}_{f,i+1}(\hat{\boldsymbol{V}}_{f,i+1})^T$ and select h new orthogonal from $\hat{\boldsymbol{U}}_{f,i+1}$. Then, the feature subspace of forgetting classes is expaneded as $\mathcal{S}_{f,i+1} = \text{span}\{\boldsymbol{B}_{f,i+1}\}$, where $\boldsymbol{B}_{f,i+1} = [\boldsymbol{B}_{f,i}; \boldsymbol{u}_1, \ldots, \boldsymbol{u}_h]$ with h new bases from $\hat{\boldsymbol{U}}_{f,i+1}$. And the feature subspace $\mathcal{S}_{r,i+1}$ of remaining classes is contracted to remove the overlapping features accordingly.

5. Experiment

5.1. Experimental Setup

Datasets, Benchmarks and Pre-trained Models. Following [83], we evaluate FG-OrIU on Classification and Recognition tasks. For classification, we use ImageNet-100 [50], CUB-200 [60], and Omnibenchmark [82], with a ViT-B/16 [17] pre-trained in PyTorch [42]. For face recognition, we use CASIA-Face100 [83], a subset of CASIA-WebFace [76], and MS-Celeb-100, sampled from [28]. The pre-trained model is the Face Transformer from [87]. We focus on class-unlearning scenarios with two benchmarks: (a) MU, varying numbers of classes are removed; (b) IU, 20 classes are removed per task over 4 tasks. **All setting is in the form of Y-X, which means *experiments start from a PTM (Y classes originally) and forget X classes*.** More details about implementations are in Appendix A.

Metrics. Following [83], we evaluate performance using average accuracy (*Acc*) for both forgetting and remaining classes. The accuracy for forgetting classes should approach zero, while the accuracy for remaining classes should remain close to the original model's performance. Additionally, we adopt the H-Mean [54] metric to assess overall performance after learning task $\mathcal{T}_t$, computed as:

$$H\text{-}Mean^{(t)} = \frac{2Acc_r^{(t)} \cdot Drop^{(t)}}{Acc_r^{(t)} + Drop^{(t)}}. \tag{13}$$

Here $Acc_r^{(t)}$ represents accuracy on the retained dataset after task $\mathcal{T}_t$, while $Drop^{(t)} = Acc_f^{(t-1)} - Acc_f^{(t)}$ mea-

Methods	Tunable Ratio ↓	100-5			100-10			100-50			100-90		
		H ↑	Acc_r ↑	Acc_f ↓	H ↑	Acc_r ↑	Acc_f ↓	H ↑	Acc_r ↑	Acc_f ↓	H ↑	Acc_r ↑	Acc_f ↓
Pre-train	-	-	73.85	70.88	-	73.81	72.74	-	72.45	74.88	-	72.32	73.86
L2*	99.73%	67.13	66.96	3.58	67.74	64.87	1.86	64.39	56.61	0.24	52.45	40.68	0.04
EWC*	99.73%	69.65	68.69	0.24	69.16	65.92	0.00	62.48	53.61	0.00	44.41	31.75	0.00
MAS*	99.73%	69.73	69.29	0.72	69.38	66.41	0.12	62.67	53.88	0.00	46.79	34.24	0.00
LwF	99.73%	67.95	68.55	0.00	70.08	67.62	0.00	63.81	55.59	0.00	61.25	52.32	0.00
DER	99.73%	69.70	68.55	0.00	70.08	67.62	0.00	63.81	55.59	0.00	57.03	46.44	0.00
DER++	99.73%	69.70	68.56	0.00	70.58	68.55	0.00	64.61	56.82	0.00	61.40	52.54	0.00
FDR	99.73%	70.31	69.74	0.00	70.03	67.51	0.00	65.40	58.04	0.00	53.85	42.37	0.00
Retrain	100.00%	16.49	9.33	0.00	18.02	10.28	0.00	19.71	11.35	0.00	46.47	33.90	0.00
SCRUB	99.73%	67.78	64.94	0.00	68.26	64.31	0.00	65.82	58.71	0.00	16.11	9.04	0.00
SCRUB-S	99.73%	70.29	69.95	0.24	71.19	69.81	0.12	57.49	51.20	9.34	17.53	9.94	0.00
LIRF*	50.66%	25.56	67.67	55.13	26.35	65.83	56.26	47.13	58.95	35.62	54.49	44.29	3.06
GS-LoRA	**1.28%**	71.02	71.16	0.00	71.76	70.81	0.00	71.29	68.05	0.02	73.71	73.56	0.00
GS-LoRA++	**1.28%**	71.12	71.36	0.00	72.04	71.35	0.00	71.56	68.52	0.00	72.85	71.86	0.00
FG-OrIU	**1.28%**	**71.78**	72.71	0.00	**72.52**	72.31	0.00	**74.18**	73.51	0.00	**75.49**	77.15	0.00

Table 2. Static Machine Unlearning results for face recognition task on CASIA-Face100. Acc_r (%) and Acc_f (%) are the accuracies of remaining and forgetting classes. * denotes the original methods with a rehearsal buffer. Retrain represents retraining the model using replay data and *the training epoch is the same as other methods to ensure a fair comparison.* Pre-train denotes the results before forgetting.

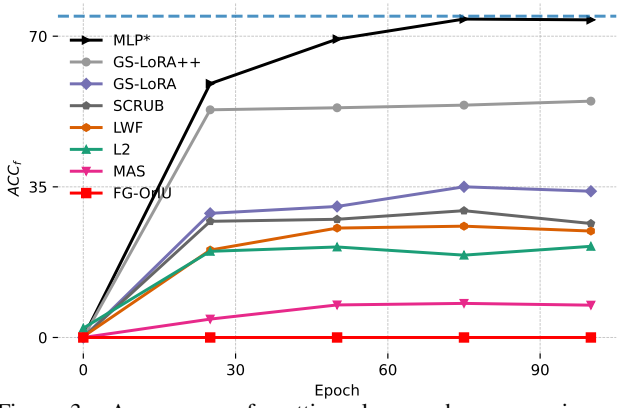


Figure 3. Accuracy on forgetting classes when recovering on CASIA-Face100. The line (Pre-train) is the result before unlearning. The line (FG-OrIU) is Our method.

sures the performance drop on forgetting classes before and after training. After unlearning task $\mathcal{T}_t$, we evaluate performance on all previously forgetting classes from tasks $\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_{t-1}$.

Baselines. Following [83], we compare FG-OrIU against *(a) continual learning (CL) methods*, including L2 regularization, EWC [34], MAS [1], LwF [37], DER [6], and FDR [3], as well as *(b) machine unlearning (MU) methods* such as BAD-T [12], LIRF [75], SCRUB/SCRUB-S [36], GS-LoRA/GS-LoRA++ [83, 84], and retraining. To ensure forgetting occurs in the backbone, we freeze the classification head for all methods. In the retraining setup, we train a randomly initialized model using replay data, ensuring the number of training epochs matches that of other methods.

5.2. Results and Comparison

Table 2 presents the results for static Machine Unlearning task. FG-OrIU exhibits significantly degraded performance on forgetting classes while maintaining the performance of remaining classes. For example, in the settings where 50 and 90 classes are removed per task, FG-OrIU achieves improvements of 2.62% and 2.64% in H, and 4.99% and 5.29% in ACC_r compared to GS-LoRA. Notably, FG-OrIU keeps the pre-trained model frozen and only trains the LoRA modules, enabling effective and efficient forgetting. Table 3 and 4 show the results for Incremental Unlearning task. FG-OrIU consistently outperforms all methods. Specifically, CL methods struggle to achieve effective forgetting on ImageNet100, as seen in DER, which retains high ACC_f of 10.40%, 10.90%, 8.80%, and 3.40% across four tasks, respectively. On the other hand, MU methods fail to maintain their forgetting effect over sequential unlearning tasks. For example, LIRF and SCRUB-S show an increase in ACC_o on CASIA-Face100, which becomes more pronounced as the sequence of unlearning tasks grows. GS-LoRA is specifically designed for IU. Compared to it, our method achieves better forgetting while also demonstrating superior performance in preserving the remaining classes, *i.e.*, substantial improvements in ACC_r, particularly in long-sequence unlearning scenarios, achieving gains of 5.15% and 4.6% in the fourth unlearning task on two different datasets, respectively.

5.3. Ablation Study

Our FG-OrIU consists of four components, *i.e.*, *feature orthogonal projection for forgetting, feature orthogonal projection for remaining, gradient orthogonal projection for forgetting, and gradient orthogonal projection for remaining*, denoted as *fop-f, fop-r, gop-f, and gop-r*. We independently remove each of these components to demonstrate their individual contributions. We observe that when the regularization term or gradient modification related to *remaining* is removed, the performance on the remaining

Methods	*Task 1*(100-20)			*Task 2*(80-20)				*Task 3*(60-20)				*Task 4*(40-20)			
	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$
Pre-train	-	74.31	74.65	-	74.50	73.80	-	-	74.80	73.91	-	-	74.47	75.11	-
L2*	66.91	62.13	2.16	66.66	61.74	1.37	9.36	66.42	61.37	1.54	11.83	66.95	61.02	0.97	8.00
EWC*	67.71	61.95	0.00	67.71	62.55	0.00	**0.00**	67.09	61.43	0.00	0.10	68.02	62.14	0.00	0.23
MAS*	67.52	61.63	0.00	68.12	63.25	0.00	**0.00**	68.15	63.23	0.00	0.03	67.70	61.61	0.00	**0.00**
LwF	69.43	65.04	0.21	69.94	66.47	0.00	0.21	70.34	67.11	0.00	**0.00**	70.89	67.12	0.00	0.04
DER	70.75	67.24	0.00	68.88	64.58	0.00	**0.00**	68.95	64.62	0.00	**0.00**	69.41	64.51	0.00	**0.00**
DER++	70.01	65.90	0.00	68.99	64.77	0.00	**0.00**	69.96	66.42	0.00	**0.00**	69.85	65.28	0.00	**0.00**
FDR	68.29	62.92	0.00	67.39	62.01	0.00	**0.00**	69.16	64.99	0.00	**0.00**	72.51	70.08	0.00	**0.00**
Retrain	17.29	9.77	0.00	31.12	19.72	0.00	**0.00**	41.17	28.80	1.72	**0.00**	52.44	41.77	4.66	0.09
BAD-T	68.90	63.97	0.00	69.56	65.95	0.21	**0.00**	70.21	66.87	0.00	0.34	72.71	70.46	0.00	0.04
LIRF*	30.59	64.46	54.60	34.05	62.03	50.34	43.50	44.56	62.53	39.29	36.58	40.36	62.62	45.34	27.96
SCRUB	70.39	66.59	0.00	70.55	67.85	0.32	**0.00**	71.01	68.33	0.00	0.44	73.36	71.68	0.00	0.05
SCRUB-S	72.41	70.34	0.05	70.06	71.53	5.16	15.22	73.38	73.09	0.24	15.81	75.33	76.24	0.68	6.47
GS-LoRA	74.40	74.16	0.00	73.59	73.37	0.00	0.05	74.36	74.88	0.06	**0.00**	73.76	72.45	0.00	1.93
GS-LoRA++	73.97	74.16	0.00	73.48	73.16	0.00	0.10	74.20	74.51	0.00	**0.00**	75.23	75.36	0.00	**0.00**
FG-OrIU	**74.45**	74.25	0.00	**74.64**	75.50	0.00	**0.00**	**75.49**	77.14	0.00	**0.00**	**77.71**	80.51	0.00	**0.00**

Table 3. Incremental Unlearning results for face recognition task on CASIA-Face100. Acc_o (%) is the accuracy of old tasks, *i.e.*, the accuracy on all previously forgetting classes in task $\{\mathcal{T}_1, \ldots, \mathcal{T}_{t-1}\}$. With four tasks, each forgetting 20 classes, desired forgetting should consider both Acc_r and Acc_f. We **bold** the best and underline the second-best H results.

