# FedOrtho: Efficient Federated Unlearning via Orthogonal Convolution and Adaptive Soft Pruning

Supplementary Material

## 7. Full Proof of Kernel Orthogonality-Induced Feature Decoupling

To rigorously establish the feature decoupling effect induced by kernel orthogonality, we present a three-part mathematical derivation chain: Part A formalizes the convolutional functional and derives kernel near-orthogonality via orthogonal regularization; Part B connects kernel near-orthogonality to the statistical irrelevance of functional responses; Part C extends irrelevant functionals to complete feature decoupling between target and retained classes. The full derivation proceeds as follows.

**Foundational Definition: Convolutional Functional in Dual Space** The following definition formalizes the mathematical essence of convolutional responses, serving as the basis for subsequent derivations.

**Definition 2** (Convolutional Functional in Dual Space)**.** *Let the input data space be $\mathbb{R}^D$, where $D = C_{in} \cdot k^2$ ($C_{in}$: number of input channels; $k$: convolutional kernel size). The linear functionals in its dual space $(\mathbb{R}^D)^*$ satisfy three properties:*

1. *Unique Correspondence: Each convolutional kernel $\boldsymbol{w}_j \in \mathbb{R}^D$ uniquely determines a functional $f_j \in (\mathbb{R}^D)^*$;*
2. *Patch-Level Response: For a flattened $k \times k$ input patch $\boldsymbol{x} \in \mathbb{R}^D$, the response of $f_j$ is*

$$f_j(\boldsymbol{x}) = \boldsymbol{w}_j^\top \boldsymbol{x}, \tag{17}$$

3. *Image-Level Response: For an image $X$ partitioned into $m$ patches $\{\boldsymbol{x}_{u,v}\}_{1\le u,v\le m}$, the model's feature response intensity to $X$ is the maximum of functional responses:*

$$\Phi_j(X) = \max_{1\le u,v\le m} f_j(\boldsymbol{x}_{u,v}). \tag{18}$$

**Part A: Kernel Near-Orthogonality via Orthogonal Regularization** We first introduce the orthogonal regularization constraint on convolutional kernels, then derive the near-orthogonality of kernel vectors.

Let $\boldsymbol{W} \in \mathbb{R}^{C_{\text{out}} \times D}$ denote the convolutional kernel matrix, where each row corresponds to a kernel vector $\boldsymbol{w}_j \in \mathbb{R}^D$ ($C_{\text{out}}$: number of output channels). We impose orthogonal regularization on convolutional kernels during training, which enforces the kernel matrix to satisfy:

$$\mathcal{L}_{\text{ortho}} = \|\boldsymbol{W}\boldsymbol{W}^\top - \boldsymbol{I}\|_F \le \varepsilon, \tag{19}$$

where $\boldsymbol{I}$ is the $C_{\text{out}} \times C_{\text{out}}$ identity matrix, and $\varepsilon > 0$ is the deviation from strict orthogonality.

**Lemma 1** (Near-Orthogonality of Kernel Vectors)**.** *If the orthogonal regularization constraint in Eq.(19) holds, then:*

1. *For any distinct kernel vectors $\boldsymbol{w}_i, \boldsymbol{w}_j$ ($i \neq j$): $|\boldsymbol{w}_i^\top \boldsymbol{w}_j| \le \varepsilon$;*
2. *For any kernel vector $\boldsymbol{w}_j$: $1 - \varepsilon \le \|\boldsymbol{w}_j\|_2^2 \le 1 + \varepsilon$.*

*Proof.* By the definition of the Frobenius norm, the orthogonal regularization loss in Eq.(19) expands to:

$$\|\boldsymbol{W}\boldsymbol{W}^\top - \boldsymbol{I}\|_F^2 = \sum_{i=1}^{C_{\text{out}}} \sum_{j=1}^{C_{\text{out}}} \left(\boldsymbol{w}_i^\top \boldsymbol{w}_j - \delta_{ij}\right)^2 \le \varepsilon^2, \tag{20}$$

where $\delta_{ij}$ is the Kronecker delta (i.e., $\delta_{ij} = 1$ if $i = j$, and $\delta_{ij} = 0$ otherwise).

1. For $i \neq j$: The double sum reduces to a single term $(\boldsymbol{w}_i^\top \boldsymbol{w}_j)^2$. Since the sum of non-negative terms is bounded by $\varepsilon^2$, we have $(\boldsymbol{w}_i^\top \boldsymbol{w}_j)^2 \le \varepsilon^2$. Taking the square root of both sides gives $|\boldsymbol{w}_i^\top \boldsymbol{w}_j| \le \varepsilon$.

2. For $i = j$: The sum term becomes $(\|\boldsymbol{w}_j\|_2^2 - 1)^2$ (since $\boldsymbol{w}_j^\top \boldsymbol{w}_j = \|\boldsymbol{w}_j\|_2^2$ and $\delta_{jj} = 1$). Similarly, $(\|\boldsymbol{w}_j\|_2^2 - 1)^2 \le \varepsilon^2$. Taking the square root and rearranging terms gives $1 - \varepsilon \le \|\boldsymbol{w}_j\|_2^2 \le 1 + \varepsilon$. □

**Part B: From Kernel Near-Orthogonality to Irrelevant Functional Responses** To connect kernel orthogonality to functional behavior, we first formalize the Statistical Property of Input Patches, then derive that functional responses are statistically irrelevant via bounded covariance.

