Appendix

A. Algorithms

In this section, we provide the pseudo-codes for our proposed privacy attack, the Reminiscence Attack (ReA) (described in Section 3), and the unlearning framework Orthogonal Unlearning & Replay (OUR) (defined in Section 4).

Algorithm 1 Class-wise *Reminiscence Attack* (ReA)

Require: OOD dataset $\mathcal{D}^{\text{ood}}$, reference OOD set $\mathcal{D}'$, assigned unlearned label y_u, loss function $\mathcal{L}$
Parameters: Max iterations $\text{Idx}_{\max}$, learning rates $\{\text{lr}_j\}_{j=1}^{J}$
if Black-Box Scenario **then**
  $\theta'_0 \leftarrow \text{ModelExtract}(F(\cdot;\theta^u))$ {Model extraction}
  $F_{\text{ini}} \leftarrow \text{Copy}(F(\cdot;\theta'_0))$ {Save initial model}
else if White-Box Scenario **then**
  $\theta'_0 \leftarrow \theta^u$ {Direct access}
end if
for $\text{lr}_j \in \{\text{lr}_j\}_{j=1}^{J}$ **do**
  Initialize $\theta' \leftarrow \theta'_0$
  for $i = 1$ **to** $\text{Idx}_{\max}$ **do**
    Compute loss: $L = \frac{1}{|\mathcal{D}^{\text{ood}}|}\sum_{x\in\mathcal{D}^{\text{ood}}}\mathcal{L}(F(x;\theta'),y_u) + \frac{1}{|\mathcal{D}'|}\sum_{x\in\mathcal{D}'}\mathcal{L}(F(x;\theta'),F_{\text{ini}}(x))$
    $\theta' \leftarrow \theta' - \text{lr}_j \cdot \nabla_{\theta'} L$ {SGD update}
    if $\text{Acc}_{\mathcal{D}^{\text{ood}}}(F(\cdot;\theta')) > 0.9$ **then**
      Record $\text{Idx}_{\text{r}}(\mathcal{D}^{\text{ood}},\text{lr}_j) \leftarrow i$
      break
    end if
  end for
end for
Compute confidence score: $\mathcal{A}'(\mathcal{D}^{\text{ood}}) = 1 - \frac{1}{J\cdot\text{Idx}_{\max}}\sum_{j=1}^{J}\text{Idx}_{\text{r}}(\mathcal{D}^{\text{ood}},\text{lr}_j)$ Equation 6
Output: Confidence score $\mathcal{A}'(\mathcal{D}^{\text{ood}})$

Algorithm 2 Sample-wise *Reminiscence Attack* (ReA)

Require: Inference dataset $\mathcal{D}_{\text{infer}}$, victim model $F(\cdot;\theta^u)$ {i.e., the unlearned model}
Parameters: Pseudo retain set size N_r, training epochs EPOCHS $= 5$, learning rate lr
Obtain confidence scores: $\{s\} \leftarrow \mathcal{A}'(\mathcal{D}_{\text{infer}})$ {Using MIA-LiRA method}
Select top-N_r indices $\{\text{ind}_i\}_{i=1}^{N_r}$ from $\{s\}$ based on confidence
Construct pseudo retain set: $\mathcal{D}'_r \leftarrow \mathcal{D}_{\text{infer}}[\{\text{ind}_i\}_{i=1}^{N_r}]$ {Subset selection}
for $e = 1$ **to** EPOCHS **do**
  Compute loss: $L = \frac{1}{|\mathcal{D}'_r|}\sum_{(x,y)\in\mathcal{D}'_r}\mathcal{L}(F(x;\theta'),y)$
  $\theta' \leftarrow \theta' - \text{lr}\cdot\nabla_{\theta'}L$ {SGD update}
end for
Output: Updated confidence scores $\{\mathcal{A}'(F(x;\theta'))\}_{x\in\mathcal{D}_{\text{infer}}}$

Algorithm 3 *Orthogonal Unlearning & Replay* (OUR) Framework

Require: Original model $F(\cdot;\theta)$, full dataset $\mathcal{D}$, unlearning set $\mathcal{D}_u$
Parameters: Phase epochs e_1, e_2, norm threshold Δ_{thr}, learning rate lr
Compute remaining set: $\mathcal{D}_r \leftarrow \mathcal{D} \setminus \mathcal{D}_u$
Store initial parameters: $\theta^{\text{ini}} \leftarrow \theta = [\theta_1, \cdots]$