Methods	*Task 1*(100-20)			*Task 2*(80-20)				*Task 3*(60-20)				*Task 4*(40-20)			
	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$	$H\uparrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$
Pre-train	-	89.93	87.40	-	90.03	89.60	-	-	89.65	90.80	-	-	92.80	86.50	-
L2*	79.39	88.33	15.30	83.59	88.37	10.30	48.10	84.01	88.20	10.60	37.15	82.72	89.90	9.90	20.97
EWC*	82.69	78.55	0.10	85.71	82.57	0.50	0.20	87.33	84.90	0.90	2.90	86.93	89.50	2.00	8.27
MAS*	80.44	74.50	0.00	81.51	74.77	0.00	0.60	82.18	75.05	0.00	0.60	81.93	77.90	0.10	1.77
LwF	85.82	85.15	0.90	87.26	85.40	0.40	2.40	87.35	84.15	0.00	2.35	86.05	85.70	0.10	1.07
DER	82.88	89.73	10.40	84.22	90.57	10.90	50.80	86.36	91.20	8.80	42.00	88.52	94.70	3.40	24.90
DER++	83.45	89.73	9.40	84.13	90.63	11.10	50.10	87.50	91.30	6.80	40.40	**89.24**	94.80	2.20	20.80
FDR	29.51	17.78	0.50	36.21	22.70	0.10	**0.00**	40.79	26.30	0.00	**0.00**	47.91	33.50	2.40	1.27
Retrain	48.20	33.28	0.00	56.28	41.30	1.30	**0.00**	61.14	47.55	5.20	0.20	67.23	60.60	11.00	0.53
BAD-T	74.74	65.29	0.00	78.09	69.21	0.00	0.10	76.65	66.32	0.00	**0.00**	76.25	68.17	0.00	**0.00**
LIRF*	52.72	64.28	42.71	61.05	69.83	35.36	32.15	60.96	60.32	29.18	30.91	58.51	68.46	35.41	21.04
SCRUB	75.87	67.03	0.00	76.23	66.33	0.00	**0.00**	75.46	64.55	0.00	**0.00**	77.56	70.30	0.00	**0.00**
SCRUB-S	83.44	79.83	0.00	82.59	76.60	0.00	0.10	79.81	71.20	0.00	**0.00**	78.23	71.40	0.00	0.07
GS-LoRA	86.13	85.00	0.10	87.33	85.27	0.10	1.10	87.84	85.15	0.10	0.35	86.30	86.30	0.20	0.07
GS-LoRA++	86.63	85.98	0.10	87.22	85.23	0.30	0.30	87.48	84.40	0.00	0.60	87.00	87.50	0.00	0.07
FG-OrIU	**87.01**	86.73	**0.10**	**88.45**	87.43	0.10	0.10	**88.90**	87.35	0.30	**0.00**	89.15	92.10	0.10	**0.00**

Table 4. Incremental Unlearning results for classification task on ImageNet100.

Variants	*Task 1(100-20)*		*Task 2(80-20)*		
	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_r\uparrow$	$Acc_f\downarrow$	$Acc_o\downarrow$
Pre-train	74.30	74.60	74.50	73.80	-
FG-OrIU	**74.25**	**0.00**	75.50	**0.00**	**0.00**
w/o *fop-f*	73.01	0.20	76.24	0.42	1.28
w/o *gop-f*	73.92	0.10	76.58	4.37	0.82
w/o *fop-r*	73.92	0.05	74.98	0.21	0.15
w/o *gop-r*	71.45	0.00	73.96	0.00	0.00

Table 5. Ablation studies on CASIA-Face100.

Exps	Metirc	Pre-train	DER	FDR	SCRUB	GS-LoRA	**Ours**
Exp1	SSIM	0.94	0.60	0.34	0.75	0.39	**0.01**
	PSNR	31.32	21.22	16.89	23.40	17.60	**8.37**
Exp2	SSIM	0.95	0.49	0.66	0.63	0.38	**0.02**
	PSNR	31.15	19.17	22.15	21.12	17.32	**8.72**
Exp3	SSIM	0.93	0.53	0.49	0.60	0.41	**0.01**
	PSNR	31.10	20.03	19.83	22.01	15.92	**8.55**

Table 6. SSIM ↑ and PSNR(db) ↑ scores for the reconstructed images using DIP. Here, Pre-train serves as the upper bound.

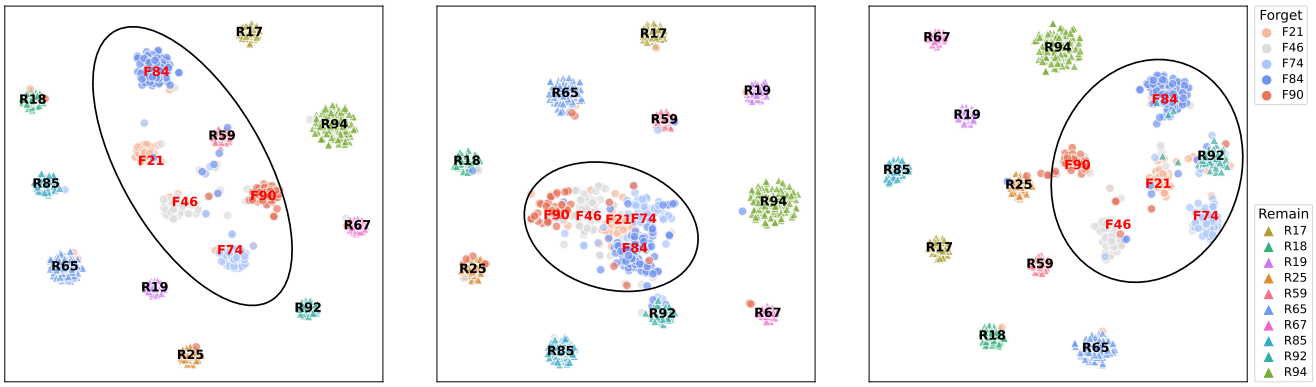
classes significantly declines. Meanwhile, when the regularization term or gradient modification related to *forgetting* is removed, the performance on the forgetting classes shows a slight improvement. Thus, each component provides clear benefits in achieving a better trade-off between forgetting and remaining while preserving the old forgetting effect.

5.4. Discussion

IU is designed to ensure strict privacy protection. This raises several key questions: (a) Does the unlearned model still retain any private information? (b) Can the forgetting effect be reversed? and (c) How does unlearning impact the feature subspace? To address these questions, we design the following experiment.

5.4.1. Semantic Residual Analysis

We first investigate the semantic information contained in features after unlearning. We utilize Deep Image Prior (DIP) [59] to reconstruct an image from the features ob-



(a) Pre-train (b) Ours (c) GS-LoRA

Figure 4. TSNE visualization with 5 forgetting classes and 10 remaining classes, using different model as feature extractor: (a) the pre-trained model, (b) the unlearned model through FG-OrIU, and (c) through GS-LoRA.

tained from the last layer of the unlearned model. Figure 1 presents the visualization of the reconstructed image, which allows us to infer the semantic information encoded in features. More results are provided in the Appendix D.

Additionally, we measure the SSIM and PSNR scores for the images generated by DIP from the unlearned model using different IU methods as a quantitative results. As shown in Table 6, the SSIM values of the reconstructed images from the compared methods range between 0.35 and 0.75, with the highest being 0.75 for SCRUB on *exp1*, while the PSNR values range between 16 and 24, with the highest being 23.40 for SCRUB on *exp1*. This indicates that the reconstructed image achieves a high similarity with the original image, suggesting that the information in features has been partially preserved. In contrast, our method yields an SSIM of only 0.01 and a PSNR of 8.55, indicating a significant difference between the reconstructed and original images. This suggests that the semantic information at the feature level has been substantially obfuscated.

5.4.2. Forgetting Recoverable Analysis

Given an unlearned model obtained through different IU methods, we freeze its backbone and retrain a new classification head using the entire training dataset, which includes both forgetting and remaining classes. As illustrated in Figure 3, MLP* represents a trivial unlearning method where the classification head's parameters for the forgetting classes are simply set to zero. Consequently, the underlying features remain intact, enabling effortless recovery of the forgotten knowledge. Additionally, existing IU methods primarily impose constraints on the final output rather than directly modifying feature representations. As a result, while features may be weakened to some extent, their core structure remains largely unchanged. Thus, these methods allow the forgotten knowledge to be recovered to 15%–70% of its original performance, which leads to *superficial forgetting*. In contrast, our method achieves *deep forgetting*, making recovery infeasible.

5.4.3. Feature Subspace Analysis

To analyze the impact of direct constraints on features and gradients for forgetting and remaining classes, we conduct an experiment. We remove 20 classes while retaining 80, then extract features from 5 forgetting and 10 remaining classes and visualize them using t-SNE.

A key observation is that GS-LoRA causes some forgetting class samples to move closer to or even overlap with certain remaining class samples (e.g., F21, F74 → R92 in c). This suggests that merely applying constraints at the final output, resulting in degraded features from the forgetting classes entangling with those of the remaining classes. Consequently, this degradation negatively impacts the performance of the remaining classes. In contrast, our method enforces forgetting at both feature and gradient level, ensuring that features from the forgetting classes do not shift toward any remaining class but instead become linearly inseparable within the forgetting set. Notably, an unexpected yet beneficial side effect emerges: in the original PTM, certain remaining class samples (e.g., R59) were slightly entangled with forgetting classes (e.g., F90 and F46). However, after applying our method, these classes become completely disentangled. This observation aligns with the results in Table 3, where the accuracy of the remaining classes improves after unlearning (e.g., 75.36 → 80.51), demonstrating that our method not only enhances forgetting effectiveness but also benefits the generalization of the remaining classes.

6. Conclusion

In this work, we propose FG-OrIU for incremental unlearning in vision PTMs. By enforcing orthogonal constraints on both features and gradients, FG-OrIU achieves irreversible forgetting while maintaining performance on the remaining classes. Its dynamic subspace adaptation ensures robust handling of sequential deletion requests. Extensive experiments confirm its superiority. While FG-OrIU is designed for class-level unlearning, extending it to sample-level settings remains a promising direction for future work.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant 62402430, 62206248, 62476238, Zhejiang Provincial Natural Science Foundation of China under Grant LQN25F020008, and Aeronautical Science Foundation of China 20240048076001. Qian Feng would like to extend his gratitude to Jiahang Tu and Hanbin Zhao from Zhejiang University for their discussions regarding the methods and experiments. Additionally, he is thankful to Mintong Kang from UIUC for his dedicated suggestions on improving the overall writing of the paper.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018. 6

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018. 4

[3] Ari S Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. *arXiv preprint arXiv:1805.08289*, 2018. 6

[4] Jacopo Bonato, Marco Cotogni, and Luigi Sabetta. Is retain set all you need in machine unlearning? restoring performance of unlearned models with out-of-distribution images. In *European Conference on Computer Vision*, pages 1–19. Springer, 2024. 1

[5] 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

[6] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 3, 6

[7] Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, pages 463–480. IEEE, 2015. 1

[8] Romit Chatterjee, Vikram Chundawat, Ayush Tarun, Ankur Mali, and Murari Mandal. A unified framework for continual learning and machine unlearning. *arXiv preprint arXiv:2408.11374*, 2024. 1, 3

[9] Jiaao Chen and Diyi Yang. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*, 2023.

