APPENDIX A

A. 1 The pseudo-code of FedCMM

Algorithm 1 The FedCMM algorithm

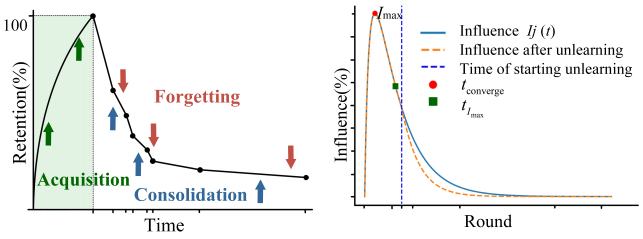
Input: Initialization parameters w^0, R, T
Output: $w_t^{\tilde{u}}$
1: **begin**
2: **for** r = 1, 2, …, R **do** *// FL phase*
3: Each client train local model with w^0 and upload $w_{i,r}$ to the CS;
4: The global parameter w_r^* is obtained according to federated average.
5: **end for**
6: **for** each federated round t = R+1, R+2, …, R+T **do**
7: **for** each client **do** *// InterFor phase*
8: **if** e_j contains forgotten data **do**
9: e_j calculates gradient g_j^u of forgotten data D_j^u and updates $w'_{j,t-1}$ with Eq. (3);
10: e_j updates the labels of D_j^u with Eqs. (4 and 6) and performs local training with all data and w_j;
11: e_j updates parameter $w_{j,t}$ with Eq. (7) and uploads the $w_{j,t}^u$ to the CS.
12: **end if** *// IntriFor phase*
13: **if** e_i contains no forgotten data **do**
14: e_i obtains augmented data $D(D_i)$ by diffusion model based on Eq. (11);
15: e_i performs local training with all data and obtains the parameters $w_{i,t}^{\tilde{u}}$;
16: e_i uploads $w_{i,t}^{\tilde{u}}$ to the CS.
17: **end if**
18: **end for**
19: The CS obtains $w_{j,t}^u$ and $w_{i,t}^{\tilde{u}}$, and performs federated aggregation to obtain $w_t^{\tilde{u}}$.
20: **end for**
21: **end**

A. 2 The pseudo-code of the conditional diffusion model

The diffusion model in IntriFor involves a forward and reverse sampling procedure.

Algorithm 2 The conditional diffusion model $D(x_i)$

Input: Diffusion steps T_i, noise schedule $\{\beta_t\}_{t=1}^T$, conditioning input c, learning rate η
Output: Generated data samples x_0
1: **begin** *// Forward diffusion process*
2: **for** $t_i = 1, 2, \ldots, T_i$ **do**
3: Sample Gaussian noise $\epsilon \sim N(0, \sigma^2 I)$;
4: Compute input with Eq. (8).
5: **end for** *// Reverse diffusion process*
6: Initialize: $x_{T_i} \sim N(0, \sigma^2 I)$
7: **for** $t_i = T_i, \ldots, 1$ **do**
8: Predict noise $\epsilon_{t_i}(x_{t_i}, c, t_i)$;
9: Estimate x_{t_i} with Eq. (10).
10: **end for**
11: **end**



(a) Memory forgetting in the brain (b) Unlearning of FU

Fig. 2. The unlearning process.

APPENDIX B

B. 1. Proof of Theorem 1

In the proposed FedCMM, the diffusion model enhances the global model's generalization ability by iteratively adding and removing noise to generate data. However, generating samples or model states may expose sensitive information, particularly after data augmentation, increasing the risk of privacy leakage. To address this, differentially private Gaussian noise is introduced, which effectively defends against reverse attacks, reducing privacy risks from queries, and enhances the randomness and robustness of privacy protection in generated data. This approach ensures a balance between model performance improvement and privacy.

Gaussian noise is random noise generated from a Gaussian distribution, whose probability density function $p(x)$ is given by

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{x^2}{2\sigma^2}). \tag{11}$$

Theorem 1. To ensure (ε, δ)-differential privacy guarantees, it is typically necessary to use a sufficiently large standard deviation for the Gaussian noise. According to the theory of the Gaussian mechanism for controlling privacy leakage risks, the standard deviation σ can be expressed as

$$\sigma \leq \frac{\Delta f \sqrt{2\ln(1.25/\delta)}}{\varepsilon}, \tag{12}$$

where $\varepsilon \in [0.1, 10]$ is the privacy budget, $\delta \in [e^{-5}, e^{-10}]$ is the deviation parameter of differential privacy, and Δf is the local model influence of e_j. After FedCMM, Δf is

$$\Delta f = I_0 \cdot \frac{t}{t+\tau} e^{-\beta t} \left(e^{-\lambda(t-t_{\text{un}})} + (N-1) \cdot \frac{1}{1 + D_{\text{rem}}(0)\, e^{\zeta(t-t_{\text{un}})}} \right).$$

Proof. Let D_j^u denote the dataset of client e_j after the unlearning request (i.e., with the forgotten samples removed). Let D_{-j} denote the aggregated dataset of all remaining clients except e_j. The sensitivity Δf measures the worst-case change in the global objective when replacing D_j^u with D_{-j}:

$$\Delta f = \max_{D_j^u \sim D_{-j}} \|f(D_j^u) - f(D_{-j})\| = I_j(t), \tag{13}$$

which follows from the standard influence approximation that relates loss variation to the contribution of client e_j at round t.