Phase 1: The Orthogonal Unlearning

for $e = 1$ **to** e_1 **do**
  Compute orthogonal loss: $L_{\text{orth}} = \sum_{(x,y)\in\mathcal{D}_u}\sum_{l=1}^{k}\|F_l(x;\theta) - F_l(x;\theta^0)\|_2^2$
  $\theta \leftarrow \theta - \text{lr}\cdot\nabla_\theta L_{\text{orth}}$ {Orthogonality Update}
  Compute the maximum parameter change: $\Delta_{\max} \leftarrow \max_{i\in|\theta|}\frac{\|\theta_i - \theta_i^{\text{ini}}\|_2}{|\theta_i|}$
  if $\Delta_{\max} > \Delta_{\text{thr}}$ **then**
    break {Early stopping}
  end if
end for

Phase 2: The Replay Phase

for $e = 1$ **to** e_2 **do**
  Compute loss: $L = \frac{1}{|\mathcal{D}_r|}\sum_{(x,y)\in\mathcal{D}_r}\mathcal{L}(F(x;\theta),y)$
  $\theta \leftarrow \theta - \text{lr}\cdot\nabla_\theta L$ {Retain knowledge}
end for
Output: Unlearned model $F(\cdot;\theta^u)$ where $\theta^u \leftarrow \theta$

B. The Proportion of Stable Neurons after Orthogonal Unlearning

Orthogonal unlearning induces rapid utility degradation by excluding the retained dataset, yet model functionality recovers efficiently in the replay phase due to minimal parameter perturbations. To validate this, we empirically analyze neuron stability in a ViT model trained on CIFAR-20 after class-wise unlearning.

We define the change for each parameter as follows. Since parameters vary in size, we define the change of parameter θ_i as:

$$\Delta(\theta_i) = \frac{\|\theta_i - \theta_i^{\text{ini}}\|_2}{|\theta_i|}, \tag{8}$$

where θ_i^{ini} represents the parameter's state before unlearning. To account for differences in parameter dimensions, we normalize the change by its size $|\theta_i|$ for better observation. The maximum parameter change is then defined as:

$$\Delta_{\max} = \max_{i\in|\theta|}\frac{\|\theta_i - \theta_i^{\text{ini}}\|_2}{|\theta_i|} \tag{9}$$

Figure 7 reports that parameter changes after unlearning remain highly localized, with magnitudes concentrated below 0.0005, which is three times smaller than the most minor changes observed in random models. Further analysis in

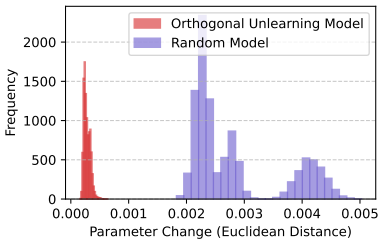


Figure 7. Histograms of Parameter Changes.

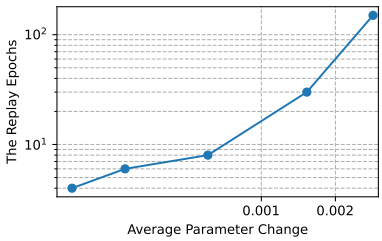
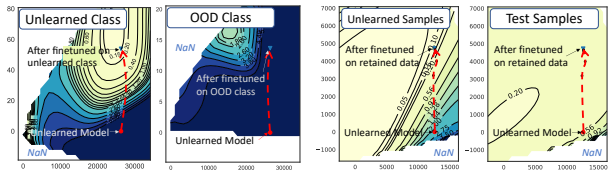


Figure 8. Parameter Changes versus The replay speed measured by epochs.



(a) Class-wise Unlearn (RUM [48]) (b) Sample-wise Unlearn (GA [40])

Figure 9. Visualized loss landscapes of unlearned data and non-training data around unlearned models in CIFAR100 and CIFAR10 experiments.

Figure 8 confirms that smaller parameter changes correlate strongly with faster utility recovery during replay, which requires fewer training epochs. These results demonstrate that orthogonal unlearning preserves critical neural patterns despite utility drops, enabling efficient restoration through targeted replay rather than full retraining.

C. Supplementary Visualized Loss Landscape

We extend Figure 2 to CIFAR-10/100-ResNet18 in Figure 9, confirming that residuals, i.e., discrepancies between unlearned and non-training data, consistently appear across architectures. These are reflected in fine-tuning trajectories (dashed lines): in class-wise unlearning (Figure 9a), the loss of unlearned classes converges to sharp minima while OOD classes remain unconverged; in sample-wise unlearning (Figure 9b), unlearned samples exhibit sharp loss drops while test losses stay stable. These differences expose membership privacy.