**Assumption 1** (Statistical Property of Input Patches)**.** *Input patches $\boldsymbol{x} \in \mathbb{R}^D$ (after Batch Normalization) satisfy two conditions:*

1. *Mean Zero: $\mathbb{E}[\boldsymbol{x}] = 0$, where $\mathbb{E}[\cdot]$ denotes the expectation over the patch distribution;*
2. *Approximate Whitening: The covariance matrix of $\boldsymbol{x}$ is*

$$\Sigma = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] = \boldsymbol{I} + \Delta, \tag{21}$$

*where $\Delta$ is the whitening error matrix with spectral norm $\|\Delta\|_2 \le \eta$ ($\eta > 0$ is the whitening error).*

**Theorem 2** (Near-Orthogonality of Functional Covariance)**.** *For any two functionals $f_i, f_j \in (\mathbb{R}^D)^*$ (induced by kernel vectors $\boldsymbol{w}_i, \boldsymbol{w}_j$ via Definition 2), their covariance satisfies:*

$$|Cov(f_i, f_j)| \le \varepsilon + (1 + \varepsilon)\eta =: \varepsilon_1, \tag{22}$$

*where $\varepsilon_1$ is a small constant that approaches 0 as $\varepsilon \to 0$ (strict kernel orthogonality) and $\eta \to 0$ (perfect patch whitening).*

*Proof.* By the definition of covariance, we have:

$$\mathrm{Cov}(f_i, f_j) = \mathbb{E}[f_i(\boldsymbol{x}) f_j(\boldsymbol{x})] - \mathbb{E}[f_i(\boldsymbol{x})]\mathbb{E}[f_j(\boldsymbol{x})]. \quad (23)$$

From Assumption 1, $\mathbb{E}[\boldsymbol{x}] = 0$. For any functional $f_k(\boldsymbol{x}) = \boldsymbol{w}_k^\top \boldsymbol{x}$ (via Definition 2), this implies $\mathbb{E}[f_k(\boldsymbol{x})] = \boldsymbol{w}_k^\top \mathbb{E}[\boldsymbol{x}] = 0$. Thus, the covariance simplifies to:

$$\mathrm{Cov}(f_i, f_j) = \mathbb{E}[f_i(\boldsymbol{x}) f_j(\boldsymbol{x})] = \mathbb{E}[\boldsymbol{w}_i^\top \boldsymbol{x} \cdot \boldsymbol{w}_j^\top \boldsymbol{x}]. \quad (24)$$

Using the linearity of expectation and the definition of covariance matrix $\Sigma = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ (from Assumption 1), we rewrite the expectation as:

$$\mathbb{E}[\boldsymbol{w}_i^\top \boldsymbol{x} \cdot \boldsymbol{w}_j^\top \boldsymbol{x}] = \boldsymbol{w}_i^\top \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top] \boldsymbol{w}_j = \boldsymbol{w}_i^\top \Sigma \boldsymbol{w}_j. \quad (25)$$

Substitute $\Sigma = \boldsymbol{I} + \Delta$ (from Assumption 1) and split the term:

$$\boldsymbol{w}_i^\top \Sigma \boldsymbol{w}_j = \boldsymbol{w}_i^\top \boldsymbol{w}_j + \boldsymbol{w}_i^\top \Delta \boldsymbol{w}_j. \quad (26)$$

By the triangle inequality, $|\boldsymbol{a} + \boldsymbol{b}| \le |\boldsymbol{a}| + |\boldsymbol{b}|$, so:

$$|\boldsymbol{w}_i^\top \Sigma \boldsymbol{w}_j| \le |\boldsymbol{w}_i^\top \boldsymbol{w}_j| + |\boldsymbol{w}_i^\top \Delta \boldsymbol{w}_j|. \quad (27)$$

For the second term, apply the Cauchy-Schwarz inequality: $|\boldsymbol{a}^\top \Delta \boldsymbol{b}| \le \|\boldsymbol{a}\|_2 \|\Delta\|_2 \|\boldsymbol{b}\|_2$. From Lemma 1, $\|\boldsymbol{w}_i\|_2 \le \sqrt{1+\varepsilon}$ and $\|\boldsymbol{w}_j\|_2 \le \sqrt{1+\varepsilon}$; from Assumption 1, $\|\Delta\|_2 \le \eta$. Thus:

$$|\boldsymbol{w}_i^\top \Delta \boldsymbol{w}_j| \le \sqrt{1+\varepsilon} \cdot \eta \cdot \sqrt{1+\varepsilon} = (1+\varepsilon)\eta. \quad (28)$$

From Lemma 1, $|\boldsymbol{w}_i^\top \boldsymbol{w}_j| \le \varepsilon$. Substituting both bounds into Eq.(27) gives Eq.(22). □

**Part C: From Irrelevant Functionals to Feature Decoupling** We first define class-specific functionals and additional statistical assumptions, then derive that irrelevant functionals lead to complete feature decoupling between target and retained classes.

**Definition 3** (Class-Specific Functional Sets)**.** *Let $A_{target,j} = \mathbb{E}_{X \sim X_{target}}[\Phi_j(X)]$ (average image-level response of $f_j$ on target-class images $X_{target}$) and $A_{retain,j} = \mathbb{E}_{X \sim X_{retain}}[\Phi_j(X)]$ (average response on retained-class images $X_{retain}$). For a threshold $\tau > 0$:*

1. *Target-Data Functional Set: $\mathcal{F}_{target} = \{f_j \in (\mathbb{R}^D)^* \mid A_{target,j} - A_{retain,j} > \tau\}$ (functionals strongly associated with the target data);*
2. *Retained-Data Functional Set: $\mathcal{F}_{retain} = (\mathbb{R}^D)^* \setminus \mathcal{F}_{target}$ (functionals associated with retained data, complementary to $\mathcal{F}_{target}$).*

*Let $m_{eff}$ denote the number of effectively independent patches in $X$ (correcting for correlation induced by patch overlap or stride).*