In FedCMM, each client maintains its own local model and participates in the aggregation of the global model at fixed time intervals. However, over time, the network may experience catastrophic forgetting similar to the brain's intrinsic forgetting, leading to a gradual diminishing of the influence of individual models (as shown in Fig. 2). Therefore, referencing the neural forgetting mechanism, the influence of e_j on the global model $I_j(t)$ at round t before making an unlearning request is defined as

$$I_j(t) = I_0 \cdot \left(\frac{t}{t+\tau}\right) \cdot e^{-\beta t}, \tag{14}$$

where I_0 is the maximum influence. τ is the smoothing parameter, which controls how fast the influence grows. β is the attenuation coefficient, which controls how fast the influence decreases.

According to the influence trend in Fig. 3, there are three nodes in calculating the model influence after unlearning.

Node 1: Maximum influence time $t_{I_{\max}}$.

Taking the derivative of $I_j(t)$,

$$\frac{dI_j(t)}{dt} = I_0 \cdot \frac{e^{-\beta t}\left[(t+\tau) - t \cdot (1+\beta(t+\tau))\right]}{(t+\tau)^2}. \tag{15}$$

Setting the derivative to zero, we get $t_{I_{\max}}$,

$$t_{I_{\max}} = \frac{-\tau + \sqrt{\tau^2 + \frac{4\tau}{\beta}}}{2}. \tag{16}$$

Node 2: Convergence time t_{converge}.

To analyze the convergence of the global model, we determine the convergence time t_{converge} based on the variation of the global loss function $f(w^t)$. When the variation of the loss function becomes smaller than a predefined threshold a, the global model is considered to have converged,

$$\left|f(w^{t+1}) - f(w^t)\right| < a. \tag{17}$$

By monitoring the variation of $f(w^t)$, the convergence round t_{converge} of the global model can be identified. Typically, the maximum influence round $t_{I_{\max}}$ of a single client typically occurs earlier than t_{converge}, as the client's influence on the global model is significant initially but diminishes over time. Therefore, the global model requests additional training rounds to converge. We assume that $t_{I_{\max}}$ and t_{converge} satisfy the following relationship,

$$t_{\text{converge}} = \gamma \cdot t_{I_{\max}}, \tag{18}$$

where γ is a certain multiple ($\gamma > 1$).

Node 3: Unlearning time t_{un} and updated influence function.

After the global model reaches convergence at t_{converge}, an unlearning request may be issued at time t_{un}. To describe the influence change due to (i) targeted unlearning of the requesting client and (ii) diffusion-based augmentation applied to the remaining clients, we introduce a multiplicative unlearning modifier $g_{\text{un}}(t)$ and update the influence as

$$I_{j,\text{new}}(t) \;=\; I_0 \cdot \frac{t}{t+\tau}\, e^{-\beta t}\; g_{\text{un}}(t), \tag{19}$$

here, $g_{\text{un}}(t)$ decomposes into the contribution from the target (forgotten) client, $g_{\text{un,tar}}(t)$, and the contribution from the remaining clients after diffusion augmentation, $g_{\text{un,rem}}(t)$:

$$g_{\text{un}}(t) \;=\; g_{\text{un,tar}}(t) \;+\; g_{\text{un,rem}}(t). \tag{20}$$

We define these two terms as follows:

- Target-client decay (exponential decay starting at t_{un}):

$$g_{\text{un,tar}}(t) = \begin{cases} 1, & t \le t_{\text{un}}, \\ e^{-\lambda(t-t_{\text{un}})}, & t > t_{\text{un}}, \end{cases} \tag{21}$$

  where $\lambda > 0$ is the decay-rate parameter controlling how fast the target client's influence is reduced after the unlearning request.
- Remaining-clients mitigation (modeled via an increasing data-diversity term $D_{\text{rem}}(t)$ produced by diffusion-based augmentation):

$$g_{\text{un,rem}}(t) \;=\; \frac{1}{1 + D_{\text{rem}}(t)}. \tag{22}$$

  We model the augmented-data factor $D_{\text{rem}}(t)$ as growing (from its value at the unlearning time) exponentially:

$$D_{\text{rem}}(t) = \begin{cases} D_{\text{rem}}(0), & t \le t_{\text{un}}, \\ D_{\text{rem}}(0)\, e^{\zeta\,(t-t_{\text{un}})}, & t > t_{\text{un}}, \end{cases} \tag{23}$$

  where $D_{\text{rem}}(0) \ge 0$ denotes the baseline (pre-unlearning) effective data-diversity contributed by the remaining clients and $\zeta > 0$ is the augmentation rate produced by the diffusion model.