D. Experimental Setups and Approximate Unlearning Benchmarks

D.1. Setups

ReA Setups. For class-wise ReA attacks, the ratio of reference data for logits constraints to inferred data is set to $14:1$ for CIFAR20 and $6:1$ for CIFAR100 during reminiscence. The maximum number of training iterations Idx_{max} is set to 75. We employ cross entropy loss $\mathcal{L}$ during its reminiscence process, with SGD with momentum (0.9) and weight decay 5e^{-4}. In sample-wise ReA, the size of "pseudo" retained dataset $\mathcal{N}_r$ is 20,000.

OUR Unlearning Method Setups. For the implementation of OUR, we adopt the following hyper-parameter configurations. The training process consists of distinct phases tailored for different architectures.

For ResNet18 experiments, the orthogonal unlearning and replay phases each span 8 epochs, with learning rates of 0.0018 and 0.005, respectively; a 0.5 decay is applied at epoch 3 during replay phases. For ViT experiments, the unlearning phase lasts 7 epochs. The replay phase runs for 7 epochs for class-wise unlearning or 11 epochs for sample-wise unlearning, with learning rates of 0.0008 and 0.0004, and decay factors of 0.5 and 0.2 at epochs 4 and 6. The ablation study on the number of epochs in OUR is provided in Appendix E.2. Besides, the regularization component involves an l_1 regularization factor of 1e^{-5} to enhance model sparsity [25].

Implementation Details. All classification experiments are conducted using PyTorch on 2×RTX 4090 GPUs, and diffusion model experiments are performed on 4×RTX 4090 GPUs.

D.2. Approximate Machine Unlearning Benchmarks

- **Fine-tuning (FT)** [25, 45]. It leverages the phenomenon of catastrophic forgetting to continue training the pre-trained model $F(.;\theta)$ on the remaining dataset $\mathcal{D}_r$ for a limited number of iterations, to forget the unlearned dataset $\mathcal{D}_u$ which is no longer trained upon.
- **Gradient Ascent (GA)** [18, 40]. This approach seeks to achieve unlearning by reversing the gradient descent process on $\mathcal{D}_u$, thereby erasing its optimization traces in the model $F(.;\theta)$.
- **Random Label (RL)** [10, 18]. It assigns random labels for $\mathcal{D}_u$ to erase the model $F(.;\theta)$'s memory of them.
- **Influence Unlearning(IU)** [25]. It utilizes the woodfisher method. It estimates the influence of $\mathcal{D}_u$ on $F(.;\theta)$ and designs perturbation strategies to erase this influence from the parameters θ.
- **FisherForgetting (FF)** [17]. FF perturbs θ with additive Gaussian noise, where the covariance is derived from the fourth root of the Fisher Information matrix over $\mathcal{D}_r$.

While theoretically rigorous, its reliance on Fisher matrix inversion limits parallel efficiency and increases computational overhead compared to gradient-based methods.

- **Boundary-based Unlearning (BU)** [8]. BU shifts the decision boundary of $F(.;\theta)$ to mimic a retrained model's behavior, thereby bypassing parameter-space optimization.

Additionally, there are five optimized unlearning frameworks.

- $l1$ **sparsity** [25]. This framework bridges approximate and exact unlearning by pruning non-critical weights. Sparsity reduces the parameter space, thus it narrows the gap between approximate and ideal unlearning outcomes while maintaining efficiency.
- **SalUn** [13]. It introduces weight saliency to focus unlearning efforts on critical parameters. Analogous to input saliency in explainable AI, it prioritizes weights with high influence on $\mathcal{D}_u$.
- **SCRUB** [28]. It uses a teacher-student architecture where the student selectively disregards the teacher's knowledge about $\mathcal{D}_u$. This "unlearning-by-disobedience" approach scales without restrictive assumptions.
- **SFRon** [24]. It unifies gradient-based MU by decomposing updates into three components: forgetting gradient ascent, retaining gradient descent, and saliency-guided weighting. It further incorporates a Hessian-aware manifold geometry to align unlearning trajectories with the output probability space to balancing utility performance and forgetting efficacy.
- **Refined-Unlearning Meta-algorithm (RUM)** [48]. It refines the forget set $\mathcal{D}_u$ into homogeneous subsets and applies specialized unlearning strategies to each. Its meta-algorithm orchestrates existing methods to comprehensively unlearn $\mathcal{D}_u$.