**Assumption 2** (Sub-Gaussian Property of Functionals)**.** *Each functional $f_j(\boldsymbol{x})$ (induced via Definition 2) is a sub-Gaussian random variable: there exists a constant $\sigma > 0$ such that for all $t \in \mathbb{R}$,*

$$\mathbb{E}\left[e^{t f_j(\boldsymbol{x})}\right] \le e^{\sigma^2 t^2/2}, \quad (29)$$

*and the variance of $f_j(\boldsymbol{x})$ satisfies $Var(f_j(\boldsymbol{x})) \le \sigma^2$.*

**Assumption 3** (Strong Activation of Target-Data Functionals)**.** *There exists a constant $\gamma > 0$ such that for all $f_i \in \mathcal{F}_{target}$ (via Definition 3), the average response on target-data images satisfies:*

$$A_{target,i} \ge \gamma. \quad (30)$$

**Theorem 3** (Data-Specific Role Separation)**.** *For any $f_i \in \mathcal{F}_{target}$ (via Definition 3) and $f_j \in \mathcal{F}_{retain}$ (via Definition 3), the average response of $f_j$ on target-class images satisfies:*

$$A_{target,j} \le C\left(\varepsilon_1 + \sigma\sqrt{\frac{\log m_{eff}}{m_{eff}}}\right) =: \epsilon_2, \quad (31)$$

*where $C > 0$ is a model-agnostic constant, and $\epsilon_2 \to 0$ as $\varepsilon_1 \to 0$ (via Theorem 2) and $m_{eff} \to \infty$ (sufficient independent patches).*

*Proof.* The proof proceeds in three steps, combining conditional expectation control and extreme value bounds for sub-Gaussian variables.

**1: Bounded Conditional Expectation** For a patch $\boldsymbol{x}$, let $f_i(\boldsymbol{x}) = t$ (response of the target-class functional $f_i$). Since $f_i$ and $f_j$ are linear functionals of $\boldsymbol{x}$ (via Definition 2), their conditional expectation satisfies a linear relationship:

$$\mathbb{E}[f_j(\boldsymbol{x}) \mid f_i(\boldsymbol{x}) = t] = \frac{\mathrm{Cov}(f_i, f_j)}{\mathrm{Var}(f_i)} \cdot t. \quad (32)$$

From Theorem 2, $|\mathrm{Cov}(f_i, f_j)| \le \varepsilon_1$; from Assumption 2, $\mathrm{Var}(f_i) \ge \sigma^2$ (lower bound of sub-Gaussian variance). Substituting these bounds gives:

$$|\mathbb{E}[f_j(\boldsymbol{x}) \mid f_i(\boldsymbol{x}) = t]| \le \frac{\varepsilon_1}{\sigma^2} \cdot |t|. \quad (33)$$

From Assumption 3, $A_{\text{target},i} \ge \gamma$, but $\varepsilon_1$ is a small constant (via Theorem 2), so the conditional expectation of $f_j$ is bounded by $O(\varepsilon_1)$ (local response suppression).

**2: Extreme Value Bound for Sub-Gaussian Variables** For $m_{\text{eff}}$ effectively independent patches, the responses $\{f_j(\boldsymbol{x}_{u,v})\}_{1 \le u,v \le m_{\text{eff}}}$ are independent sub-Gaussian variables (via Assumption 2). By the Borell-TIS inequality (a

fundamental result for sub-Gaussian extremes), the expectation of their maximum satisfies:

$$\mathbb{E}\left[\max_{1\le u,v\le m_{\text{eff}}} f_j(\boldsymbol{x}_{u,v})\right] \le O\left(\sigma\sqrt{\frac{\log m_{\text{eff}}}{m_{\text{eff}}}}\right). \quad (34)$$

This bound shows that the global maximum response of $f_j$ decays to 0 as $m_{\text{eff}}$ increases (global response suppression).

**3: Merge Local and Global Bounds** The average response $A_{\text{target},j} = \mathbb{E}_{X\sim X_{\text{target}}}[\Phi_j(X)]$ (via Definition 3) is the expectation of the maximum functional response over patches. Combining the local conditional expectation bound Eq.(33) and global extreme value bound Eq.(34), we obtain Eq.(31), where $C > 0$ is a constant integrating the two bounds. □

**Theorem 4** (Feature Decoupling)**.** *For any $f_i \in \mathcal{F}_{target}$ (via Definition 3) and $f_j \in \mathcal{F}_{retain}$ (via Definition 3), the covariance of their image-level responses on target-data images satisfies:*

$$\mathbb{E}[\Phi_i(X)\Phi_j(X)] \le B\cdot\epsilon_2 =: \delta, \quad (35)$$

*where $B > 0$ is the upper bound of $\Phi_i(X)$ (activation saturation in deep networks), and $\delta \to 0$ as $\varepsilon \to 0$, $\eta \to 0$, and $m_{eff} \to \infty$.*

*Proof.* By the law of total expectation, the expectation of the product $\Phi_i(X)\Phi_j(X)$ can be bounded using the maximum response of $\Phi_i(X)$:

$$\mathbb{E}[\Phi_i(X)\Phi_j(X)] \le \max_{X\sim X_{\text{target}}} \Phi_i(X)\cdot\mathbb{E}[\Phi_j(X)]. \quad (36)$$

Let $B = \max_{X\sim X_{\text{target}}} \Phi_i(X)$. From Definition 3, $\mathbb{E}[\Phi_j(X)] = A_{\text{target},j}$, so:

$$\mathbb{E}[\Phi_i(X)\Phi_j(X)] \le B\cdot A_{\text{target},j}. \quad (37)$$

From Theorem 3, $A_{\text{target},j} \le \epsilon_2$. Substituting this bound into Eq.(37) gives Eq.(35).