Combining the above, for $t > t_{\text{un}}$ we have the concise expression

$$\begin{aligned} I_{j,\text{new}}(t) = I_0 \cdot \frac{t}{t+\tau}\, e^{-\beta t} \Big(& e^{-\lambda(t-t_{\text{un}})} \\ & + (N-1) \cdot \frac{1}{1 + D_{\text{rem}}(0)\, e^{\zeta(t-t_{\text{un}})}} \Big). \end{aligned} \tag{24}$$

For $t \le t_{\text{un}}$, $I_{j,\text{new}}(t) = I_j(t)$ (i.e. no change). Therefore, substituting $\Delta f(t) = I_j(t)$ into the standard Gaussian mechanism directly yields the upper bound on the required noise standard deviation σ. □

B. 2. Proof of Theorem 2

In FL, the local objective function of each client e_i is denoted as $f(w_i)$, and the global objective function is

$$f(w) = \frac{1}{N} \sum_{i=1}^{N} f(w_i), \tag{25}$$

The model update under FedCMM is:

$$w^{t+1} = w^t - \eta(g^t + \rho^t) + \epsilon^t, \tag{26}$$

where η is the learning rate, g^t represents the conventional gradient descent estimate, ρ^t denotes the gradient ascent perturbation of InterFor; $\epsilon^t \sim \mathcal{N}(0, \sigma_\epsilon^2 I)$ is Gaussian noise injected for differential privacy and it satisfies:

$$\sigma_\epsilon^2 = \frac{2d\ln(1.25/\delta)}{\varepsilon^2} \sum_{j=1}^{K} I_j^2(t). \tag{27}$$

Assumption 1 (L-smoothness). For any objective function $f(w)$ that is L-smooth, we have

$$f(w^*) \leq f(w^t) + \langle \nabla f(w), w^* - w^t \rangle + \frac{L}{2}\|w^* - w^t\|^2. \quad (28)$$

Assumption 2 (Unbiased estimation and bounded variance). The gradient computed by any client j is an unbiased estimate of the local gradient, and the variance of the stochastic gradient is bounded:

$$\mathbb{E}[g^t] = \nabla F(w^t), \quad (29)$$

$$\mathbb{E}\|g_j^t - \nabla F_j(w^t)\|^2 \leq \sigma_g^2. \quad (30)$$

Assumption 3 (Bounded gradient of InterFor). The variance of gradient updates in InterFor is upper bounded:

$$\mathbb{E}\|g_j^t\|^2 \leq \sigma_\rho^2. \quad (31)$$

Assumption 4 (Bounded gradient noise). The variance of stochastic noise in IntriFor is bounded:

$$\mathbb{E}\|\epsilon_i^t\|^2 \leq \sigma^2, \quad (32)$$

where $\mathbb{E}[\epsilon_i^t] = 0$.

Let w^* denote the global optimal solution, such that $\nabla f(w^*) = 0$. The convergence analysis is as follows.

Theorem 2. If Assumptions 1 - 4 hold and $\eta \leq (1-\gamma_1)/2L$, we can conclude

$$\begin{aligned} &\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \\ &\leq \frac{F(w^0) - F(w^*)}{\eta C_1 T} + \frac{2L\eta\sigma_g^2}{C_1} + \frac{L\eta\sigma_\rho^2}{C_1} + \frac{L\sigma^2}{2\eta C_1}. \end{aligned} \quad (33)$$

Proof. Let $\Delta w = w^{t+1} - w^t$. According to the updated rule and L-smoothness, taking the expectation yields, we have

$$\begin{aligned} &\mathbb{E}[F(w^{t+1})] \\ &\leq \mathbb{E}[F(w^t)] + \mathbb{E}\langle \nabla F(w^t), \Delta w \rangle + \frac{L}{2}\mathbb{E}\|\Delta w\|^2 \\ &= \mathbb{E}[F(w^t) - \eta \underbrace{\langle \nabla F(w^t), g^t + \rho^t \rangle}_{T_1} + \frac{L\eta^2}{2}\underbrace{\|g^t + \rho^t\|^2}_{T_2} \\ &+ \frac{L}{2}\underbrace{\|\epsilon^t\|^2}_{T_3}], \end{aligned} \quad (34)$$

where $\mathbb{E}\|\epsilon^t\| = 0$.