E. Complementary Experimental Results

E.1. Ablation Study of Layers $\{l\}_k$ Seleted in OUR

To determine the optimal layer configuration for orthogonal unlearning (OUR), we conduct an ablation study evaluating three distinct layer-selection strategies for l_k: using output layers from the first three transformer blocks (**First 3**), the last three blocks (**Last 3**), and a distributed set comprising the first, middle, and final blocks (**Span 3**). Experiments on CIFAR-20 evaluate unlearning performance through ToW of Unlearning and resistance against relearning attacks through ReA Accuracy, where values approaching 1.0 and 50% respectively indicate optimal outcomes.

As shown in Figure 10, the Span 3 configuration achieves the most favorable balance across both evaluation metrics for class-wise and sample-wise OUR implementations, which attains near-optimal ToW while maintaining ReA accuracy closest to random-guess performance.

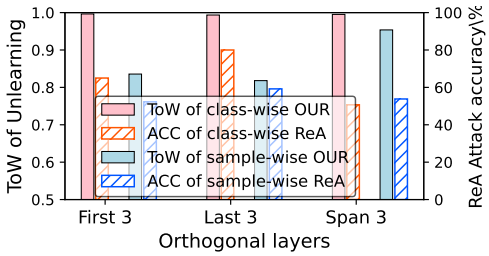


Figure 10. OUR performance with different $\{l\}_k$ settings: first 3 / last 3 / span 3 denote the output layers of the first 3 blocks, last 3 blocks, and first, middle, and last blocks, respectively.

E.2. Ablation Study of Number of Epochs in OUR

To optimize the training efficiency and unlearning effectiveness of OUR unlearning, we conduct an ablation study examining epoch configurations for its two-phase training: orthogonal unlearning ($e1$) and replay ($e2$). We evaluate three critical metrics across five repeated trials on CIFAR20-ViT, CIFAR10-ResNet18, and CIFAR100-ResNet18: unlearning performance (ToW of Unlearning), and resistance against relearning attacks (ReA Attack Accuracy).

As shown in Figure 11, optimal configurations emerge for each setting: CIFAR20-ViT requires 2 $e1$ and 7 $e2$ for sample-wise and 7 $e1$ and 11 $e2$ for class-wise OUR; CIFAR10-ResNet18 requires 8 $e1$ and 8 $e2$ for sample-wise OUR; CIFAR100-ResNet18 requires 8 $e1$ and 8 $e2$ for class-wise OUR. These configurations simultaneously achieve near-optimal ToW ($\approx$ 1.0) and strong ReA resistance while minimizing RTE.

E.3. Ablation Study of ReA Learning Rate Set

The learning rate (lr) during the reminiscence process largely impacts class-wise ReA efficacy, as it directly affects the scores ($\mathcal{A}'$ in Equation 3) for membership detection. In contrast, sample-wise ReA remains robust to lr variations, requiring only a minimal lr of $0.1\times$ the training lr for model convergence. Here, we evaluate whether our multi-lr aggregation strategy (Section 3.1) enhances robustness to this parameter in class-wise ReA.

Figure 12 compares resonance differences (the difference of resonance index between OOD and unlearned classes, $\Delta\text{Idx}_\text{r}$) for all MU methods at different lr on CIFAR-20, which is presented as bars. The left y-axis shows resonance differences $\Delta\text{Idx}_\text{r}$ (positive values indicate faster convergence for unlearned classes), averaged over 10 trials. The right y-axis contrasts ReA privacy attack accuracy under single-lr (blue lines) and multi-lr aggregation (red dashed line) strategies. **Key observations.** First, single lr$=0.001$ maximizes ReA effectiveness for most MU methods except GA [40] and its variant, SFRon [24], which indicates that optimal lr ranges are dependent on MU methods. Second, ReA's performance collapses at lr$=0.1$. Third,