[10] Xinwen Cheng, Zhehao Huang, Wenxin Zhou, Zhengbao He, Ruikai Yang, Yingwen Wu, and Xiaolin Huang. Remaining-data-free machine unlearning by suppressing sample contribution. *arXiv preprint arXiv:2402.15109*, 2024. 1, 2, 3

[11] Zhi-Yi Chin, Chieh-Ming Jiang, Ching-Chun Huang, Pin-Yu Chen, and Wei-Chen Chiu. Prompting4debugging: Red-teaming text-to-image diffusion models by finding problematic prompts. *arXiv preprint arXiv:2309.06135*, 2023. 1, 3

[12] 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. 1, 2, 3, 6

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1

[14] Meng Ding, Jinhui Xu, and Kaiyi Ji. Why fine-tuning struggles with forgetting in machine unlearning? theoretical insights and a remedial approach. *arXiv preprint arXiv:2410.03833*, 2024. 3

[15] Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1

[16] Jiahua Dong, Hongliu Li, Yang Cong, Gan Sun, Yulun Zhang, and Luc Van Gool. No one left behind: Real-world federated class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(4): 2054–2070, 2024. 1

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 5, 1

[18] Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. *arXiv preprint arXiv:2310.12508*, 2023. 2, 3

[19] Qian Feng, Da-Wei Zhou, Hanbin Zhao, Chao Zhang, Jiahua Dong, Dengxin Dai, and Hui Qian. Lw2g: Learning whether to grow for prompt-based continual learning. *arXiv preprint arXiv:2409.18860*, 2024. 1

[20] Qian Feng, Hanbin Zhao, Chao Zhang, Jiahua Dong, Henghui Ding, Yu-Gang Jiang, and Hui Qian. Pectp: Parameter-efficient cross-task prompts for incremental vision transformer. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 1

[21] Rita Fermanian, Mikael Le Pendu, and Christine Guillemot. Regularizing the deep image prior with a learned denoiser for linear inverse problems. In *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2021. 5

[22] Jack Foster, Stefan Schoepf, and Alexandra Brintrup. Fast machine unlearning without retraining through selective synaptic dampening. In *Proceedings of the AAAI conference on artificial intelligence*, pages 12043–12051, 2024. 1, 2, 3

[23] Yosef Gandelsman, Assaf Shocher, and Michal Irani. " double-dip": unsupervised image decomposition via coupled deep-image-priors. In *Proceedings of the IEEE/CVF con-*

ference on computer vision and pattern recognition, pages 11026–11035, 2019. 5

[24] Chongyang Gao, Lixu Wang, Chenkai Weng, Xiao Wang, and Qi Zhu. Practical unlearning for large language models. *arXiv preprint arXiv:2407.10223*, 2024. 1, 2, 3

[25] 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

[26] Keltin Grimes, Collin Abidi, Cole Frank, and Shannon Gallagher. Gone but not forgotten: Improved benchmarks for machine unlearning. *arXiv preprint arXiv:2405.19211*, 2024. 1

[27] Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. *arXiv preprint arXiv:1911.03030*, 2019. 1

[28] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 87–102. Springer, 2016. 5, 1

[29] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024. 1

[30] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 4

[31] Zhehao Huang, Xinwen Cheng, JingHao Zheng, Haoran Wang, Zhengbao He, Tao Li, and Xiaolin Huang. Unified gradient-based machine unlearning with remain geometry enhancement. *arXiv preprint arXiv:2409.19732*, 2024. 1

[32] Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*, 2022. 1

[33] Jinghan Jia, Jiancheng Liu, Parikshit Ram, Yuguang Yao, Gaowen Liu, Yang Liu, Pranay Sharma, and Sijia Liu. Model sparsity can simplify machine unlearning. *Advances in Neural Information Processing Systems*, 36:51584–51605, 2023. 1, 3

[34] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 3, 6

[35] Vishnuprasadh Kumaravelu, Sayanta Adhikari, and PK Srijith. Uncle: An unlearning framework for continual learning. 1, 3

[36] Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. *Advances in neural information processing systems*, 36: 1957–1987, 2023. 2, 3, 6

[37] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 6

[38] Bo Liu, Qiang Liu, and Peter Stone. Continual learning and private unlearning. In *Conference on Lifelong Learning Agents*, pages 243–254. PMLR, 2022. 1, 3

[39] 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

[40] Gary Mataev, Peyman Milanfar, and Michael Elad. Deepred: Deep image prior powered by red. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 5

[41] Qihe Pan, Zhen Zhao, Zicheng Wang, Sifan Long, Yiming Wu, Wei Ji, Haoran Liang, and Ronghua Liang. Towards small object editing: A benchmark dataset and a training-free approach. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3257–3265, 2024. 1

[42] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5, 1

[43] Laiqiao Qin, Tianqing Zhu, Linlin Wang, and Wanlei Zhou. Machine unlearning on pre-trained models by residual feature alignment using lora. *arXiv preprint arXiv:2411.08443*, 2024. 1, 2, 3

[44] Sadia Qureshi, Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Jianming Yong, and Xiaohua Jia. Exploring incremental unlearning: Techniques, challenges, and future directions. *arXiv preprint arXiv:2502.16708*, 2025. 1, 3

[45] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 1

[46] Protection Regulation. General data protection regulation. *Intouch*, 25:1–5, 2018. 1

[47] Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3341–3350, 2020. 5

[48] Jeffrey Rosen. The right to be forgotten. *Stan. L. Rev. Online*, 64:88, 2011. 1

[49] Dan Roth and Wen-tau Yih. Probabilistic reasoning for entity & relation recognition. In *COLING 2002: The 19th International Conference on Computational Linguistics*, 2002. 3

[50] 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. 5

[51] Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021. 2, 5

[52] Thanveer Shaik, Xiaohui Tao, Haoran Xie, Lin Li, Xiaofeng Zhu, and Qing Li. Exploring the landscape of machine unlearning: A comprehensive survey and taxonomy. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. 1

[53] Yingdan Shi and Ren Wang. Redefining machine unlearning: A conformal prediction-motivated approach. *arXiv preprint arXiv:2501.19403*, 2025. 1

[54] Takashi Shibata, Go Irie, Daiki Ikami, and Yu Mitsuzumi. Learning with selective forgetting. In *IJCAI*, page 6, 2021. 3, 5

[55] Mihai Surdeanu and Heng Ji. Overview of the english slot filling track at the tac2014 knowledge base population evaluation. In *Proc. Text Analysis Conference (TAC2014)*, 2014. 3

[56] Jiahang Tu, Qian Feng, Chufan Chen, Jiahua Dong, Hanbin Zhao, Chao Zhang, and Hui Qian. Ce-sdwv: Effective and efficient concept erasure for text-to-image diffusion models via a semantic-driven word vocabulary. *arXiv preprint arXiv:2501.15562*, 2025. 1, 2, 3, 4

[57] Jiahang Tu, Hao Fu, Fengyu Yang, Hanbin Zhao, Chao Zhang, and Hui Qian. Texttoucher: Fine-grained text-to-touch generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7455–7463, 2025. 1

[58] Jiahang Tu, Wei Ji, Hanbin Zhao, Chao Zhang, Roger Zimmermann, and Hui Qian. Driveditfit: Fine-tuning diffusion transformers for autonomous driving data generation. *ACM Transactions on Multimedia Computing, Communications and Applications*, 21(3):1–29, 2025. 1

[59] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454, 2018. 1, 2, 7, 5

[60] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 5

[61] Cheng-Long Wang, Qi Li, Zihang Xiang, and Di Wang. Has approximate machine unlearning been evaluated properly? from auditing to side effects. *arXiv e-prints*, pages arXiv–2403, 2024. 1, 2, 3

[62] Fangyikang Wang, Hubery Yin, Yue-Jiang Dong, Huminhao Zhu, Hanbin Zhao, Hui Qian, Chen Li, et al. Belm: Bidirectional explicit linear multi-step sampler for exact inversion in diffusion models. *Advances in Neural Information Processing Systems*, 37:46118–46159, 2024. 1

[63] Fangyikang Wang, Huminhao Zhu, Chao Zhang, Hanbin Zhao, and Hui Qian. Gad-pvi: A general accelerated dynamic-weight particle-based variational inference framework. *Entropy*, 26(8):679, 2024.

[64] Fangyikang Wang, Hubery Yin, Lei Qian, Yinan Li, Shaobin Zhuang, Huminhao Zhu, Yilin Zhang, Yanlong Tang, Chao Zhang, Hanbin Zhao, et al. Unleashing high-quality image generation in diffusion sampling using second-order levenberg-marquardt-langevin. *arXiv preprint arXiv:2505.24222*, 2025.

[65] Fangyikang Wang, Hubery Yin, Shaobin Zhuang, Huminhao Zhu, Yinan Li, Lei Qian, Chao Zhang, Hanbin Zhao, Hui Qian, and Chen Li. Efficiently access diffusion fisher: Within the outer product span space. *arXiv preprint arXiv:2505.23264*, 2025. 1

[66] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 139–149, 2022. 1

[67] Yiming Wu, Qihe Pan, Zhen Zhao, Zicheng Wang, Sifan Long, and Ronghua Liang. Soediff: Efficient distillation for small object editing. *ACM Transactions on Multimedia Computing, Communications and Applications*. 1

[68] Yiming Wu, Wei Ji, Kecheng Zheng, Zicheng Wang, and Dong Xu. Mote: Learning motion-text diffusion model for multiple generation tasks. *arXiv preprint arXiv:2411.19786*, 2024.

[69] Yiming Wu, Hangfei Li, Fangfang Wang, Yilong Zhang, and Ronghua Liang. Self-distilled dynamic fusion network for language-based fashion retrieval. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3260–3264. IEEE, 2024.

[70] Yiming Wu, Huan Wang, Zhenghao Chen, and Dong Xu. Individual content and motion dynamics preserved pruning for video diffusion models. *arXiv preprint arXiv:2411.18375*, 2024. 1

[71] Heng Xu, Tianqing Zhu, Lefeng Zhang, Wanlei Zhou, and Philip S Yu. Machine unlearning: A survey. *ACM Computing Surveys*, 56(1):1–36, 2023. 2

[72] Jie Xu, Zihan Wu, Cong Wang, and Xiaohua Jia. Machine unlearning: Solutions and challenges. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024. 1

[73] Lingling Xu, Haoran Xie, Si-Zhao Joe Qin, Xiaohui Tao, and Fu Lee Wang. Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment. *arXiv preprint arXiv:2312.12148*, 2023. 1

[74] Haonan Yan, Xiaoguang Li, Ziyao Guo, Hui Li, Fenghua Li, and Xiaodong Lin. Arcane: An efficient architecture for exact machine unlearning. In *IJCAI*, page 19, 2022. 2, 3

[75] Jingwen Ye, Yifang Fu, Jie Song, Xingyi Yang, Songhua Liu, Xin Jin, Mingli Song, and Xinchao Wang. Learning with recoverable forgetting. In *ECCV*, 2022. 6

[76] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. 5, 1

[77] Seungryong Yoo, Eunji Kim, Dahuin Jung, Jungbeom Lee, and Sungroh Yoon. Improving visual prompt tuning for self-supervised vision transformers. In *International Conference on Machine Learning*, pages 40075–40092. PMLR, 2023. 5, 7

[78] Jaehong Yoon, Shoubin Yu, Vaidehi Patil, Huaxiu Yao, and Mohit Bansal. Safree: Training-free and adaptive guard for safe text-to-image and video generation. *arXiv preprint arXiv:2410.12761*, 2024. 2, 4

[79] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in Neural Information Processing Systems*, 33:5824–5836, 2020. 2, 5

[80] Dan Zhang, Tao Feng, Lilong Xue, Yuandong Wang, Yuxiao Dong, and Jie Tang. Parameter-efficient fine-tuning for foundation models. *arXiv preprint arXiv:2501.13787*, 2025. 1

[81] Haibo Zhang, Toru Nakamura, Takamasa Isohara, and Kouichi Sakurai. A review on machine unlearning. *SN Computer Science*, 4(4):337, 2023. 2

[82] Yuanhan Zhang, Zhenfei Yin, Jing Shao, and Ziwei Liu. Benchmarking omni-vision representation through the lens of visual realms. In *European Conference on Computer Vision*, pages 594–611. Springer, 2022. 5

[83] Hongbo Zhao, Bolin Ni, Junsong Fan, Yuxi Wang, Yuntao Chen, Gaofeng Meng, and Zhaoxiang Zhang. Continual forgetting for pre-trained vision models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28631–28642, 2024. 1, 3, 5, 6

[84] Hongbo Zhao, Fei Zhu, Bolin Ni, Feng Zhu, Gaofeng Meng, and Zhaoxiang Zhang. Practical continual forgetting for pre-trained vision models. *arXiv preprint arXiv:2501.09705*, 2025. 1, 2, 3, 6

[85] Nanxuan Zhao, Zhirong Wu, Rynson WH Lau, and Stephen Lin. What makes instance discrimination good for transfer learning? *arXiv preprint arXiv:2006.06606*, 2020. 5, 7

[86] Zhen Zhao, Zhizhong Zhang, Xin Tan, Jun Liu, Yanyun Qu, Yuan Xie, and Lizhuang Ma. Rethinking gradient projection continual learning: Stability/plasticity feature space decoupling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3718–3727, 2023. 2, 5

[87] Yaoyao Zhong and Weihong Deng. Face transformer for recognition. *arXiv preprint arXiv:2103.14803*, 2021. 5, 1

[88] Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*, 2020. 3

FG-OrIU: Towards Better Forgetting via Feature-Gradient Orthogonality for Incremental Unlearning

Supplementary Material

Overview. In this supplementary material, we provide the source code in the 'FG-OrIU' folder and present additional experiments related to our method as follows:

In Section A, we provide detailed information about the datasets used, the benchmarks constructed separately for machine unlearning and incremental unlearning, the pre-trained models adopted along with their sources, and the hyperparameters used in our experiments.