To confirm $\delta \to 0$, we trace the error chain:

1. From Theorem 2, $\varepsilon \to 0$ and $\eta \to 0$ imply $\varepsilon_1 \to 0$ (via Eq.(22));
2. From Theorem 3, $\varepsilon_1 \to 0$ and $m_{\text{eff}} \to \infty$ imply $\epsilon_2 \to 0$ (via Eq.(31));
3. $\delta = B\cdot\epsilon_2 \to 0$, meaning the responses of target-data and retained-data functionals are asymptotically uncorrelated, which means feature decoupling is achieved.

□

# 8. Motivational Observations

## 8.1. Feature Similarity and Redundancy Analysis

Fig. 6 presents the feature similarity distribution curves of convolutional kernels in the shallow, middle, and deep layers after applying orthogonality constraints (the x-axis represents feature similarity values, and the y-axis denotes the proportion of samples with the corresponding similarity). We select three key convolutional layers from each of the three hierarchical levels (shallow, middle, and deep) to intuitively illustrate the statistical distribution characteristics of pairwise feature similarity through the distribution curves. Experimental results show that: when orthogonality constraints are only applied to deep convolutional kernels (Fig. 6 (a)), the peak of the curve shifts significantly toward the low-similarity interval, and the proportion of low-similarity samples increases substantially, indicating that redundant correlations among deep features are effectively weakened. In contrast, when orthogonality constraints are applied to shallow or middle layers (Fig. 6 (b)-(c)), the similarity distribution curve of deep features remains concentrated in the high-similarity interval, with no significant increase in the proportion of low-similarity samples, making it difficult to achieve efficient suppression of deep feature redundancy. It is worth noting that deep features are the core of the model's high-level semantic representation, and their redundancy directly affects the model's feature discrimination ability. This result fully verifies that orthogonality constraints on deep convolutional kernels are a key means to guide deep features toward a low-similarity distribution and reduce the redundancy of feature representations.

## 8.2. Gram Matrix Visualization Analysis

Fig. 7 presents the hierarchical Gram matrix heatmaps of the baseline non-orthogonal model and the model with orthogonality constraints only applied to deep convolutional kernels, covering the initial convolutional kernels and three key convolutional layers (shallow, middle, and deep). As a core quantitative indicator of orthogonality in the feature space, the element $G_{i,j}$ of the Gram matrix essentially corresponds to the inner product of the $i$-th and $j$-th convolutional kernel feature maps. Its magnitude directly reflects the linear correlation between cross-channel feature maps, when the inner product is close to zero, the feature maps are approximately orthogonal in the Hilbert space, achieving the lowest feature redundancy. Notably, comparing the Gram matrix distributions of the shallow layers and initial convolutional kernels in Fig. 7(a) and Fig. 7(b), there is no significant difference in the off-diagonal response intensity and diagonal energy distribution between the two. This indicates that the orthogonality constraint has strict layer-specificity in optimizing deep features, without interfering with the inherent representational ability of shallow and

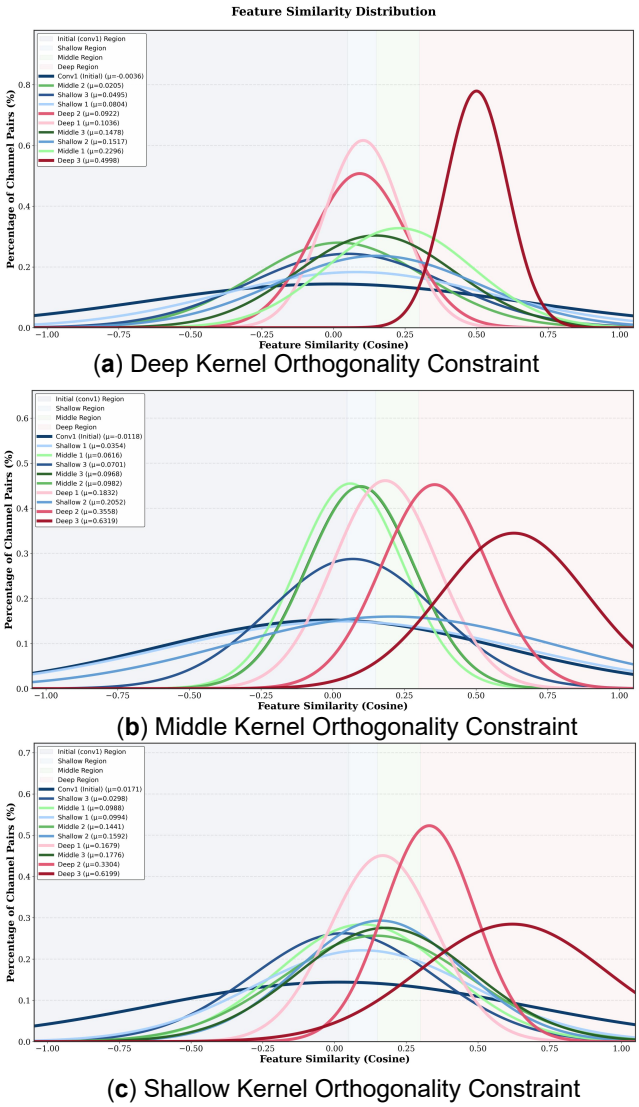



Figure 6. Feature Similarity Visualization and Redundancy Suppression via Orthogonality Constraints on Shallow, Middle, and Deep Layers

initial convolutional kernels for basic visual features (e.g., edges, textures), thereby preserving the native representational bias of the network. Combined with the overall performance metrics, this layer-specific constraint strategy not only effectively suppresses the redundancy of deep high-level semantic features but also maintains the integrity of shallow basic features. It fully verifies the method's favorable trade-off between feature decoupling and performance preservation, while being highly consistent with the functional division of labor in deep networks, where shallow layers extract basic features and deep layers model high-level semantics.