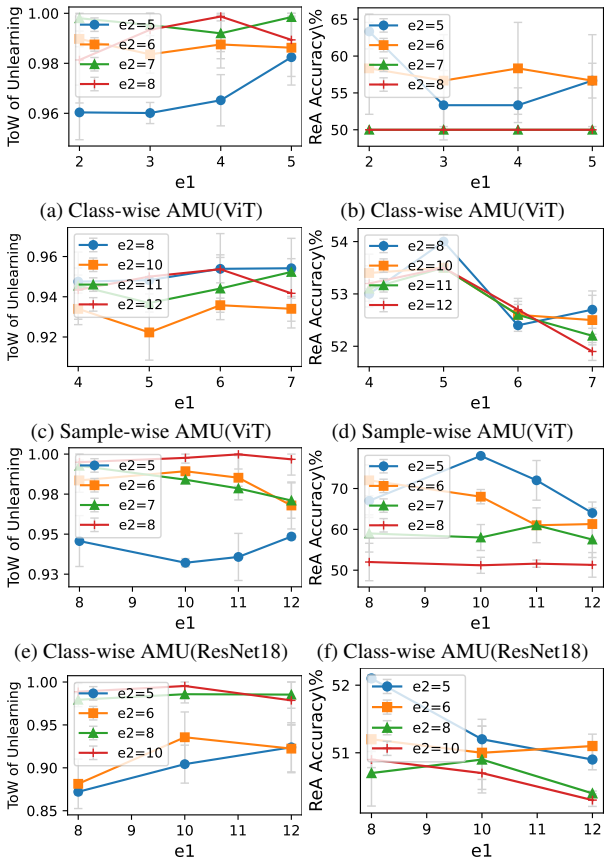


Figure 11. Ablation study of e_1 and e_2 in OUR. Gray vertical bars indicate error bars. Class-wise ReA is evaluated 10 times for both unlearn and OOD classes per experiment. Results are averaged over 5 independent trials.

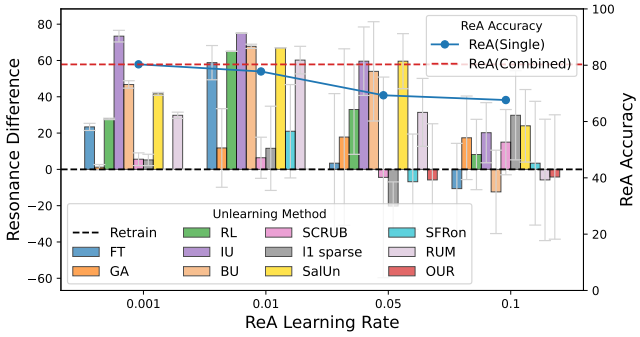


Figure 12. Resonance difference (Idx_r of OOD classes minus that of unlearned classes) and ReA attack accuracy across learning rates in ReA. Gray bars represent error bars.

the multi-lr aggregation strategy achieves superior results without method-specific tuning. Notably, our OUR method exhibits near-zero resonance difference, demonstrating inherent resistance to ReA attacks.

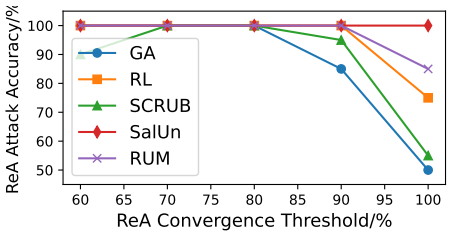


Figure 13. ReA Attack Accuracy versus Convergence Threshold

E.4. Ablation Study of ReA Convergence Threshold

To evaluate how convergence threshold affects ReA performance, we conduct an ablation study on this key parameter. The convergence threshold represents the prediction accuracy threshold in class-wise ReA to determine the resonance index. Experiments perform on CIFAR-20 and evaluate the ReA performance targeting five selected unlearning benchmarks: GA, RL, SCRUB, SalUn, and RUM. The results in Figure 13 confirm that thresholds between 70% and 80% yield optimal ReA attack accuracy across all evaluated unlearning methods.

E.5. The Complete ROC Analysis of Privacy Attacks in Section 5.2

This section provides the full ROC analysis for machine unlearning benchmarks in CIFAR20 experiments. Figure 14 shows the ROC figures of class-wise membership inference attacks (MIA). Figure 15 shows the ROC figures of sample-wise membership inference attacks (MIA).

E.6. The Complete Representation Visualization Analysis of Experiments in Section 5.3

This appendix provides additional visualizations of lower-dimensional embeddings for Figure 4 in Section 5.3. All visualization results are presented in Figure 16. Moreover, the representation attributes are quantified using the following metrics:

- **Intra-class Variance (Variance)** (var) [19]: Measures class compactness by calculating the average squared distance from each point to the class centroid. A significantly lower variance than in the retrained model suggests that the unlearned data are densely clustered, indicating knowledge residue.
- **Silhouette Score** (s) [38]: Assesses clustering quality through the average silhouette coefficient for each point within the unlearned class, ranging from -1 to 1. High scores suggest that the model retains familiarity with class knowledge, pointing to knowledge residue.
- **Overlap Degree (Overlap)** (o) [2]: Evaluates the overlap between the unlearned subset and others using kernel density estimation (KDE). A lower score suggests an effective classification of unlearned data, indicating knowl-