In Section C, we provide the pseudocode of our FG-OrIU method.

In Section B, we present additional experimental results in the IU scenario, including results from the face recognition and image classification tasks. Furthermore, we provide additional visualization results using T-SNE.

In Section D, we provide a detailed introduction to Deep Image Prior, the technique used for feature reconstruction in this work, including its principles, implementation, and additional visualization results.

A. Implementation Details

In this section, we discuss the detailed implementation in FG-OrIU, including the introduction of datasets, selection or pre-training of the pre-trained models, and hyperparameters.

A.1. Datasets

In this section, we introduce the datasets used in this paper. The details of the five adopted datasets are listed in Table 7. Specifically, CASIA-Face100 is a subset of CASIA-WebFace [76], provided by [83]. Following a similar approach, we sample a subset with 100 face identities from the large-scale face dataset MS-Celeb-1M [28]. Additionally, CUB200, ImageNet-100, and OmniBenchmark are widely used benchmark datasets for continual learning, frequently adopted in studies such as [15, 16, 19, 20, 41, 45, 57, 58, 62–65, 67–70].

A.2. Pre-Trained Models

Before performing unlearning, we first need to obtain a pre-trained model. Therefore, to validate that our method can be flexibly scaled and effectively applied across models of different parameter sizes for incremental unlearning, we evaluate multiple pre-trained models, detailed as follows:

Face Recognition Task on CASIA-Face100 Dataset

Following [83], we train a face transformer with six transformer blocks from scratch on the CASIA-Face100 dataset and use it as the original model.

Face Recognition Task on MS-Celeb-100 Dataset

We directly utilize the checkpoints released by [87], specifically a face transformer with 20 transformer blocks. This model was pre-trained on the large-scale face dataset MS-Celeb-1M [28], and the released checkpoint includes a classification head with a dimensionality of 93,431. We then freeze the backbone and perform a linear probe, retraining a classification head with a dimensionality of 100 on the MS-Celeb-100 dataset to obtain the original model.

Image Classification Task on ImageNet-100 Dataset

Following [83], we use the pre-trained ViT-B/16 model [17], which was trained in PyTorch [42] on ImageNet-21K [13], as the original model.

Image Classification Task on CUB-200 dataset

We utilize the pre-trained ViT-B/16 model, freeze its backbone, and perform a linear probe by retraining a classification head with a dimensionality of 200 on the CUB-200 dataset to obtain the original model.

Image Classification Task on OmniBenchmark Dataset

Similarly, we use the pre-trained ViT-B/16 model, freeze its backbone, and perform a linear probe by retraining a classification head with a dimensionality of 300 on the OmniBenchmark dataset to obtain the original model.

A.3. Implementations and Hyper-Parameters

We follow the implementation of GS-LoRA[4] [83] to re-implement all the compared methods, including L2*, EWC*, MAS*, LwF, DER, DER++, FDR, Retrain, BAD-T, LRIF*, SCRUB, SCRUB-S, GS-LoRA, and GS-LoRA++. To ensure a fair comparison, we strictly follow the implementation details provided in [83], including data augmentation strategies and random seed settings. Following [66, 83], all results are averaged over five runs with different seeds (42, 288, 488, 688, 1337). We use 1 4090 GPU for experiments in Face recognition Task and 1 A800 GPU for experiments in Image Classification Task.

Our FG-OrIU method involves three hyperparameters: λ_1, λ_2, and ϵ. The first two coefficients control the orthogonal constraints applied at the feature level. To analyze the impact of these coefficients in Eq. (8) on both forgetting and retention, we conduct experiments with different values of λ_1 and λ_2. The results are presented in Figure 5.

We vary λ_1 across $\{0.1, 0.2, 0.5, 1, 5\}$ and find that the model achieves optimal performance when $\lambda_1 = 1$. Similarly, we experiment with λ_2 values of $\{0.1, 0.2, 0.5, 1,$

[4] https://github.com/bjzhb666/GS-LoRA

Table 7. Introduction about benchmark datasets. CASIA-Face100 and MS-Celeb-100 are for Face Recognition Task. CUB-200, ImageNet-100, and OmniBenchmark are for Image Classification Task.

Dataset	# training instances	# testing instances	# Classes	Link
CASIA-Face100	35713	8984	100	Link
MS-Celeb-100	10,517	2684	100	Link
CUB-200	9,430	2,358	200	Link
ImageNet-100	130,000	5000	100	Link
OmniBenchmark	89,668	5,983	300	Link

5} and observe that our method attains the highest accuracy when $\lambda_2 = 0.2$.

The parameter ϵ serves as a threshold for both constructing the feature subspace and its dynamic adaptations. Following [51, 86], we set $\epsilon = 0.99$ as the default value.

B. Extra Experimental Evaluations

B.1. Additional Results on Computational cost and memory overhead

FG-OrIU is designed for efficiency. Both the SVD step 4.1 and dynamic subspace updates 11 are performed **only once per task**. Profiling with *ptflops* shows the extra computational cost is only ≈1.5 GFLOPs per layer, compared to ≈17 GFLOPs per layer for ViT-B/16.

In terms of memory, we provide detailed subspace statistics using the 6-layer Face-Transformer. In Task 1, the number of bases per layer is [13,33,49,55,27,19] for the forgetting subspace and [14,35,55,71,65,69] for the remaining subspace; in Task 4, [14,35,55,71,64,67] (increases) and [14,33,49,57,34,20] (decreases). Each base is a 768-dim float32 vector, requiring only ≈0.2–0.3 MB per layer,far smaller than storing raw features/activations.

To further assess practical impact, we compare wall-clock Runtime and peak GPU memory with baseline methods. As shown in Table 8, FG-OrIU remains lightweight with moderate Runtime and GPU usage, while achieving the highest performance with a notable gain over all baselines.

Metric	SCRUB	DER++	GS-LoRA	FG-OrIU
Runtime (s) / GPU Memory (MiB)	1900/3753	2100/9611	2700/6299	4000/6371
Performance (*H-Mean* via Eq.13)	73.36	69.85	73.76	77.71

Table 8. Runtime and GPU memory comparison on A800 GPU.

B.2. Additional Comparison under Identical Tunable Ratio

A lower tunable ratio enhances parameter efficiency and mitigates interference with remaining-class representations, thereby better preserving retention performance. To further investigate this effect, we conducted additional experiments under a fixed tunable ratio of 1.28%, consistent with GS-LoRA and FG-OrIU, where the backbone is frozen and only the LoRA modules are trainable. As shown in Table 9, although reducing the tunable ratio improves performance on $\mathbf{D_r}$, it significantly impairs forgetting effectiveness on $\mathbf{D_f}$. Existing methods struggle to balance this trade-off. In contrast, our approach achieves a more favorable balance between forgetting and retention, while maintaining high parameter efficiency.

B.3. Additional Results on Incremental Unlearning

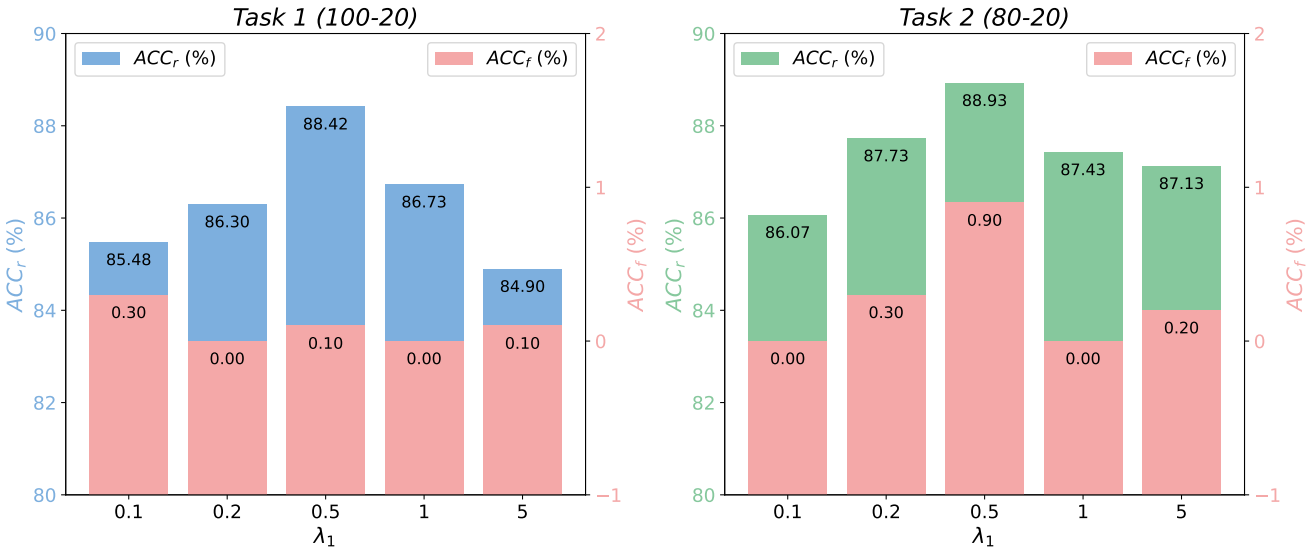
In this section, we conduct experiments to evaluate the scalability of FG-OrIU. Specifically, we perform incremental unlearning on the CUB-200 dataset, MS-Celeb-100, and OmniBenchmark using pre-trained models. On CUB-200, we set up four unlearning tasks, each with 20 classes designated as forgetting classes. On MS-Celeb-100, we similarly set up four unlearning tasks, each with 20 forgetting classes. On OmniBenchmark, we set up four unlearning tasks, but each task contains 50 forgetting classes. For a fair comparison, we re-implemented all baseline methods, and the corresponding results are presented in Tables 1, 2, and 3, respectively.