Using Assumption 2 and defining the perturbation term ρ^t, we have

$$\mathbb{E}\langle \nabla F(w^t), g^t + \rho^t \rangle = \|\nabla F(w^t)\|^2 + \mathbb{E}\langle \nabla F(w^t), \rho^t \rangle. \quad (35)$$

To control the complexity of analysis, assume that the angle between ρ^t and $\nabla F(w^t)$ satisfies:

$$\mathbb{E}\langle \nabla F(w^t), \rho^t \rangle \geq -\gamma_1 \|\nabla F(w^t)\|^2, \quad (36)$$

where $\gamma_1 \in [0, 1)$ indicates the degree to which InterFor suppresses the downward trend. Thus,

$$T_1 \leq (1-\gamma_1)\|\nabla F(w^t)\|^2. \quad (37)$$

Using the parallelogram law for norms and independence,

$$\begin{aligned} \mathbb{E}\|g^t + \rho^t\|^2 &= \mathbb{E}\|g^t\|^2 + \mathbb{E}\|\rho^t\|^2 + 2\mathbb{E}\langle g^t, \rho^t \rangle \\ &\leq \mathbb{E}\|g^t\|^2 + \sigma_\rho^2 + 2\sqrt{\mathbb{E}\|g^t\|^2 \cdot \sigma_\rho^2} \\ &\leq \mathbb{E}\|g^t\|^2 + \sigma_\rho^2 + \mathbb{E}\|g^t\|^2 + \sigma_\rho^2 \\ &= 2\mathbb{E}\|g^t\|^2 + 2\sigma_\rho^2. \end{aligned} \quad (38)$$

Using the Cauchy-Schwarz and Young inequalities,

$$\begin{aligned} \mathbb{E}\|g^t\|^2 &= \mathbb{E}\|g^t - \nabla F(w^t) + \nabla F(w^t)\|^2 \\ &\leq 2\sigma_g^2 + 2\|\nabla F(w^t)\|^2. \end{aligned} \quad (39)$$

Then, we obtain

$$T_2 \leq 4\|\nabla F(w^t)\|^2 + 4\sigma_g^2 + 2\sigma_\rho^2. \quad (40)$$

Using the Assumption 4, we have

$$T_3 \leq \sigma_\epsilon^2. \quad (41)$$

Combining these results, we get:

$$\begin{aligned} \mathbb{E}[F(w^{t+1})] &\leq F(w^t) - \eta(1-\gamma_1)\|\nabla F(w^t)\|^2 \\ &+ \frac{L\eta^2}{2}(4\|\nabla F(w^t)\|^2 + 4\sigma_g^2 + 2\sigma_\rho^2) + \frac{L\sigma^2}{2}, \end{aligned} \quad (42)$$

Taking the sum and then averaging on both sides for $t = 0, \ldots, T-1$, and let $C_1 = 1 - \gamma_1 - 2L\eta$, when $\eta \leq (1-\gamma_1)/2L$,

$$\begin{aligned} &\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla F(w^t)\|^2 \\ &\leq \frac{F(w^0) - F(w^*)}{\eta C_1 T} + \frac{2L\eta\sigma_g^2}{C_1} + \frac{L\eta\sigma_\rho^2}{C_1} + \frac{L\sigma^2}{2\eta C_1} \\ &\leq \mathcal{O}\left(\frac{1}{T}\right). \end{aligned} \quad (43)$$

It is proved that FedCMM is convergent. □

APPENDIX C

C. 1. Evaluation setup

1) Implementation: We implement the FedCMM prototype in a real-world distributed environment consisting of a PC as the server and five Jetson AGX Xavier devices serving as the client nodes. The server runs Ubuntu 18.04.6 and is equipped with dual NVIDIA A40 GPUs and an Intel Xeon Gold 6226 CPU. Each AGX device is equipped with an 8-core ARM CPU and runs Ubuntu 18.04. All devices communicate wirelessly via Python-based socket packages, forming a heterogeneous federated learning system. The entire framework is developed in Python 3.10 with PyTorch 2.3. The prototype system is illustrated in Fig. 3.

2) Datasets and models: To evaluate the performance of FedCMM, we conduct extensive experiments on five benchmark datasets: MNIST, FMNIST, CIFAR-10, CIFAR-100 and TinyImageNet. A CNN is utilized for MNIST and FMNIST, and ResNet18 is adopted for CIFAR-10, CIFAR-100, and TinyImageNet. Conditional diffusion models are employed for data augmentation on remaining clients. We adopt a lightweight U-Net with three downsampling/upsampling blocks and 64 base channels. The diffusion process uses T=50

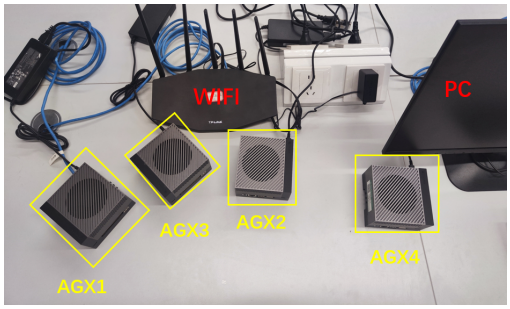


Fig. 3. User-differentiated federated unlearning prototype system.

steps with a cosine noise schedule under a class-conditional setting. As this augmentation is executed once offline prior to FL training, it introduces no additional computation or communication overhead during subsequent FL rounds. The parameter settings are detailed as follows: $N = 5$, mini-batch size $B = 128$, $I_0 = 1$, $\tau = 3$, $\beta = 1$, $\lambda = 0.05$.