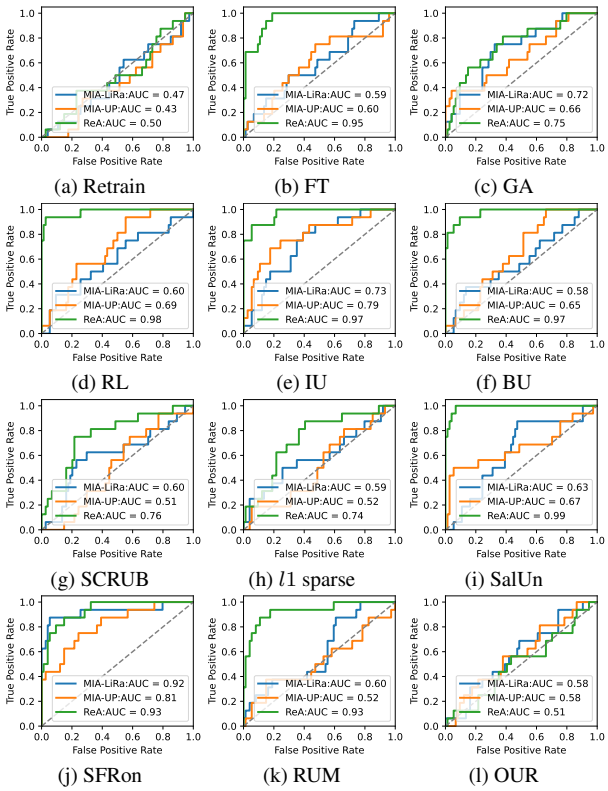


Figure 14. ROC of class-wise MIA. The notable fluctuations result from having only 300 inferred objects in each figure.

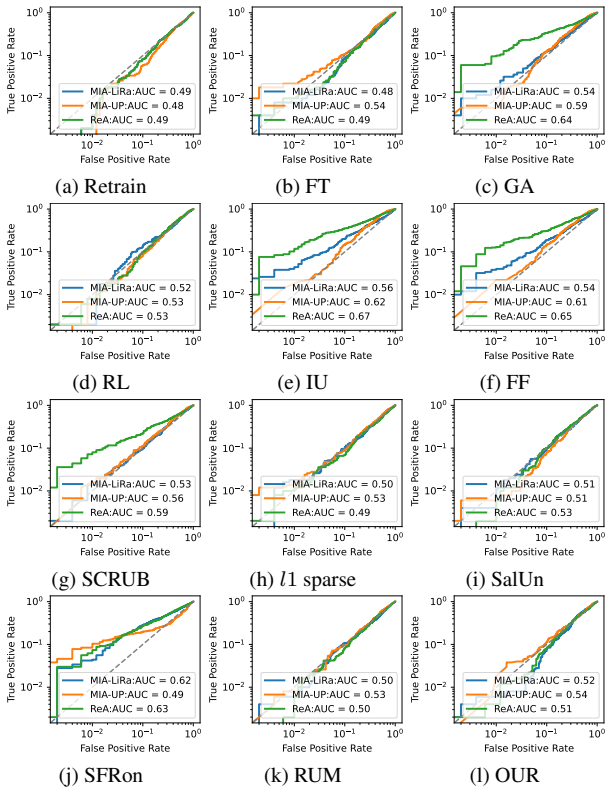


Figure 15. ROC of sample-wise MIA.

edge residue.

E.7. Supplementary Description of Setups and Experiments in Concept Unlearning

E.7.1. Setups

Models. We utilize the pretrained stable diffusion model `stable-diffusion-v1-5` [36] with prior preservation enabled, fine-tuning both the text encoder and diffusion model for 50 epochs using AdamW optimizer (learning rate is $5e^{-7}$), consistent with the work [43].

OUR Setups. For OUR unlearning method, the orthogonal unlearning is applied only to output layers of the last two hidden modular blocks (`up_blocks`) and the final convolutional layer (`conv_out`), with scaled orthogonal loss factors [$5e^{-9}$, $5e^{-8}$] to stabilize training. The two-phase optimization spans 10 epochs: 1 epoch for orthogonal unlearning (learning rate is $5e^{-6}$) and 9 epochs for the replay phase (learning rate is $8e^{-7}$), both using AdamW. Since the generation task involves fine-tuning a highly parameterized model, we do not apply L1 regularization to avoid compromising its capability.