B.4. Additional Results on Machine Unlearning

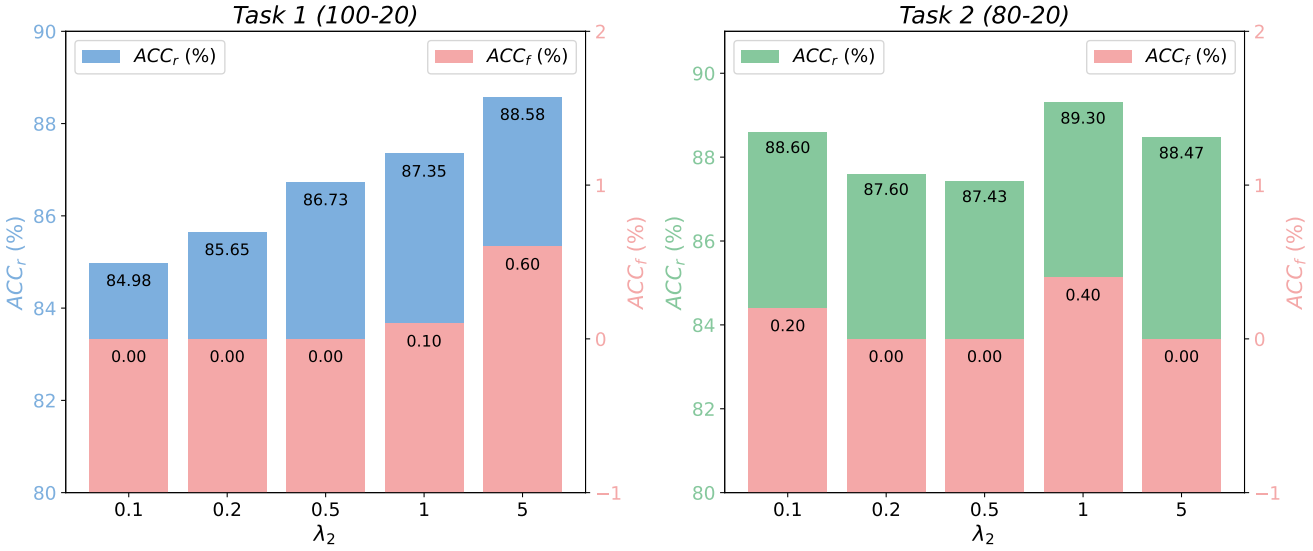
Furthermore, we extend our evaluation to static machine unlearning on CASIA-Face100 and ImageNet-100 by varying the number of forgetting classes across different settings:$\{1, 5, 10, 20, 40, 60, 80, 90, 95, 99\}$. The results for CASIA-Face100 and ImageNet-100 are reported in Tables 13 and 14, respectively.

B.5. More Visualization Results with T-SNE

We present visualizations of the feature subspace after unlearning 20 classes from CASIA-Face100. In Figures 6, 7, and 8, we randomly sample five classes from the forgetting set and ten classes from the remaining set to construct a feature matrix, which is then visualized using T-SNE. Figures 9 and 10 further illustrate the distribution of all 20 forgetting classes and 80 remaining classes when processed by the unlearned models obtained through our FG-OrIU method and GS-LoRA.



(a) Ablations on λ_1 condition with $\lambda_2 = 1$.



(b) Ablations on λ_2 condition with $\lambda_1 = 0.2$.

Figure 5. Ablation studies on λ_1 and λ_2 on ImageNet100 of Incremental Unlearning.

Method	*Task 1*(100-20)		*Task 2*(80-20)			*Task 3*(60-20)			*Task 4*(40-20)		
	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$
EWC*	71.60	11.67	73.37	13.88	9.97	73.52	7.75	5.64	74.52	4.48	1.42
DER++	71.92	13.11	73.37	15.46	15.21	74.15	7.04	9.72	75.47	7.44	2.23
GS-LoRA	74.16	0.00	73.37	0.00	0.05	74.88	0.06	**0.00**	72.45	0.00	1.93
FG-OrIU	74.25	0.00	75.50	0.00	**0.00**	77.14	0.00	**0.00**	80.51	0.00	**0.00**

Table 9. Extended results for Table 3 **under 1.28% tunable ratio**.

Based on these results, we draw the following analysis. Prior works, such as GS-LoRA, impose constraints only at the final output, which results in the unlearned model extracting degraded features for the forgetting classes. These degraded features become easily entangled with those from the remaining classes, leading to a dilemma where the

Methods	*Task 1*(100-20)		*Task 2*(80-20)			*Task 3*(60-20)			*Task 4*(40-20)		
	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$
Pre-train	99.53	99.61	99.63	99.25	-	99.45	100.0	-	99.30	99.63	-
L2*	98.43	0.19	98.58	0.74	23.71	98.18	0.76	25.52	97.69	0.37	8.52
EWC*	99.44	0.00	99.32	0.00	0.00	99.55	0.00	0.00	98.93	0.00	0.00
MAS*	99.44	0.00	99.38	0.00	0.00	99.45	0.00	0.00	98.93	0.00	0.00
LwF	98.80	0.00	99.51	0.00	0.00	99.64	0.00	0.00	99.47	0.00	0.00
DER	99.72	0.00	99.69	0.00	0.00	99.90	0.00	0.00	99.46	0.00	0.00
DER++	99.67	0.00	99.69	0.00	0.00	99.63	0.00	0.00	99.64	0.00	0.00
FDR	52.58	0.00	28.84	0.00	0.00	42.49	0.00	0.00	53.28	0.00	0.00
Retrain	38.27	0.00	50.06	1.11	0.00	57.32	11.09	0.00	64.12	25.75	1.07
BAD-T	99.13	0.00	99.62	0.00	0.00	99.58	0.00	0.00	99.77	0.00	0.00
LIRF*	86.35	81.64	81.81	74.03	55.64	76.71	56.60	40.87	73.00	63.99	25.05
SCRUB	99.58	0.00	99.14	0.00	0.00	98.27	0.00	0.00	97.87	0.00	0.00
SCRUB-S	99.49	0.00	99.57	0.00	0.00	99.09	0.00	0.00	98.93	0.00	0.00
GS-LoRA	99.40	0.00	99.44	0.00	0.00	99.09	0.00	0.00	99.12	0.00	0.00
GS-LoRA++	99.35	0.00	99.45	0.00	0.00	99.00	0.00	0.00	99.29	0.00	0.00
FG-OrIU	99.80	0.00	99.71	0.00	0.00	99.88	0.00	0.00	99.70	0.00	0.00

Table 10. Incremental Unlearning results for face recognition task on MS-Celeb-100. With four tasks, each forgetting 20 classes.

Methods	*Task 1*(100-20)		*Task 2*(80-20)			*Task 3*(60-20)			*Task 4*(40-20)		
	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$
Pre-train	62.60	66.55	61.53	61.25	-	61.67	63.56	-	61.49	63.92	-
L2*	64.17	0.00	64.31	3.80	15.70	68.05	4.75	26.29	64.97	2.58	21.42
EWC*	65.30	0.00	65.12	0.00	3.75	65.61	0.00	3.44	65.32	0.00	3.06
MAS*	64.30	0.00	63.82	0.00	0.00	63.15	0.00	0.25	62.47	0.00	0.64
LwF	66.51	0.00	68.62	0.00	0.00	68.61	0.18	0.17	69.34	0.00	0.17
DER	42.59	3.75	26.95	2.08	7.00	14.99	1.76	5.50	11.60	0.69	7.85
DER++	57.37	2.56	60.19	1.90	4.26	59.87	1.93	3.44	60.63	0.69	6.29
FDR	8.00	2.21	11.32	4.15	2.05	14.94	1.93	2.57	18.79	1.71	2.31
Retrain	14.27	0.00	13.60	0.00	0.00	14.08	0.00	0.00	14.65	0.00	0.00
BAD-T	44.53	0.00	45.90	3.10	0.00	43.97	2.90	0.00	35.44	2.60	0.00
LIRF*	34.67	26.70	35.29	25.43	0.00	31.09	18.47	0.00	30.28	11.08	0.00
SCRUB	48.58	0.00	47.86	4.15	0.00	46.45	26.41	0.00	46.44	29.90	0.75
SCRUB-S	67.87	0.00	67.52	0.00	0.00	66.05	25.70	0.00	66.09	51.89	5.20
GS-LoRA	66.61	0.00	68.06	0.00	0.00	69.89	0.17	0.00	70.78	0.00	0.05
GS-LoRA++	66.99	0.00	68.79	0.00	0.00	68.76	0.00	0.00	70.23	0.00	0.05
FG-OrIU	68.70	0.00	69.10	0.00	0.00	70.02	0.00	0.00	71.03	0.00	0.00

Table 11. Incremental Unlearning results for image classification task on CUB-200. With four tasks, each forgetting 20 classes.

method either fails to completely forget or significantly degrades the performance on the remaining classes. In contrast, our FG-OrIU method directly enforces constraints at both the feature and gradient levels for both forgetting and remaining classes, achieving thorough unlearning while effectively preventing feature entanglement between the two sets. Furthermore, Figures 6, 7, and 8, reveal that compared to the pre-trained model, our method enforces constraints on features in a way that results in a more compact decision boundary. This compactness reduces misclassifications, leading to slight improvements in certain cases over the original pre-trained model.

Methods	*Task 1*(100-20)		*Task 2*(80-20)			*Task 3*(60-20)			*Task 4*(40-20)		
	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$Acc_o \downarrow$
Pre-train	63.93	66.23	64.48	57.39	-	65.92	64.76	-	63.80	65.16	-
L2*	61.37	13.03	66.23	11.96	33.37	67.62	13.71	23.08	72.75	19.47	25.73
EWC*	54.34	1.70	61.52	4.82	2.00	62.85	4.80	4.97	65.04	7.63	6.65
MAS*	50.55	0.50	54.78	0.30	0.00	55.29	1.00	0.00	57.54	1.80	0.40
LwF	58.63	1.60	58.49	2.41	1.50	64.42	2.00	1.60	63.65	3.51	1.60
DER	46.36	15.43	33.59	9.65	19.84	25.53	6.81	12.69	20.23	8.13	9.58
DER++	59.68	12.73	62.40	7.94	11.62	66.29	7.41	7.98	68.60	12.55	7.45
FDR	1.60	0.30	2.73	3.11	0.60	6.62	2.60	1.00	12.46	0.90	1.50
Retrain	12.31	0.00	14.68	0.00	0.00	17.01	0.10	0.00	19.88	1.70	0.00
BAD-T	43.60	0.00	45.79	0.00	0.00	54.01	0.00	0.00	53.98	0.00	0.00
LIRF*	23.45	43.07	43.52	41.01	21.30	34.29	29.05	18.43	54.29	23.10	15.98
SCRUB	46.78	0.00	50.48	0.00	0.00	54.69	0.00	0.00	56.58	0.60	0.00
SCRUB-S	50.89	0.00	52.70	0.00	0.00	58.40	0.00	0.00	57.68	0.00	0.00
GS-LoRA	58.25	0.00	55.71	0.10	2.61	0.63	0.00	0.00	2.80	0.00	0.00
GS-LoRA++	58.89	0.60	55.53	0.20	1.90	55.49	0.00	0.45	55.23	0.00	1.40
FG-OrIU	60.01	0.00	56.97	0.10	0.00	60.17	0.00	0.00	61.64	0.00	0.00

Table 12. Incremental Unlearning results for image classification task on OmniBenchmark. With four tasks, each forgetting 50 classes.

Methods	100-1			100-5			100-10			100-20			100-40		
	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$
Pre-train	-	74.37	74.35	-	74.57	70.17	-	74.44	73.67	-	73.62	74.01	-	74.48	74.21
FG-OrIU	73.35	72.37	0.00	71.70	72.71	0.00	72.98	72.31	0.00	73.45	72.90	0.00	73.86	73.51	0.00
Methods	100-60			100-80			100-90			100-95			100-99		
	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$
Pre-train	-	74.77	74.11	-	74.47	74.32	-	74.58	74.34	-	72.26	74.48	-	70.48	74.41
FG-OrIU	74.03	73.95	0.00	74.54	74.76	0.00	75.72	77.15	0.00	72.66	70.92	0.00	85.33	100.00	0.00

Table 13. Static Machine Unlearning results for face recognition task on CASIA-Face100 with varying number of forgetting classes.