# 9. Experiment Settings

## 9.1. Federated Training Setup

The hardware platform for federated learning training is equipped with Intel Xeon Platinum 24-core processors, two NVIDIA RTX 4090 GPUs (24 GB VRAM per GPU), 251.406 GB DDR5 system memory, and two HDDs (labeled sda and sdb). It runs on the Ubuntu 22.04 system with Python 3.9 and the PyTorch 2.1.0 framework (supported by CUDA 12.1). The experiments use the CIFAR-100 (100 classes, 32×32) and TinyImageNet (200 classes, 64×64) datasets, deploying 100 virtual clients. Data is partitioned following a Dirichlet distribution ($\alpha = 0.3/0.6$ to control non-IID distribution), with 10% of clients randomly selected per round for training. The models used are ResNet-18, ResNet-50, and VGG-16 (all adapted to input sizes and class numbers). Training employs the SGD optimizer (momentum 0.9, weight decay $1 \times 10^{-3}$) with an initial learning rate of 0.1, combined with a per-round learning rate decay of 0.998. A total of 1000 global communication rounds are performed, with 5 local epochs per round and a batch size of 50 (using mixed-precision training). Model aggregation adopts the FedAvg strategy, with orthogonality constraints ($\lambda_{\text{ortho}} = 0.1$, parameter search range: 0.01, 0.05, 0.1, 0.2, 0.5) and regularization ($\lambda_{\text{reg}} = 0.1$, parameter search range: 0.01, 0.05, 0.1, 0.2, 0.5) applied to deep convolutional layers. Data is augmented via random cropping, horizontal flipping, and random erasing, normalized using dataset-specific statistics. For the unlearning phase, fine-tuning is performed on all non-forgetting clients with a learning rate of 0.001. Results are reported as the mean over 3 runs with different random seeds (0, 20, 23).

## 9.2. Centralized Training Setup

Experiments are conducted on a workstation equipped with dual 14-core Intel Xeon CPUs and an NVIDIA TITAN Xp GPU (12GB VRAM), running on the Ubuntu 20.04 operating system with CUDA 11.7 and PyTorch 1.13 as the deep learning framework. The training uses the SGD optimizer with a momentum of 0.9 and a weight decay of $5 \times 10^{-4}$, with an initial learning rate of 0.1. The batch size is set to 256 for TinyImageNet and 128 for CIFAR-10/100. The training is performed for 200 epochs on TinyImageNet with a multi-step learning rate scheduler (milestones at 30 and 60 epochs) and 150 epochs on CIFAR datasets with cosine annealing. Models are adapted to match the input resolution and label dimension of each dataset. For data preprocessing, all images are normalized using dataset-specific statistics and augmented with random cropping and horizontal flipping. Experimental results are averaged over three independent runs with different random seeds to ensure reliability.

## 10. Unlearning Performance Under Different Settings

To verify the effectiveness of our proposed method (FedOrtho), we design two major experimental scenarios: federated unlearning experiments and centralized class-level unlearning experiments. The experiments cover multiple datasets, models, and data distributions. By comparing with classical and mainstream baseline methods, we evaluate the comprehensive performance of the method from multiple dimensions, including forgetting effectiveness, retention performance, privacy protection, and training efficiency. The results ultimately confirm that FedOrtho achieves excellent performance across all metrics.

### 10.1. Federated Unlearning Experiments

Based on the ResNet18 model, we conduct experiments on the CIFAR-100 datasets, covering both IID and Non-IID data distribution scenarios (weak heterogeneity with $\alpha$=0.6 and strong heterogeneity with $\alpha$=0.3). Three core unlearning tasks are completed: client unlearning, class unlearning, and sample unlearning. Classical federated unlearning methods such as Retrain, FT, NoT, and FUSED are selected as baselines. We focus on verifying the multi-task adaptability and comprehensive advantages of FedOrtho under heterogeneous data distributions through three evaluation metrics: forgetting accuracy ($A_{\mathrm{Te}_{D_u}}$), retention accuracy ($A_{\mathrm{Te}_{D_r}}$), and membership inference attack (MIA) for privacy protection.

### 10.2. Centralized Class-Level Unlearning Experiments

For the centralized class-level unlearning scenario, we expand the coverage of models and datasets: the models include Res18, Res34, and VGG-16, while the datasets consist of CIFAR-10, CIFAR-100, and Tiny ImageNet. Mainstream centralized unlearning schemes such as Retrain, SSD, and EMNI, Salun, BTF are adopted as baselines. Considering the practical demand for efficiency in centralized scenarios, training time is added as an efficiency evaluation dimension on the basis of the three evaluation metrics used in federated unlearning. The results show that FedOrtho not only stably ensures the effectiveness of forgetting and the reliability of retention performance but also features extremely short unlearning time, achieving the triple optimization of ”effectiveness-performance-efficiency” in centralized scenarios.

### 10.3. Case Study

Fig. 8 presents the Grad-CAM heatmaps of target convolutional kernels for ResNet50, ResNet18, and VGG-16 before and after pruning on CIFAR-10. The visualization covers all ten classes, demonstrating that the proposed pruning method can maintain target feature extraction across different network architectures and classes, thus showing excellent generalization ability.

## 11. Limition

Due to communication restrictions in FL, current experiments focus on CNNs and classification tasks, but the effectiveness of non-conv architectures (e.g., Transformers) remains to be verified. We plan to explore extending the orthogonal idea to deeper, multi-task networks in the future.