Concept-Unlearning Benchmark Setups. SalUn [13] utilizes relabeled prompts (e.g., "This is a photo of James") to erase targeted concepts, while Meta Unlearning [15] hinders the reconstruction of unlearned concepts from related concepts during unlearning, which is described by "this is a photo of a woman". In terms of parameter configuration, SalUn and SFRon adopt identical hyper-parameters with learning rates of $1e^{-6}$ across 10 training epochs, whereas Meta Unlearning operates at a learning rate of $1e^{-5}$ over 50 optimization steps. These settings strictly adhere to their original designs.

Privacy Attack Setups. For ReA, distinct resonance indexes emerge between the unlearned and out-of-distribution (OOD) classes when fine-tuning with only the inferred class. Thus, ReA does not rely on retained or pseudo-retained datasets in concept unlearning. It employs a learning rate of $5e^{-6}$, and each inferred class consists of four samples. The prompt it used is a random name, e.g., Emily. For DiffAtk [47], the number of adversarial tokens is set to 5, and the embedding method for the adversarial prompt is prefix-based.

E.7.2. Metrics

The evaluation metrics for generated portrait outputs (defined in Section 5.5) are detailed as follows.

- **Identity Score Matching (ISM)** [43]. It evaluates identity consistency between generated and reference faces using ArcFace embeddings [11]. Lower values indicate better identity preservation. For identity-specific concept

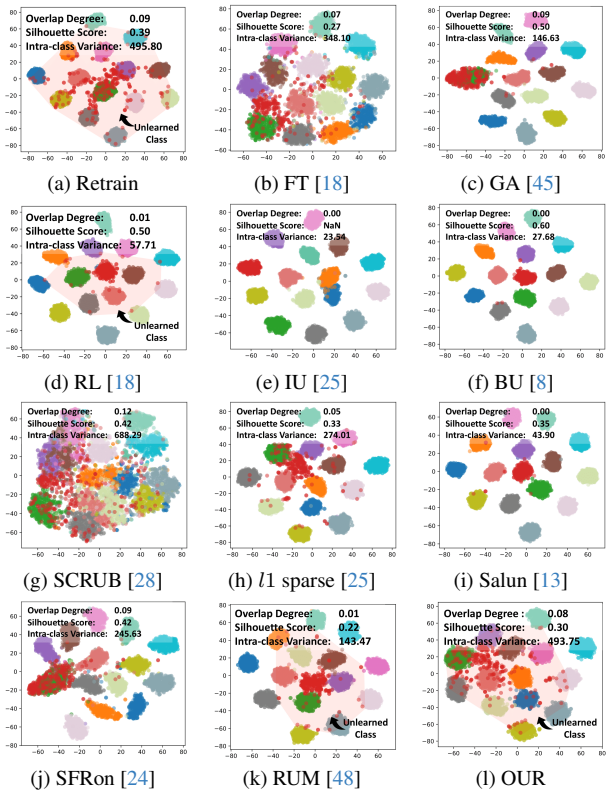


Figure 16. Representation space visualization of the class-wise unlearned model in CIFAR20 experiments.

Table 7. Performance of Identity-specific Concept Unlearning

Metrics	UISM	RISM	FID(↓)	BRISQUE(↓)	RTE (s)
Retrain	$0.13_{\pm 0.02}$(0.00)	$0.76_{\pm 0.02}$(0.00)	$101.82_{\pm 8.33}$	$10.76_{\pm 1.38}$	937.50
SFRon [24]	$0.50_{\pm 0.08}$(0.37)	$0.59_{\pm 0.03}$(0.17)	134.93	17.58	483.15

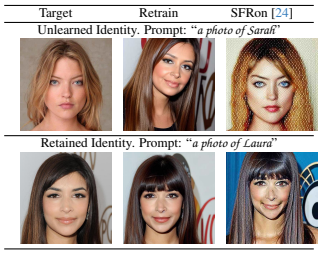
unlearning, we introduce two metrics: **Unlearning ISM (UISM)**, which evaluates the ISM of the unlearned identity, and **Retaining ISM (RISM)**, which evaluates the ISM of the retained identity. Both are used to observe how effectively the unlearned identity is removed and the remaining identity is preserved.

- **Fréchet Inception Distance (FID)** [26]. It quantifies the similarity between real and generated face distributions via Inception-v3 features. Lower scores reflect closer alignment to real data statistics.
- **BRISQUE** [34]. It assesses perceptual quality without reference by detecting unnatural patterns (e.g., artifacts, textures) in spatial features. Lower values denote more natural outputs.
- **Resonance Index in ReA Attack** (Idx_r). It measures the speed of identity recovery during the ReA process, which records the iteration where the ISM reaches 0.7.