Methods	100-1			100-5			100-10			100-20			100-40		
	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$
Pre-train	-	89.35	96.0	-	89.18	94.00	-	89.47	89.00	-	88.91	90.02	-	90.03	88.50
FG-OrIU	91.60	87.58	0.00	89.69	85.75	0.00	88.11	87.24	0.00	88.45	86.93	0.00	86.24	84.10	0.00
Methods	100-60			100-80			100-90			100-95			100-99		
	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$	$H \uparrow$	$Acc_r \uparrow$	$Acc_f \downarrow$
Pre-train	-	89.65	89.27	-	92.80	88.50	-	93.00	89.00	-	94.80	89.14	-	88.00	89.41
FG-OrIU	86.79	84.45	0.00	86.77	85.10	0.00	86.95	85.00	0.00	92.44	96.00	0.00	94.41	100.00	0.00

Table 14. Static Machine Unlearning results for image classification task on Imagenet-100 with varying number of forgetting classes.

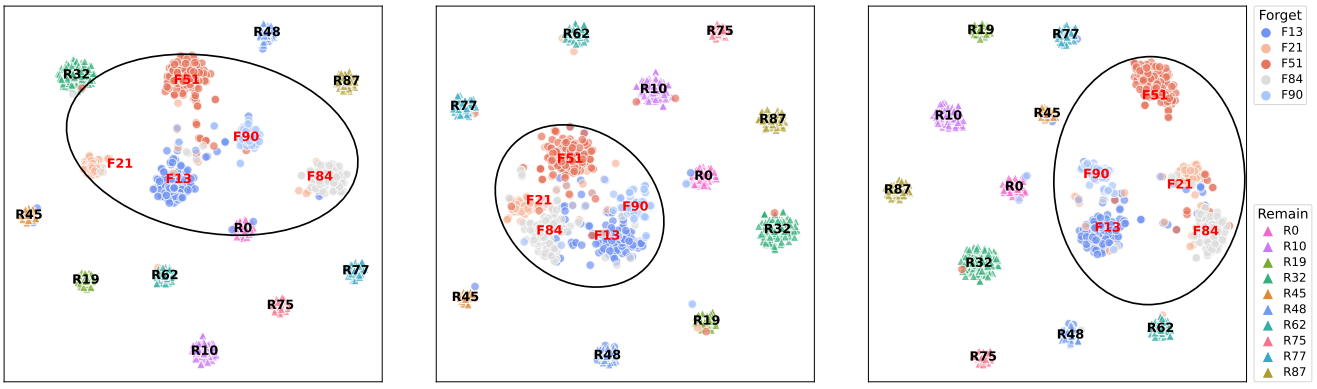
C. Algorithm

D. Details about DIP

D.1. DIP and Image Reconstruction

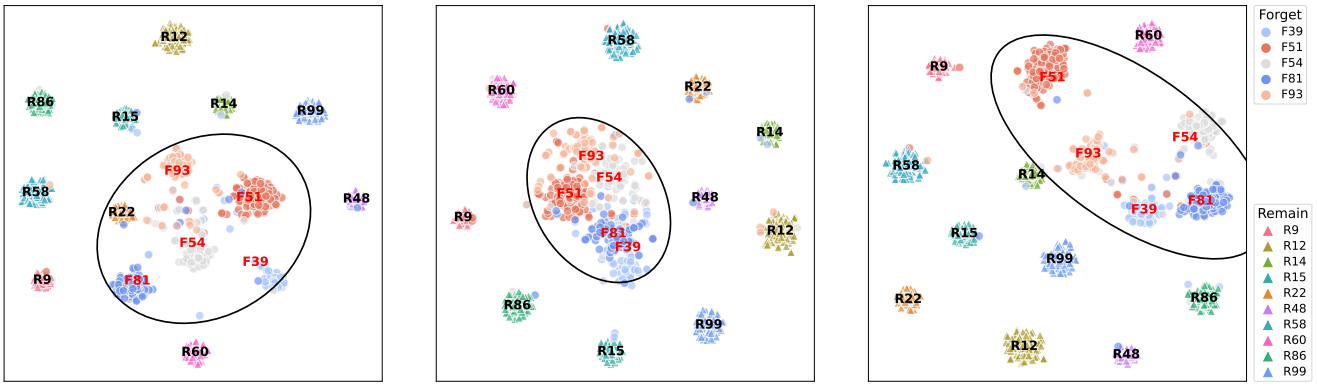
Deep Image Prior (DIP) [59] leverages the inherent structure of convolutional neural networks (CNNs) as an implicit prior for natural images by optimizing a randomly initialized CNN to reconstruct a target image from noise, progressively capturing structured information before overfitting to noise. This self-supervised approach exploits the inductive bias of CNNs without requiring external datasets or supervision, making it effective for image restoration tasks such as denoising, inpainting, and super-resolution [21, 23, 40, 47]. By aligning reconstructed images with feature representations from pre-trained models, DIP provides insights into the semantic information encoded in the extracted features.

In our study, we follow [77, 85] to investigate the semantic information encoded in the features extracted from



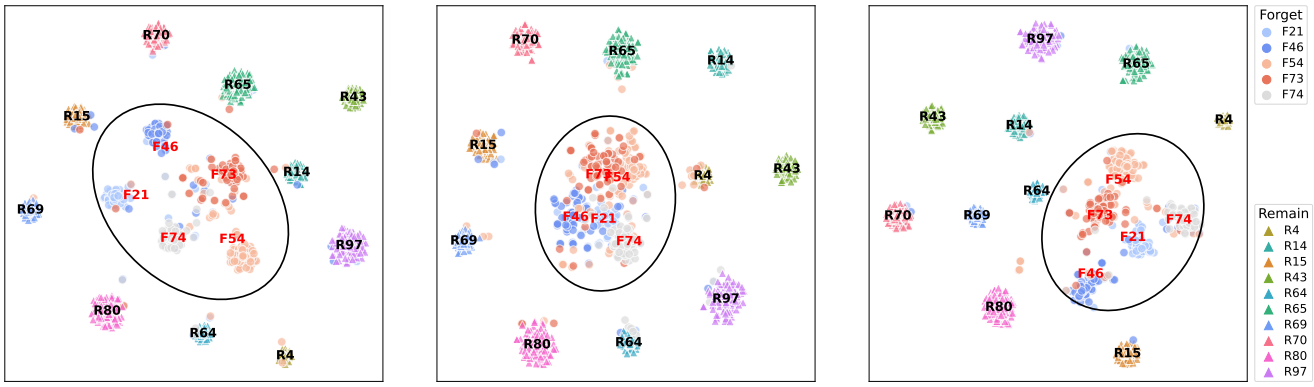
(a) Pre-train (b) Ours (c) GS-LoRA

Figure 6. T-SNE visualizations of the feature distributions for 5 forgetting classes and 10 remaining classes using (a) the Pre-trained model, (b) the unlearned model obtained through our FG-OrIU, and (c) GS-LoRA as feature extractors.



(a) Pre-train (b) Ours (c) GS-LoRA

Figure 7. T-SNE visualizations of the feature distributions for 5 forgetting classes and 10 remaining classes using (a) the Pre-trained model, (b) the unlearned model obtained through our FG-OrIU, and (c) GS-LoRA as feature extractors.



(a) Pre-train (b) Ours (c) GS-LoRA

Figure 8. T-SNE visualizations of the feature distributions for 5 forgetting classes and 10 remaining classes using (a) the Pre-trained model, (b) the unlearned model obtained through our FG-OrIU, and (c) GS-LoRA as feature extractors.

the last layers of the pre-trained model and the unlearned model. Specifically, we reconstruct images from features corresponding to forgetting classes in the unlearned model using DIP, then evaluate whether these reconstructions retain semantic attributes of the forgetting classes through (1) qualitative visual analysis (e.g., visualization of the reconstructed image) and (2) quantitative assessment using SSIM and PSNR metrics (the definition is as follows). These ap-

Algorithm 1 Pseudo code of Feature Subspace Decomposition and Dynamic Subspace Adaptation

Input: Pre-trained model $\mathcal{M}$ and its layers N, the Pre-trained model's function $f_{\mathcal{M}}$, LoRA module $\boldsymbol{\theta}$, a sequence of unlearning task $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_T\}$, forgetting classes $\mathbf{D}_{f,t}$ and remaining classes $\mathbf{D}_{r,t}$ for each unlearning task, threshold ϵ, hyper-parameter λ_1 and λ_2, learning rate η

for t in $\{1, \dots T\}$ **do**
  if $t = 1$ **then**
    Stage One: Feature Subspace Decomposition
    for l in $\{1, \dots, N\}$ **do**
      build forgetting feature subspace
      $\boldsymbol{R}^l_{f,1} \leftarrow f_{\mathcal{M}}(\mathbf{D}_{f,1})$
      $\boldsymbol{U}^l_{f,i}\boldsymbol{\Sigma}^l_{f,i}(\boldsymbol{V}^l_{f,i})^T \leftarrow \text{SVD}(\boldsymbol{R}^l_{f,1})$
      $\boldsymbol{B}^l_{f,1} \leftarrow \boldsymbol{U}^l_{f,i}[:k]$
      $\mathcal{S}^l_{f,1} = \text{span}\{\boldsymbol{B}^l_{f,1}\}$
      build remaining feature subspace
      $\boldsymbol{R}^l_{r,1} \leftarrow f_{\mathcal{M}}(\mathbf{D}_{r,1})$
      $\boldsymbol{U}^l_{r,i}\boldsymbol{\Sigma}^l_{r,i}(\boldsymbol{V}^l_{r,i})^T \leftarrow \text{SVD}(\boldsymbol{R}^l_{r,1})$
      $\boldsymbol{B}^l_{r,1} \leftarrow \boldsymbol{U}^l_{r,i}[:k]$
      $\mathcal{S}^l_{r,1} = \text{span}\{\boldsymbol{B}^l_{r,1}\}$
    end for
  else
    Stage Three: Dynamic Subspace Adaptation
    update forgetting feature subspace
    $\mathcal{S}^l_{f,i} \leftarrow$ via Eq.(11)
    update remaining feature subspace
    $\mathcal{S}^l_{r,i} \leftarrow$ via Eq.(11)
  end if
  Stage Two: Dual Orthogonal Projection
  for each batch $(\boldsymbol{x}_f, \boldsymbol{x}_r)$ in $(\mathbf{D}_{f,t}, \mathbf{D}_{r,t})$ **do**
    feature orthogonal projection
    $\mathcal{L}_{fop-f} \leftarrow$ via Eq.(4)
    $\mathcal{L}_{fop-r} \leftarrow$ via Eq.(7)
    $\mathcal{L}_{ce} \leftarrow$ via Eq.(5)
    $\mathcal{L}_{total} \leftarrow$ via Eq.(8)
    for l in $\{1, \dots, N\}$ **do**
      gradient orthogonal projection
      $\boldsymbol{g_l} \leftarrow \nabla_{\theta_l}\mathcal{L}_{total}$
      $\hat{\boldsymbol{g}}_{\boldsymbol{l}} \leftarrow$ via Eq.(9)
      $\boldsymbol{\theta}_l \leftarrow \boldsymbol{\theta}_l - \eta\hat{\boldsymbol{g}}_{\boldsymbol{l}}$
    end for
  end for
end for

proaches enable systematic evaluation of whether features in unlearned models retain traces of semantic information from intentionally forgotten classes.

SSIM (Structural Similarity Index)

The Structural Similarity Index (SSIM) measures the similarity between two images. It is defined as follows:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{14}$$

where x and y are two images (or image patches) being compared. μ_x and μ_y are the mean intensities of x and y, respectively:

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i \tag{15}$$

where σ_x^2 and σ_y^2 are the variances of x and y:

$$\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2, \quad \sigma_y^2 = \frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2 \tag{16}$$

where σ_{xy} is the covariance between x and y:

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{17}$$

where C_1 and C_2 are small constants to stabilize the division, typically defined as:

$$C_1 = (K_1 L)^2, \quad C_2 = (K_2 L)^2 \tag{18}$$

where L is the dynamic range of pixel values (e.g., 255 for an 8-bit image), and K_1 and K_2 are small constants, commonly set to 0.01 and 0.03, respectively.