3) Comparison methods: We compare our approach with six approaches: (1) Retrain, which retrains the model from scratch to achieve unlearning; (2) Continue to Train (Contrain), which removes the target client and continues training with the remaining clients; (3) FedEraser, which adjusts model parameters based on historical updates from retained clients; (4) Flipping, which performs unlearning by training local models with label-flipped data samples; (5) SIFU, which mitigates data influence through random noise perturbation of global parameters; and (6) FedOSD, which applies constrained gradient updates along directions orthogonal to the retained clients' gradients.

4) Evaluation metrics: The evaluation encompasses three key dimensions: unlearning effectiveness, model utility, and computational efficiency. Specifically: (1) Effectiveness is quantified via Clean-Acc (accuracy on retained test data) and Backdoor-Acc (accuracy on attack data associated with the target client), with ideal unlearning characterized by high Clean-Acc and near-zero Backdoor-Acc; (2) Model utility is further assessed via cosine similarity between the unlearning model and the retrained model; (3) Efficiency is measured by the speed-up ratio in unlearning time relative to the retrained baseline. In the subsequent sections, we present detailed experimental analyses, including ablation studies, evaluation of unlearning quality, similar between the unlearned and Retrain model, efficiency comparisons, scalability under varying client populations, and performance with multiple unlearning requirements.

5) Implementation Details: The experiment consists of four phases: FL training, unlearning, post-unlearning, and returning. During the FL training, the CS and clients perform standard FL training. In the unlearning phase, client 0 is designated as the target client removed from the FL process, and injected with backdoor samples at a ratio of 0.9. client 0 withdraws and requests the removal of its influence from the global model. The post-unlearning phase involves the remaining clients continuing FL training with the CS, while in the returning phase, client 0 reenters the FL process. The inclusion of a returning phase models a realistic scenario where a client withdraws from the federation (e.g., for privacy concerns) and requests data removal, but later decides to rejoin. This scenario also serves as a robustness test: after forgetting the client's influence, the global model should still retain the ability to efficiently relearn the client's knowledge. The dataset is split into 90% for training and 10% for validation. Training rounds for each phase are: 50, 5, 15, and 15 for MNIST and FMNIST; 100, 10, 30, and 30 for CIFAR-10, CIFAR-100 and TinyImageNet.

Table 3. Performance (%) comparison of different methods on CIFAR-10

Methods	IU		PU		IR		Time(s)
	CA	BA	CA	BA	CA	BA	
Case 1	**75.96**	98.29	**80.82**	38.96	76.21	98.21	1144.28
Case 2	64.20	0.43	78.99	9.69	75.37	98.12	3053.97
FedCMM	70.83	**0.24**	80.45	**9.37**	**77.42**	**98.39**	**801.58**

IU: In Unlearning PU: Post Unlearning IR: In Returning
CA: Clean-Acc BA: Backdoor-Acc

C. 2. Ablation study

To evaluate the contribution of each component in FedCMM, we conduct an ablation study by systematically removing the InterFor and IntriFor modules, yielding two simplified variants: Case 1 (FedCMM w/o InterFor) and Case 2 (FedCMM w/o IntriFor). The evaluation is performed on the CIFAR-10 dataset under three stages: Unlearning (IU), Post-Unlearning (PU), and Returning (IR), with both Clean-Acc (CA) and Backdoor-Acc (BA) reported in Table 3.

We first examine Case 1, in which the InterFor module is disabled, meaning that no explicit unlearning mechanism is applied. Consequently, the Backdoor-Acc remains high across all stages, indicating that the influence of the target client is largely retained. However, the Clean-Acc remains relatively high in both the IU and PU stages, suggesting that the overall model utility is well preserved. These results confirm that without InterFor, the model fails to forget malicious behaviors but benefits from IntriFor, which enhances utility through internal gradient modulation among retained clients.

In Case 2, the IntriFor module is removed while InterFor is retained for unlearning. Compared to Case 1, this variant achieves a significant reduction in Backdoor-Acc during IU and PU, demonstrating that InterFor is effective in mitigating the backdoor influence of the target client. However, this gain comes at the cost of a notable degradation in Clean-Acc, highlighting the absence of a compensatory mechanism to maintain model utility during forgetting. These results indicate that InterFor alone is insufficient to strike a balance forgetting and performance retention.

Finally, FedCMM achieves the most favorable balance between unlearning effectiveness and model utility. It significantly reduces Backdoor-Acc to 0.24%, while maintaining strong Clean-Acc and achieving the lowest overall runtime. These results validate that InterFor is essential for effective unlearning, while IntriFor plays a complementary role in stabilizing and restoring model performance.