Table 8. Privacy Attack against Identity-specific Concept Unlearning

Attacks	Resonance Index, Idx_r, (Maximum = 10) $\times$ 10 @ ISM	
	Retrain	SFRon [24]
DiffAtk [47]	$10_{\pm 0.0}$@0.18	$10_{\pm 0.0}$@-0.04
ReA	$6.5_{\pm 1.0}$@0.71	$4_{\pm 0.5}$@0.79

Table 9. Images of Identity-specific Concept Unlearning



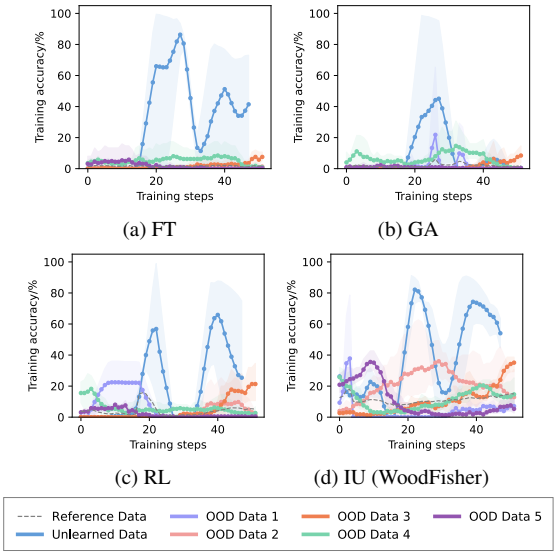


Figure 17. Training trajectories in the *Reminiscence Attack* under black-box scenarios. 3% of each type of data is available. The light-colored areas represent the error bars at different learning rates.

E.7.3. Experimental Results on an Additional Concept Unlearning Benchmark: SFRon [24]

This section presents an evaluation of SFRon [24] as an additional concept unlearning benchmark. Table 7 shows its performance in identity-specific concept unlearning, where UISM remains as high as 0.50, indicating ineffective removal of identity information. Table 9 provides further evidence, showing that the unlearning process mainly reduces image quality while preserving key identity features. As a result, SFRon is very vulnerable when attacked by ReA. It converges quickly in ReA, as shown in Table 8, where its Resonance Index reaches only 10, far below the retrained model's 70.

Table 10. Model Extraction Performance

Dataset	ME Accuracy (Fidelity) / %			
	FT	GA	RL	IU
Cifar20 (-5)	60.36 (77.73)	61.64 (78.79)	58.81 (73.65)	56.02 (69.53)
Cifar100 (-5)	54.83 (69.14)	56.74 (71.24)	58.81 (73.63)	56.29 (70.54)

Table 11. *Reminiscence Attack* Under Black-Box Scenarios

Dataset	Accuracy (TPR @ 0.1 FPR) / %				Cost (s)
	FT	GA	RL	IU	
Cifar20 (-5)	83.00 (90.00)	65.50 (55.00)	82.50 (70.00)	77.50 (80.00)	$9.16e^3$
Cifar100 (-5)	80.00 (85.00)	58.50 (40.00)	76.25 (57.50)	80.25 (65.00)	$1.36e^4$

E.8. Evaluating Reminiscence Attack under Black-box Scenarios

We investigate the class-wise ReA attack performance in black-box scenarios, to explore the transferability of knowledge residue from approximate unlearning in model extraction [41]. In white-box attacks, the attacker has direct access to the target model, allowing the ReA to be launched immediately. Conversely, in black-box scenarios, an additional preparatory step is required. This involves using data-free model extraction (ME) attacks (DFME) [41] to obtain a local substitute model on which the ReA is then launched. In black-box scenarios, we set the attacker's query budget to 20M and 30M in the ME step for Cifar20 (-5) and Cifar100 (-5), around the primary setup of DFME [41]. Four approximate unlearning benchmarks are evaluated.

Table 10 presents the accuracy and fidelity of substitute models obtained via DFME, and Table 11 reports the attack performance and computational cost of class-wise ReA against different unlearning methods. The adversarial set size is 3% for CIFAR20 (-5) and 20% for CIFAR100 (-5). Additionally, Figure 17 visualizes the reminiscence process in black-box settings, revealing that despite instability in training due to fidelity loss in ME, the unlearned dataset consistently exhibits distinct resonance.

Results. Overall, ReA remains highly effective in black-box scenarios, nearly matching its white-box performance despite higher computational costs. This suggests that **local models from ME inherit significant class-type residual knowledge**, exposing fundamental vulnerabilities in approximate unlearning methods.