PSNR (Peak Signal-to-Noise Ratio)

The Peak Signal-to-Noise Ratio (PSNR) is a metric used to measure the quality of an image compared to a reference. It is defined as:

$$PSNR = 10\log_{10}\left(\frac{L^2}{MSE}\right) \tag{19}$$

where: MSE (Mean Squared Error) is calculated as:

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(x_{i,j} - y_{i,j})^2 \tag{20}$$

where $x_{i,j}$ and $y_{i,j}$ are the pixel values of the reference and distorted images, respectively. And L is the maximum possible pixel value (e.g., 255 for an 8-bit image).

D.2. Implementation of DIP

We follow [77, 85] and the open-source implementation [5] to implement DIP. Specifically, we use an autoencoder as the trainable network, with an encoder composed of four convolutional layers with increasing channel depths (64, 128,

[5] https://github.com/atiyo/deep_image_prior

256, and 512), each followed by a ReLU activation, and a decoder symmetric to the encoder structure but utilizing transposed convolutional layers. The final layer applies a Sigmoid activation to constrain pixel values between 0 and 1, ensuring a proper reconstructed image output.

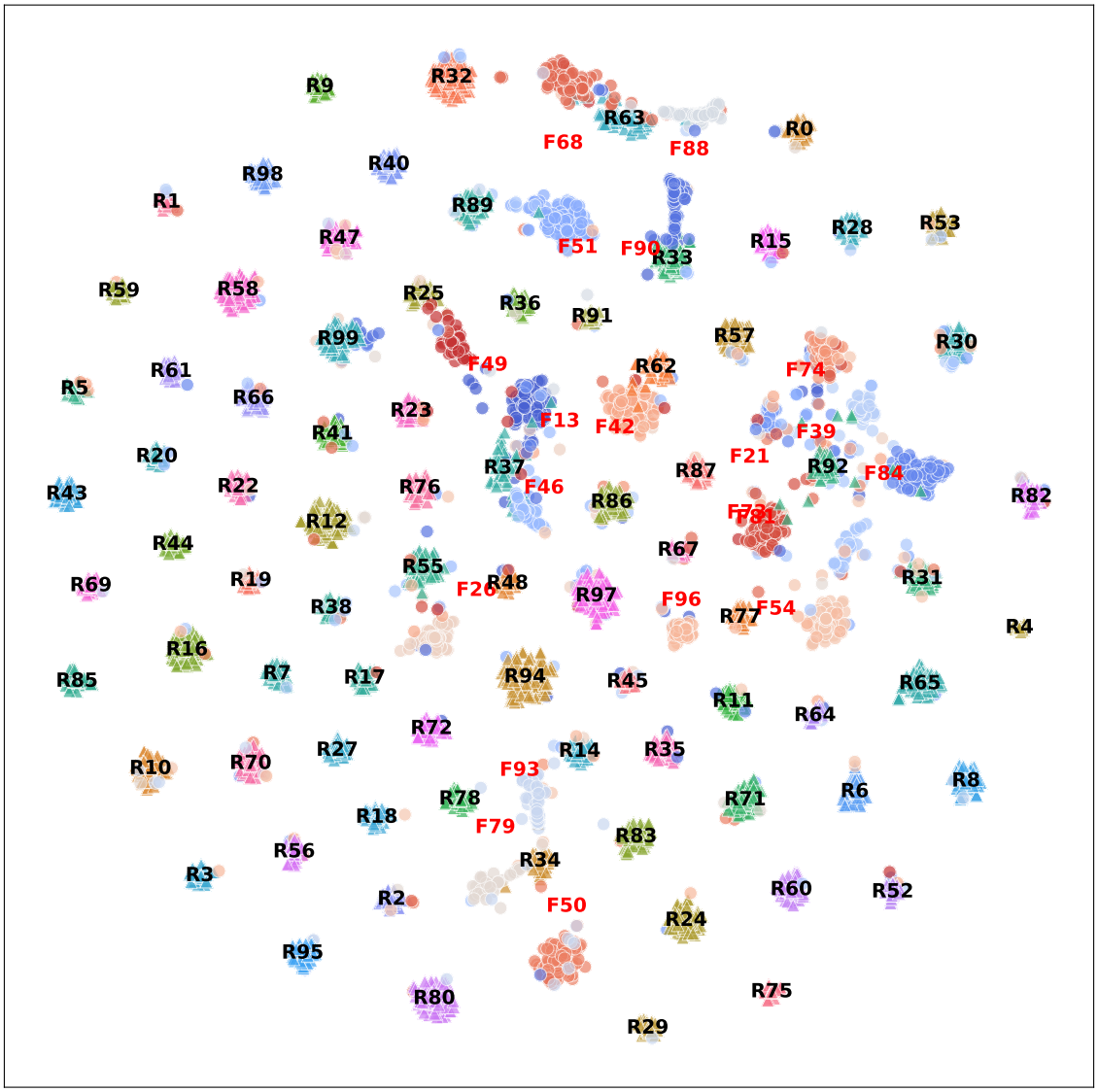
During training, we use the unlearned model $f_{\mathcal{M}}$ to extract the feature from a sample belonging to the forgetting class as the training target, $f_{\mathcal{M}}(\boldsymbol{x}_f)$. We initialize a fixed noise $\boldsymbol{x}_{noise}$ with the same size as the sample (or resize it accordingly) and feed the noise into the AE network to obtain the output $f_{AE}(\boldsymbol{x}_{noise})$. We then extract features from $f_{AE}(\boldsymbol{x}_{noise})$ using the unlearned model $f_{\mathcal{M}}(\boldsymbol{x}_f)$ to obtain $f_{\mathcal{M}}(f_{AE}(\boldsymbol{x}_{noise}))$. Finally, the AE network is updated by minimizing the following loss function:

$$\mathcal{L}_{recon} = ||f_{\mathcal{M}}(f_{AE}(\boldsymbol{x}_{noise})) - f_{\mathcal{M}}(\boldsymbol{x}_f)||_F^2 \tag{21}$$

The reconstructed image is output every fixed number of iterations. We obtain the unlearned model using different IU methods and reconstruct two examples, with the reconstruction process illustrated in Figure 11 and Figure 12, respectively. In addition, we provide SSIM and PSNR metrics during the reconstruction process to quantitatively evaluate the reconstructed images.

D.3. More Visualization Results

Here, we provide additional visualization results of DIP on both samples from forgetting classes and remaining classes. We select two samples from each of the four forgetting classes and two samples from each of the four remaining classes. The DIP reconstruction results are presented in Figure 13 and Figure 14, respectively.



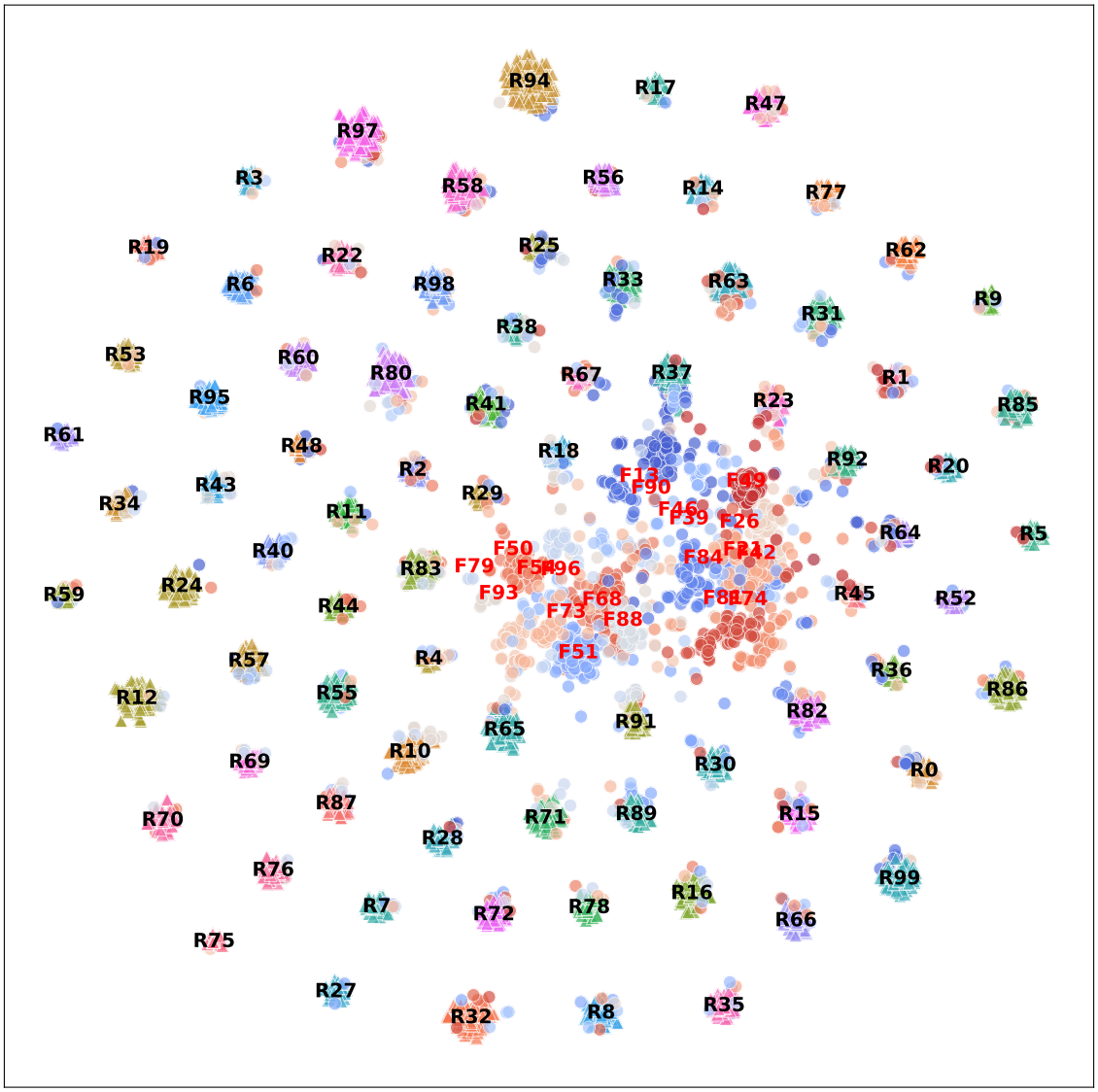
Forget Classes

F13	F26	F42	F49	F51	F68	F74	F81	F88	F93
F21	F39	F46	F50	F54	F73	F79	F84	F90	F96

Remain Classes

R0	R8	R17	R27	R35	R45	R58	R66	R77	R89
R1	R9	R18	R28	R36	R47	R59	R67	R78	R91
R2	R10	R19	R29	R37	R48	R60	R69	R80	R92
R3	R11	R20	R30	R38	R52	R61	R70	R82	R94
R4	R12	R22	R31	R40	R53	R62	R71	R83	R95
R5	R14	R23	R32	R41	R55	R63	R72	R85	R97
R6	R15	R24	R33	R43	R56	R64	R75	R86	R98
R7	R16	R25	R34	R44	R57	R65	R76	R87	R99

Figure 9. T-SNE visualizations of the feature distributions for 20 forgetting classes and 80 remaining classes using the unlearned model obtained through GS-LoRA as feature extractors.



Forget Classes

F13	F26	F42	F49	F51	F68	F74	F81	F88	F93
F21	F39	F46	F50	F54	F73	F79	F84	F90	F96

Remain Classes

R0	R8	R17	R27	R35	R45	R58	R66	R77	R89
R1	R9	R18	R28	R36	R47	R59	R67	R78	R91
R2	R10	R19	R29	R37	R48	R60	R69	R80	R92
R3	R11	R20	R30	R38	R52	R61	R70	R82	R94
R4	R12	R22	R31	R40	R53	R62	R71	R83	R95
R5	R14	R23	R32	R41	R55	R63	R72	R85	R97
R6	R15	R24	R33	R43	R56	R64	R75	R86	R98
R7	R16	R25	R34	R44	R57	R65	R76	R87	R99

Figure 10. T-SNE visualizations of the feature distributions for 20 forgetting classes and 80 remaining classes using the unlearned model obtained through our FG-OrIU as feature extractors.