Table 4. Hyperparameter λ sensitivity analysis of FedCMM on CIFAR-10

λ	CA (%)			BA (%)		
	IU	PU	IR	IU	PU	IR
0.01	72.09	79.82	76.12	1.14	12.75	98.03
0.05	**70.83**	**80.45**	**77.42**	**0.24**	**9.37**	**98.39**
0.1	68.37	78.94	75.16	0.58	15.27	97.86

Table 5. Hyperparameter ζ sensitivity analysis of FedCMM on CIFAR-10

ζ	CA (%)			BA (%)		
	IU	PU	IR	IU	PU	IR
0.01	68.72	78.16	75.26	0.87	14.35	98.14
0.05	**70.83**	**80.45**	**77.42**	**0.24**	**9.37**	**98.39**
0.1	67.35	76.91	74.25	0.41	16.76	97.91

Table 6. Hyperparameter T sensitivity analysis of FedCMM on CIFAR-10

T	CA (%)			BA (%)		
	IU	PU	IR	IU	PU	IR
20	68.77	77.69	75.13	0.69	13.44	98.05
50	**70.83**	**80.45**	**77.42**	**0.24**	**9.37**	**98.39**
100	69.14	78.36	75.87	0.43	11.78	98.13

IU: In Unlearning PU: Post Unlearning IR: In Returning

CA: Clean-Acc BA: Backdoor-Acc

C. 3. Hyperparameter sensitivity analysis of FedCMM

We conduct a hyperparameter sensitivity analysis to evaluate how the decay rate λ, the augmentation rate ζ, and the diffusion sampling steps T influence the performance of FedCMM. These three hyper-parameters control the forgetting intensity in InterFor, the diversity of augmented representations in IntriFor, and the generation quality of diffusion-based augmentation, respectively. The experiments are performed on CIFAR-10, and the results are summarized in Tables 4 to 6.

From Table 4, we observe that the decay rate λ has a clear impact on both forgetting effectiveness and model utility. A moderate decay rate ($\lambda = 0.05$) achieves the most desirable balance, maintaining a high Clean-Acc while substantially reducing Backdoor-Acc across the IU, and PU stages. When λ is set too low (e.g., 0.01), the forgetting force applied to the target client becomes insufficient, leading to noticeably higher Backdoor-Acc. Conversely, setting λ too high (e.g., 0.1) causes overly aggressive forgetting, resulting in a drop in Clean-Acc due to excessive suppression of benign knowledge. A similar trend is observed for the augmentation rate ζ in Table 5. The optimal performance appears at $\zeta = 0.05$, where the model benefits from adequate augmentation diversity while avoiding unnecessary noise. Smaller values (e.g., 0.01) do not provide sufficient representational variation to counteract harmful patterns, whereas larger values (e.g., 0.1) inject excessive perturbations that impair Clean-Acc. Regarding diffusion sampling steps, Table 6 shows that $T = 50$ is the most effective setting. Fewer steps (e.g., $T = 20$) result in lower-quality synthetic samples that reduce the forgetting robustness, while excessive steps (e.g., $T = 100$) bring only marginal improvements yet significantly increase computational overhead.

Overall, these results reveal a consistent trend across all three hyper-parameters: extremely small values weaken the forgetting capability or reduce augmentation quality, whereas overly large values introduce unnecessary disturbance or computational cost. In contrast, moderate settings yield the most favorable balance between Clean-Acc and Backdoor-Acc. This confirms that careful hyperparameter selection is essential for ensuring both effective unlearning and reliable model performance in FedCMM.

Table 7. Performance (%) comparison on Returning phase

Methods	MNIST		FMNIST		CIFAR-10		CIFAR-100	
	CA	BA	CA	BA	CA	BA	CA	BA
Retrain	99.13	93.73	90.19	78.63	75.33	98.11	36.87	84.66
Contrain	99.16	99.77	90.36	97.87	75.71	98.27	33.29	94.71
FedEraser	99.05	92.27	88.87	82.39	75.09	97.56	34.13	89.30
Flipping	99.17	**99.48**	87.28	96.94	75.63	98.20	32.49	92.23
SIFU	99.10	92.37	89.39	82.74	76.01	97.42	36.60	89.61
FedOSD	99.03	93.75	88.42	96.09	75.12	96.95	34.12	89.30
FedCMM	**99.21**	**99.32**	**90.56**	**97.27**	**77.42**	**98.39**	**41.14**	**92.37**