Table 7. Federated unlearning performance of ResNet18 on IID/Non-IID datasets. Metrics:$A_{D_u}$/$A_{\mathrm{Te}_{D_u}}$ (Forget Accuracy on revoked training/test data),$A_{\mathrm{Te}_{D_r}}$ (Retain Accuracy), MIA (privacy metric).

**Non-IID (Dirichlet=0.6)**

| Model | Method | Client Unlearning | | | Class Unlearning | | | Sample Unlearning | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $A_{D_u}\downarrow$ | $A_{\mathrm{Te}_{D_r}}\uparrow$ | MIA $\downarrow$ | $A_{\mathrm{Te}_{D_u}}\downarrow$ | $A_{\mathrm{Te}_{D_r}}\uparrow$ | MIA $\downarrow$ | $A_{D_u}\downarrow$ | $A_{\mathrm{Te}_{D_r}}\uparrow$ | MIA $\downarrow$ |
| ResNet18 | Retrain | 36.80±0.92 | 46.72±0.23 | 89.00±0.58 | 0.00±0.00 | 46.91±0.45 | 75.93±0.00 | 41.00±1.05 | 46.57±0.19 | 94.82±0.47 |
| | FT | 40.00±1.15 | **46.87**±0.62 | 92.80±0.70 | 4.93±0.44 | **47.45**±0.17 | 87.33±0.93 | 44.00±0.88 | **47.89**±0.03 | 98.67±0.35 |
| | NoT | 37.00±2.98 | 44.65±0.66 | 87.46±1.95 | 7.26±0.53 | 46.78±0.91 | 72.17±0.29 | 38.33±1.17 | 47.12±0.96 | 94.33±0.54 |
| | FUSED | 35.73±1.03 | 39.41±0.85 | 67.87±0.67 | 6.00±0.22 | 41.57±1.07 | 57.43±0.23 | 35.73±0.67 | 40.58±1.13 | 72.67±0.71 |
| | FedOrtho | **33.06**±0.04 | 44.92±0.25 | **57.33**±0.55 | **0.00**±0.00 | 46.13±0.06 | **51.13**±0.26 | **36.87**±0.08 | 46.31±0.21 | **65.33**±0.73 |

**Non-IID (Dirichlet=0.3)**

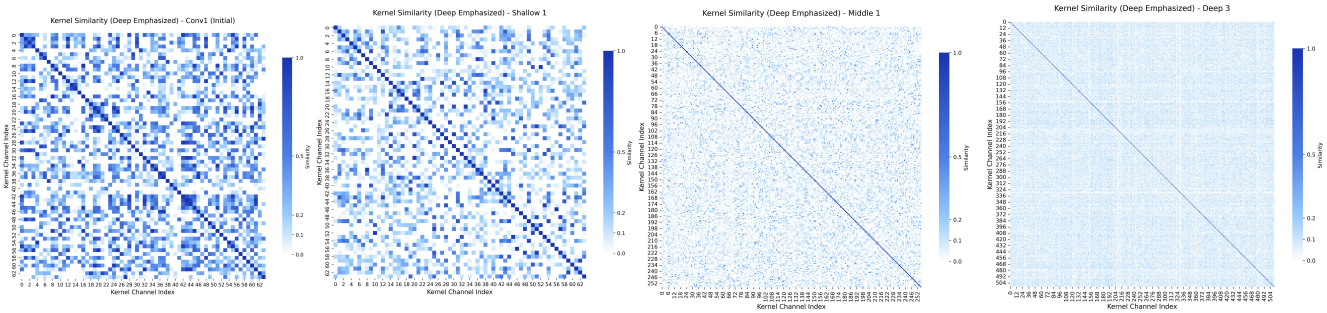
| Model | Method | Client Unlearning | | | Class Unlearning | | | Sample Unlearning | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $A_{D_u}\downarrow$ | $A_{\mathrm{Te}_{D_r}}\uparrow$ | MIA $\downarrow$ | $A_{\mathrm{Te}_{D_u}}\downarrow$ | $A_{\mathrm{Te}_{D_r}}\uparrow$ | MIA $\downarrow$ | $A_{D_u}\downarrow$ | $A_{\mathrm{Te}_{D_r}}\uparrow$ | MIA $\downarrow$ |
| ResNet18 | Retrain | 33.20±0.93 | 43.88±0.29 | 90.47±0.66 | 0.00±0.00 | 44.35±0.91 | 77.26±0.00 | 38.67±1.04 | 43.86±0.15 | 96.67±0.45 |
| | FT | 42.68±1.10 | **44.52**±0.94 | 94.32±0.60 | 6.75±0.49 | **45.19**±0.71 | 89.14±0.95 | 43.59±1.07 | **45.37**±0.20 | 99.21±0.34 |
| | NoT | 39.73±2.89 | 42.03±0.04 | 89.20±0.68 | 8.93±0.54 | 43.96±0.87 | 75.76±2.38 | 39.67±1.05 | 42.97±0.90 | 96.00±0.47 |
| | FUSED | 38.13±0.99 | 39.29±0.89 | 69.47±0.65 | 8.00±0.24 | 37.84±0.92 | 59.20±0.30 | 34.00±0.98 | 41.42±0.88 | 75.00±0.64 |
| | FedOrtho | **31.06**±0.05 | 41.57±0.11 | **69.80**±0.73 | **0.00**±0.00 | 43.62±0.04 | **63.33**±0.32 | **37.00**±0.03 | 42.79±0.17 | **67.58**±0.52 |