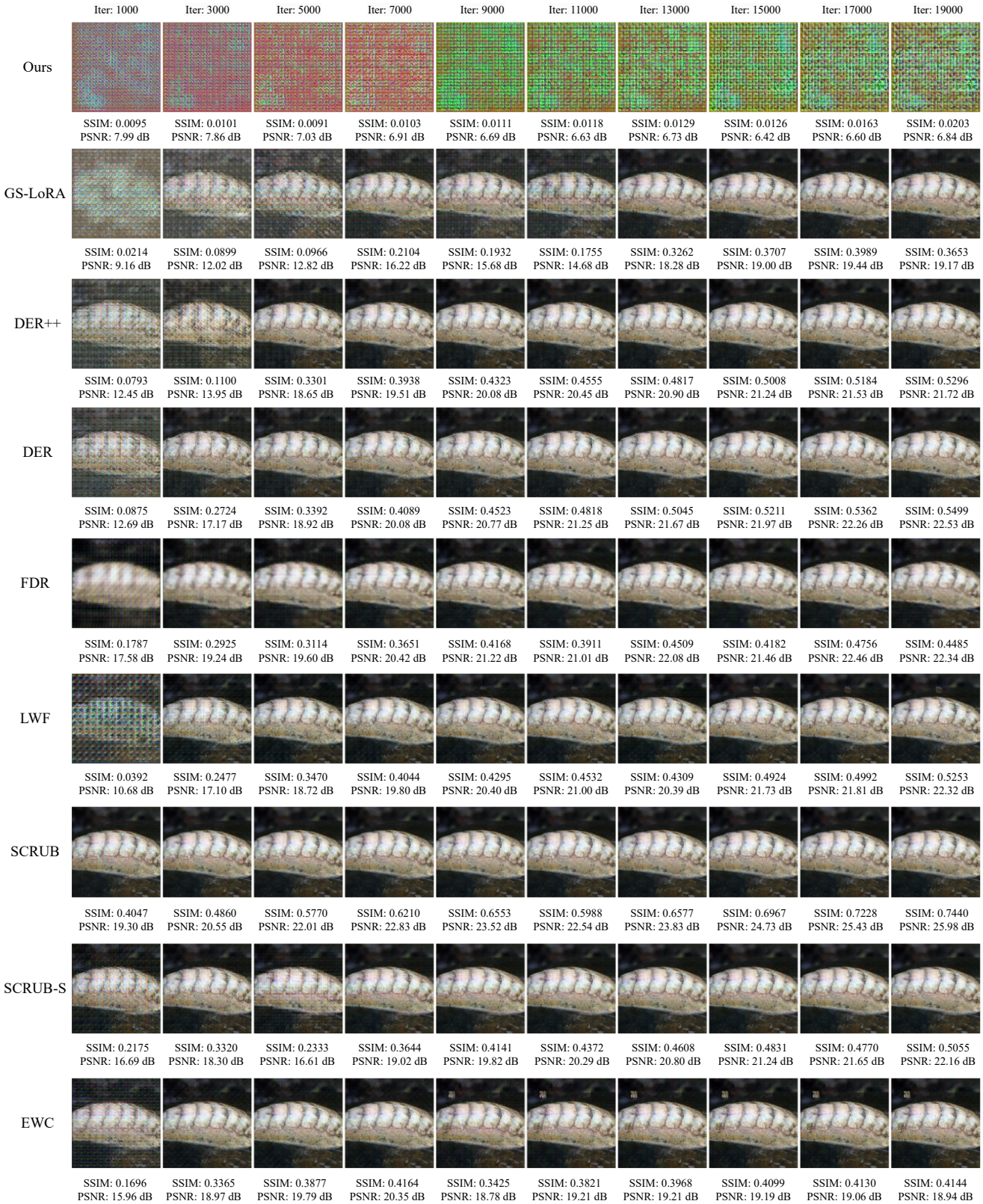


Figure 11. The image reconstruction process of example 1 from forgetting classes.

Figure 12. The image reconstruction process of example 2 from forgetting classes.

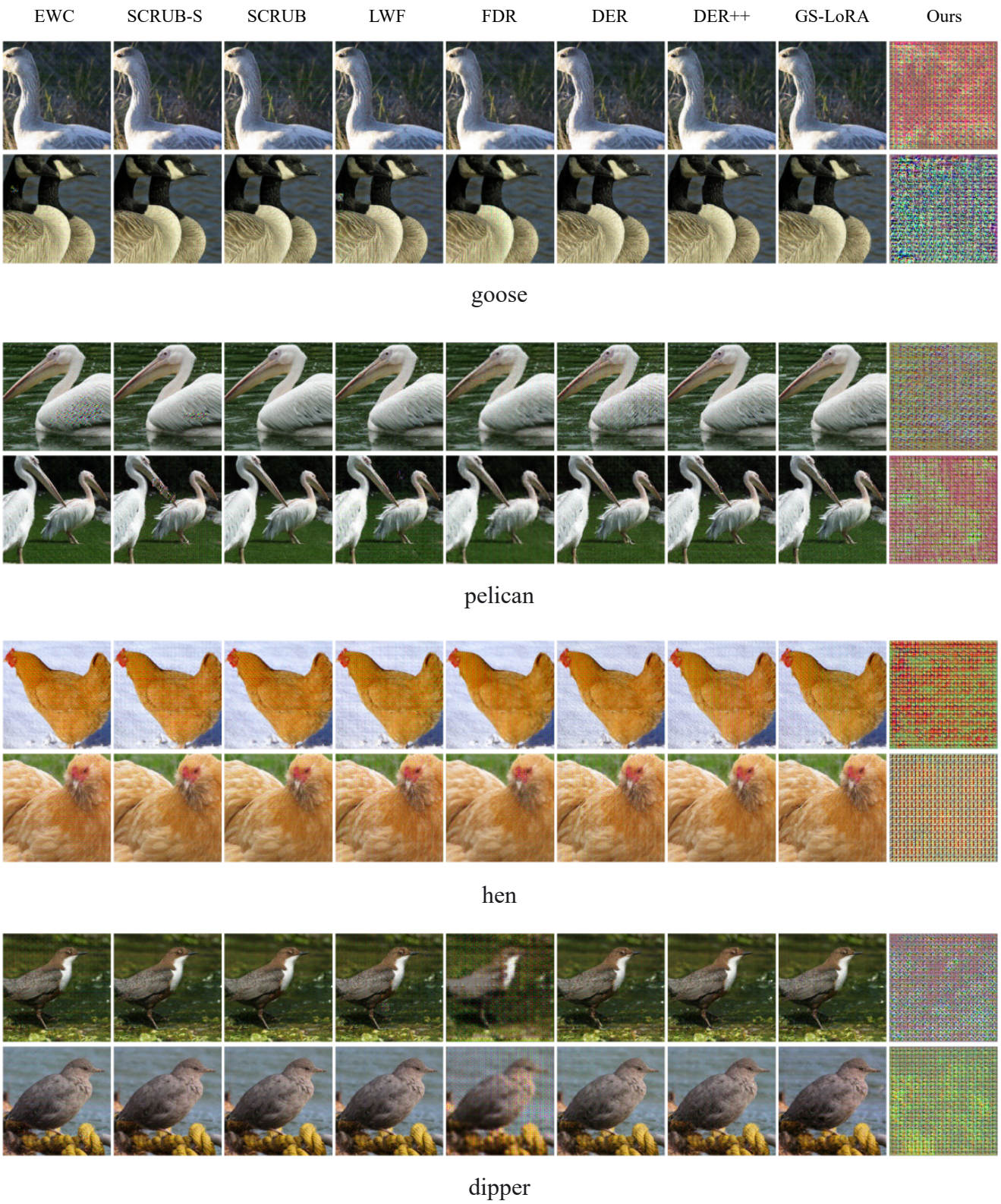


Figure 13. The image reconstruction of forgetting classes' samples.

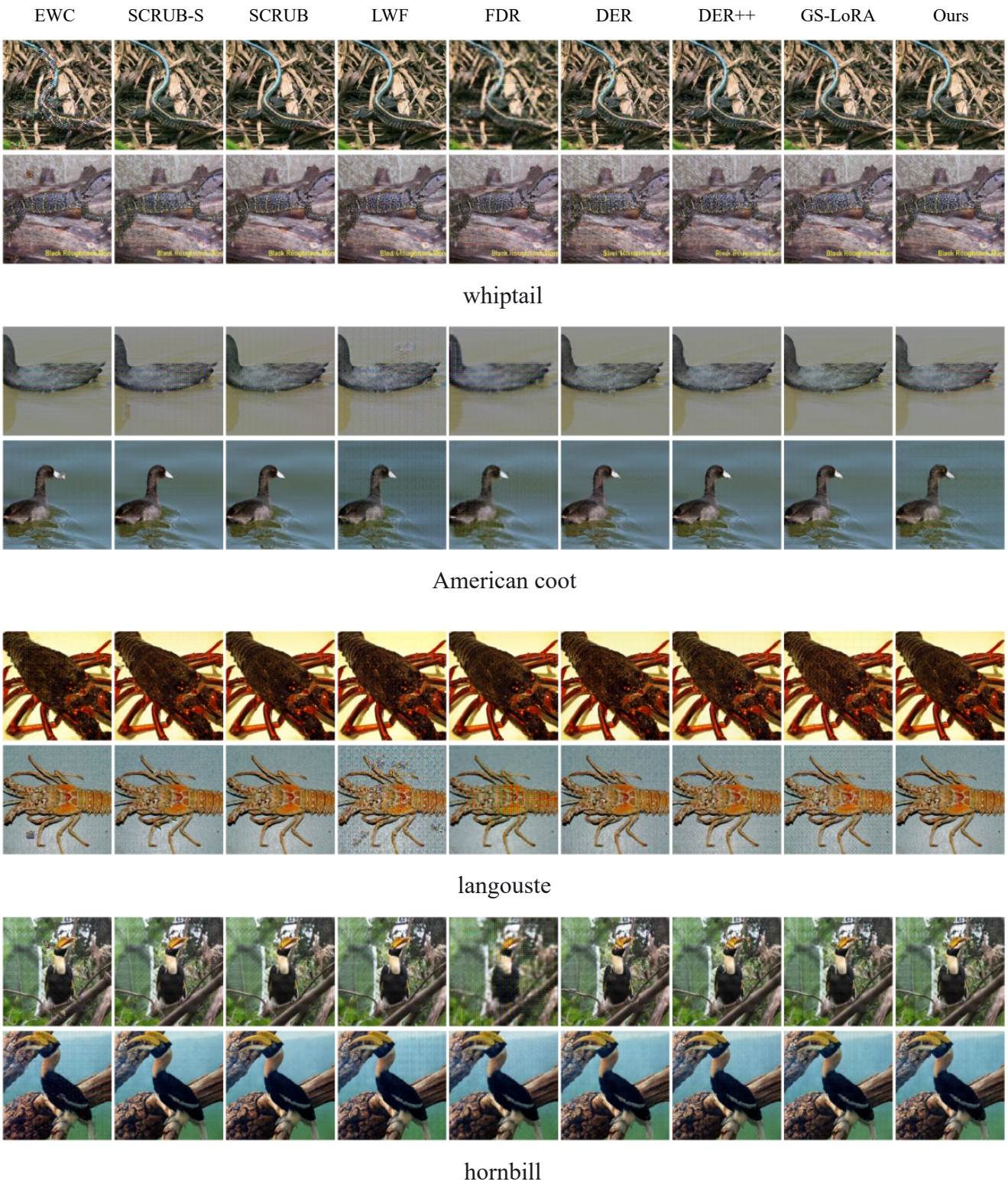


Figure 14. The image reconstruction of remaining classes' samples.