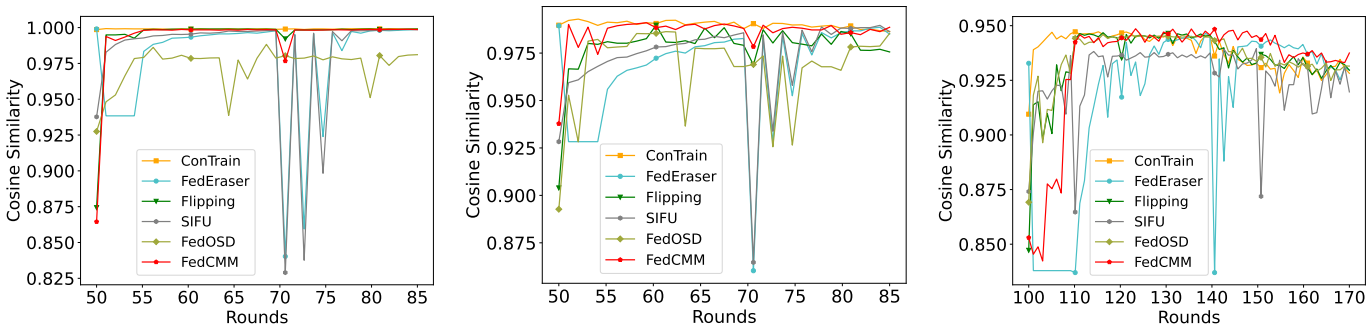
CA: Clean-Acc BA: Backdoor-Acc

C. 4. Evaluation of performance in returning phase

We evaluate the performance of each unlearning method during the returning phase, where a previously unlearned client reenters the federated learning process. This setting reflects a practical scenario in federated systems and also serves as a stress test for assessing whether the global model can correctly restore the client's knowledge without compromising overall generalization. Clean-Acc and Backdoor-Acc on MNIST, FMNIST, CIFAR-10, and CIFAR-100 are compared against six representative FU baselines, with results summarized in Table 7.

As shown in Table 7, FedCMM exhibits consistently strong performance across all datasets. On CIFAR-10, it achieves 77.42% Clean-Acc and 98.39% Backdoor-Acc, both exceeding all competing methods. A similar trend is observed on CIFAR-100, where FedCMM attains 41.14% Clean-Acc and 92.37% Backdoor-Acc. The higher Backdoor-Acc in this phase reflects a more complete restoration of the previously forgotten client's knowledge, while the high Clean-Acc indicates that this restoration does not harm the overall model utility. Compared to prior approaches such as Retrain, Contrain, and FedEraser, FedCMM provides both stronger recovery ability and better generalization.

Overall, the results demonstrate that the proposed FedCMM is highly resilient in the returning phase: it successfully reabsorbs the rejoining client's information and achieves the best balance between recovery completeness and global model performance. This highlights its robustness in dynamic federated environments where client participation may change over time.



(a) (b) (c)

Fig. 4. The cosine similarity between the unlearned and Retrain model on (a) MNIST, (b) FMNIST, and (c) CIFAR-10.

C. 5. The similar between the unlearned and Retrain model

In Fig. 4, we evaluate the impact of unlearning on the global model by measuring the cosine similarity between the unlearned model and the Retrain model on the backdoor test sets of MNIST, FMNIST, and CIFAR-10. A higher cosine similarity indicates a closer resemblance between the unlearned model and Retrain model.

Across different datasets, FedEraser, SIFU and FedOSD exhibit significant fluctuations, indicating that unlearning substantially affects the accuracy of the remaining clients. Contrain shows higher similarity to Retrain model by ignoring unlearning requests and continuing federated training thus minimally impacting the global model, particularly on MNIST. However, its performance on CIFAR-10 is suboptimal, indicating that increased dataset complexity exacerbates the impact on the global model. In contrast, FedCMM exhibits a more stable curve, achieving the highest similarity, indicating that our method closely approximates the Retrain model while minimizing the impact of unlearning on the global model.

C. 6. Evaluation with different client numbers

To assess the scalability of FedCMM, we conduct further experiments under varying client numbers on CIFAR-10. The results are shown in Fig. 5 and Fig. 6.

In Fig. 4, when the number of clients increases from 5 to 30, FedCMM maintains a high level of Clean-Acc, decreasing only slightly from 81.37% to 72.74%. In contrast, FedEraser's accuracy drops sharply to 31.19%. This robustness shows that FedCMM effectively mitigates the utility degradation typically caused by aggressive unlearning under large-scale participation. Meanwhile, FedCMM achieves low Backdoor-Acc across all configurations, remaining below 1.41% even with 30 clients. This performance is competitive with FedEraser's forgetting capability, yet FedCMM avoids the substantial utility loss observed in FedEraser. By maintaining strong forgetting efficacy without compromising global performance, FedCMM proves to be more balanced and reliable.

Efficiency-wise, FedCMM exhibits moderate time consumption that scales smoothly with the number of clients in Fig. 5. At 30 clients, its unlearning cost remains under 7.2×10^3 seconds, a substantial improvement over retraining (33.4×10^3 s) and more scalable than SIFU, FedOSD, and FedEraser.

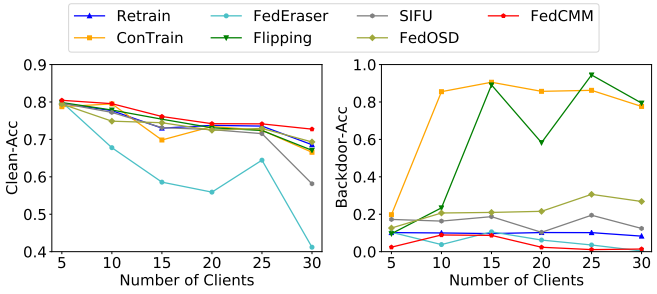


Fig. 5. Performance comparison of Clean-Acc and Backdoor-Acc on CIFAR-10.