**IID**

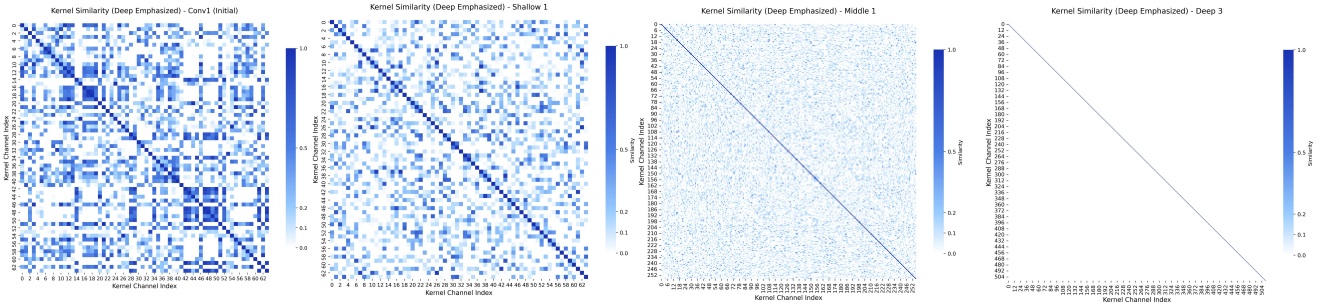
| Model | Method | Client Unlearning | | | Class Unlearning | | | Sample Unlearning | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $A_{D_u}\downarrow$ | $A_{\mathrm{Te}_{D_r}}\uparrow$ | MIA $\downarrow$ | $A_{\mathrm{Te}_{D_u}}\downarrow$ | $A_{\mathrm{Te}_{D_r}}\uparrow$ | MIA $\downarrow$ | $A_{D_u}\downarrow$ | $A_{\mathrm{Te}_{D_r}}\uparrow$ | MIA $\downarrow$ |
| ResNet18 | Retrain | 34.40±0.87 | 48.25±0.09 | 86.40±0.64 | 0.00±0.00 | 49.57±0.18 | 73.17±0.00 | 33.00±0.69 | 48.93±0.10 | 92.67±0.20 |
| | FT | 37.20±1.08 | **49.61**±1.05 | 90.53±0.70 | 3.86±0.45 | **50.23**±0.60 | 85.47±0.91 | 38.33±0.99 | **50.19**±0.08 | 97.33±0.33 |
| | NoT | 34.33±0.94 | 47.19±0.20 | 85.13±0.67 | 6.17±0.52 | 49.14±0.96 | 71.03±2.82 | 33.67±1.63 | 48.35±0.99 | 92.00±0.48 |
| | FUSED | 33.06±0.42 | 41.87±0.91 | 65.73±0.59 | 4.53±0.21 | 47.39±0.93 | 55.67±0.21 | 32.67±1.02 | 45.96±0.94 | 70.33±0.61 |
| | FedOrtho | **26.53**±0.06 | 46.89±0.14 | **55.06**±0.56 | **0.00**±0.00 | 48.91±0.05 | **49.20**±0.24 | **32.00**±0.05 | 48.15±0.08 | **63.00**±0.63 |

Table 8. Model Performance Comparison Under the Centralized Class Unlearning Scenario (Forgetting a Single Class, Averaged Over 3 Runs with Different Random Seeds, Res18/Res34/VGG-16, CIFAR-10/CIFAR-100/Tiny ImageNet)

| Model | Method | CIFAR-10 | | | | CIFAR-100 | | | | Tiny ImageNet | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $A_{Te_{D_u}}\downarrow$ | $A_{Te_{D_r}}\uparrow$ | Time$_\downarrow$ | MIA$_\downarrow$ | $A_{Te_{D_u}}\downarrow$ | $A_{Te_{D_r}}\uparrow$ | Time$_\downarrow$ | MIA$_\downarrow$ | $A_{Te_{D_u}}\downarrow$ | $A_{Te_{D_r}}\uparrow$ | Time$_\downarrow$ | MIA$_\downarrow$ |
| Res18 | Retrain | 0.00% | 92.29% | 4360.80s | 0.00% | 0.00% | 78.17% | 5557.26s | 0.00% | 0.00% | 64.20% | 15935.06s | 0.00% |
| | SSD | 4.37% | 89.30% | 21.60s | 3.82% | 1.67% | 76.29% | 21.38s | 2.67% | **0.00%** | 61.56% | 136.77s | 6.67% |
| | EMNI | 9.76% | **91.80%** | 322.02s | 8.29% | 5.00% | 73.89% | 335.40s | 6.00% | 4.00% | 59.43% | 348.42s | 10.0% |
| | Salun | **0.00%** | 87.45% | 20.47$s$ | **0.60%** | **0.00%** | 73.93% | 21.55$s$ | **0.00%** | **0.00%** | 58.27% | 45.79$s$ | 6.67% |
| | BTF | 8.61% | 88.34% | 23.26s | 7.74% | 2.00% | 77.00% | 24.36s | 4.67% | 1.33% | 59.70% | 47.52s | 4.67% |
| | FedOrtho | **0.00%** | 91.34% | **0.1254s** | 1.82% | **0.00%** | **77.45%** | **0.1135s** | 2.00% | **0.00%** | **63.11%** | **0.1244s** | **2.00%** |
| Res34 | Retrain | 0.00% | 93.42% | 5377.80s | 0.00% | 0.00% | 79.00% | 6875.10s | 0.00% | 0.00% | 64.73% | 17051.00s | 0.00% |
| | SSD | 1.93% | 90.91% | 48.60s | **0.20%** | **0.00%** | 77.87% | 49.00s | 2.33% | **0.00%** | 61.39% | 317.22s | **0.00%** |
| | EMNI | 3.27% | 91.46% | 556.26s | 8.60% | 7.33% | 75.38% | 562.20s | 6.67% | 2.67% | 60.04% | 583.80s | 6.00% |
| | Salun | **0.00%** | 88.17% | 27.57$s$ | 0.00% | **0.00%** | 76.44% | 29.20$s$ | **0.00%** | **0.00%** | 59.79% | 62.04$s$ | 2.67% |
| | BTF | 4.40% | 86.92% | 41.29s | 8.30% | 1.67% | 76.10% | 41.53s | 3.67% | 0.67% | 59.64% | 81.60s | 4.00% |
| | FedOrtho | **0.00%** | **92.05%** | **0.1342s** | 2.60% | **0.00%** | **78.52%** | **0.1290s** | 2.67% | **0.00%** | **63.68%** | **0.1350s** | 2.67% |
| VGG-16 | Retrain | 0.00% | 92.61% | 3976.71s | 0.00% | 0.00% | 74.48% | 4407.80s | 0.00% | 0.00% | 60.89% | 10033.68s | 0.00% |
| | SSD | **0.00%** | 86.19% | 13.83$s$ | 4.20% | **0.00%** | 70.43% | 13.72$s$ | 2.67% | **0.00%** | **59.95%** | 61.48$s$ | 4.67% |
| | EMNI | 2.47% | 90.28% | 289.81s | 2.50% | 3.33% | 71.57% | 300.53s | 8.33% | 2.67% | 56.38% | 312.63s | 5.33% |
| | Salun | **0.00%** | 87.81% | 15.77s | **0.00%** | **0.00%** | 71.77% | 16.19s | **0.00%** | **0.00%** | 54.59% | 33.29s | **1.33%** |
| | BTF | 9.13% | 88.83% | 17.32s | 4.33% | 3.67% | 72.49% | 18.07s | 5.33% | 0.67% | 55.78% | 38.64s | 6.67% |
| | FedOrtho | **0.00%** | **91.32%** | **0.0780s** | 0.10% | **0.00%** | **73.65%** | **0.0850s** | 2.67% | **0.00%** | 59.24% | **0.1020s** | 3.33% |