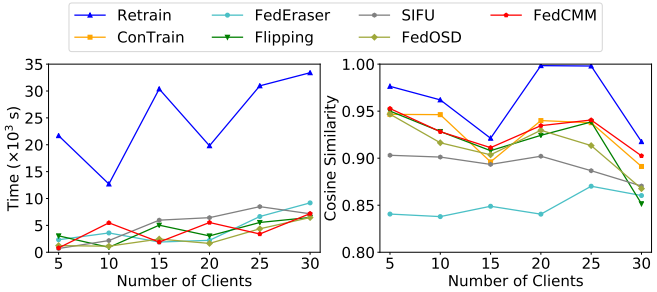


Fig. 6. Performance comparison of the unlearning time and cosine similarity on CIFAR-10.

This ensures practical deployability in real-world systems with limited resources or strict latency constraints. Moreover, FedCMM preserves model semantics throughout the unlearning process. Its cosine similarity with the original global model remains above 0.90 across all settings, indicating that the overall representation space is largely intact. Compared to FedEraser and SIFU, which exhibit greater semantic drift at larger client scales, FedCMM offers more stable and interpretable model updates.

C. 7. Evaluation with multiple unlearning requests

We further evaluate the capability of unlearning methods to handle multiple unlearning requests issued in sequence or overlap, a common scenario in practical deployments. Specifically, client 0 begins unlearning at round 101 and completes at round 110, while client 1 initiates its unlearning process

Table 8. Performance comparison with multiple unlearning on FMNIST

Methods	CA (%)	BA (%)	Time (s)
Retrain	88.34	2.64	591.67 (1.00×)
Contrain	88.62	94.16	/
FedEraser	84.05	8.94	106.18 (5.57×)
Flipping	85.67	8.45	86.91 (6.80×)
SIFU	86.68	6.81	**80.46 (7.35×)**
FedOSD	85.95	15.76	106.39 (5.56×)
FedCMM	**87.91**	**4.08**	82.49 (7.16×)

Table 9. Performance comparison with multiple unlearning on CIFAR-10

Methods	CA (%)	BA (%)	Time (s)
Retrain	80.08	3.94	43100.97 (1.00×)
Contrain	79.16	80.19	/
FedEraser	71.83	18.61	5973.16 (9.10×)
Flipping	76.18	6.11	2171.54 (19.85×)
SIFU	78.19	11.94	1418.73 (30.38×)
FedOSD	72.94	20.43	1492.08 (28.91×)
FedCMM	**81.93**	**0.84**	**1387.17 (31.07×)**

Table 10. Performance comparison with multiple unlearning on CIFAR-100

Methods	CA (%)	BA (%)	Time (s)
Retrain	44.92	0.05	32674.57 (1.00×)
Contrain	43.85	40.97	/
FedEraser	37.17	13.20	4942.15 (6.61×)
Flipping	41.55	4.91	4631.37 (7.05×)
SIFU	43.76	2.48	3326.10 (9.82×)
FedOSD	41.03	10.94	3648.86 (8.95×)
FedCMM	**45.19**	**0.06**	**3058.30 (10.68×)**

CA: Clean-Acc BA: Backdoor-Acc

at round 108 and finishes at round 117. In this setting, the forgetting process must simultaneously remove the influence of each target client while preserving the retained model knowledge or avoiding conflicts between concurrent forgetting operations. The results of Clean-Acc, Backdoor-Acc and time on different datasets FMNIST, CIFAR-10, and CIFAR-100 are presented in Tables 8 to 10, respectively.

Experimental results show that methods such as Retrain and Contrain fail to achieve this balance: Retrain incurs prohibitive computational overhead, while Contrain preserves clean accuracy but entirely neglects the removal of sensitive information. Parameter adjustment methods like FedEraser and FedOSD partially mitigate individual forgetting requests but suffer from cumulative performance degradation or residual influence when multiple forgetting rounds occur. SIFU achieves efficient unlearning by leveraging rollback, but its forgetting effect fluctuates under overlapping requests, indicating instability in maintaining global model consistency.

In contrast, FedCMM achieves both stable clean accuracy and effective forgetting across multiple sequential unlearning requests. By decoupling the forgetting and utility restoration processes, FedCMM mitigates gradient interference during unlearning and preserves the global model's integrity across multiple unlearning rounds. FedCMM achieves improvements in global model accuracy from 1.23% to 10.09% compared to the SOTA federated unlearning methods. Moreover, its runtime remains efficient and scales well with increasing request complexity, demonstrating practical advantages over retraining-based or rollback-based methods in multi-client unlearning scenarios.