(**a**) Gram Matrix of Non-Orthogonal Model (Initial Convolutional Kernels - Shallow Layers - Middle Layers - Deep Layers)




(**b**) Gram Matrix of Orthogonal Model (Initial Convolutional Kernels - Shallow Layers - Middle Layers - Deep Layers)

Figure 7. Gram matrix comparison across layers, showing deep-layer orthogonalization reduces redundancy while preserving shallow features.

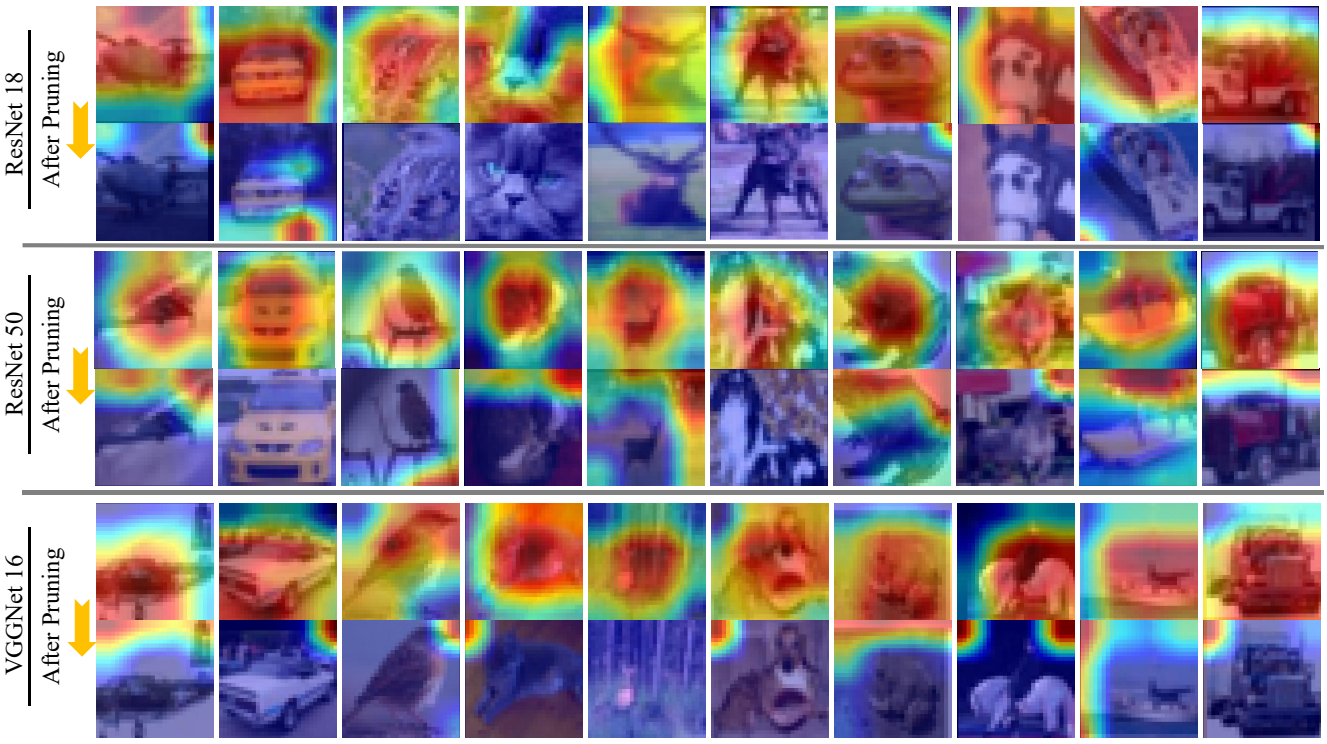



Figure 8. Comparison of convolutional kernel Grad-CAM heatmaps before and after pruning ResNet-50, ResNet-18, and VGG-16 on the CIFAR-10 dataset