# Consensus Learning with Multi-Party Perturbation Triggers for Secure Model Access

**Yizhun Zhang[1], Jie Huang[1,2*], Zeping Zhang[1], Shuaishuai Zhang[1], Changhao Ding[1],Xuan Chen[1]**

[1]School of Cyber Science and Engineering, Southeast University
[2]Purple Mountain Laboratories
{zhangyizhun,jhuang,zhangzp9970,sszhang,230239281,chenxuan49}@seu.edu.cn

## Abstract

With the widespread deployment of deep learning models in multi-party collaborative scenarios, the issues of secure model access control and intellectual property (IP) protection have become increasingly critical. To address the limitations of existing methods that lack proactive defense mechanisms in such settings, this paper introduces a novel paradigm Consensus Learning which enables fine-grained control over model execution permissions via a multi-party joint authorization mechanism. Building on this, we propose the Collaborative Perturbation Trigger Method (CPTM), which allows participating parties to collaboratively generate perturbation-based trigger data that embed identity features. The model can only be activated using the collectively constructed trigger, enforcing tightly bound access control without modifying the model architecture. Extensive experiments on CIFAR-10, CIFAR-100, MNIST, and Face-LFW datasets demonstrate that the proposed method maintains prediction accuracy within 2% of the baseline unprotected models on authorized data. In contrast, under unauthorized or adversarial inputs, model accuracy drops below 10%, showcasing strong access control capabilities and robustness. This study offers a novel direction for building scalable, robust, and proactively protected deep learning models in multi-party collaborative environments.

## Introduction

With the rapid advancement and widespread adoption of deep learning technologies, deep neural models have increasingly evolved into valuable digital assets (He et al. 2022; Mu et al. 2024). Against the backdrop of growing concerns over data security and intellectual property (IP) protection, a pressing challenge emerges: how to effectively prevent unauthorized duplication, misuse, and leakage of proprietary information (Li et al. 2022a; Shen et al. 2025), while ensuring fine-grained, secure, and controllable management of model usage in collaborative multi-party environments (Peigné et al. 2025; Ma, Yao, and Xu 2024; Li et al. 2025; Hao, Zhang, and Li 2025). Addressing this challenge has become a critical issue that demands innovative breakthroughs.

To address the above challenges, researchers have proposed a variety of model protection mechanisms. Xue et al. embedded adversarial perturbations into model weights to construct a dynamic obfuscation mechanism, making it difficult for attackers to reconstruct model behavior even when partial parameters are exposed (Xue et al. 2023). Ren et al. developed an availability assurance strategy that integrates identity authentication with access control, enabling dynamic monitoring of model invocation behaviors (Ren et al. 2024). Li et al. introduced SecureNet, which combines backdoor learning with trigger mechanisms to enhance intellectual property (IP) control over deep models (Li et al. 2024). These approaches have demonstrated improvements in model privacy and robustness, particularly showing promising results in single-party deployment scenarios.

However, in multi-party collaborative scenarios such as federated learning, ensuring model security becomes significantly more challenging (Cao, Jia, and Gong 2021; Liao et al. 2025; Yi et al. 2025). Federated learning aims to enhance data privacy by performing local training and aggregating parameters, thereby avoiding centralized exposure of raw data. Despite this advantage, the resulting global model must still be transmitted and stored in plaintext on central servers or communication channels. If any participating party is compromised, or if malicious insiders succeed in extracting model copies during training, the shared model becomes highly susceptible to unauthorized replication, misuse, or redistribution. As a result, the original privacy benefits of federated learning can be severely undermined.

To address the threats of unauthorized access and misuse in collaborative deep learning, we propose a novel Consensus Learning Paradigm. This paradigm enforces that model activation and execution must be jointly authorized by all participating parties—no single party can operate the model independently. During training, each participant embeds identity-bound trigger signals, which collectively form a unified and indivisible authorization structure.A model follows the Consensus Learning Paradigm if it satisfies: **(1)Joint Authorization:** All parties must contribute valid trigger factors for activation; missing or forged components render the model inactive. **(2)Collaborative Embedding:** Trigger signals are individually embedded during training to define secure, distributed access control. Unlike conventional federated learning, which lacks strict inference-time access control, Consensus Learning ensures execution rights are strongly bound to multi-party cooperation. Even if one

*Corresponding authors.

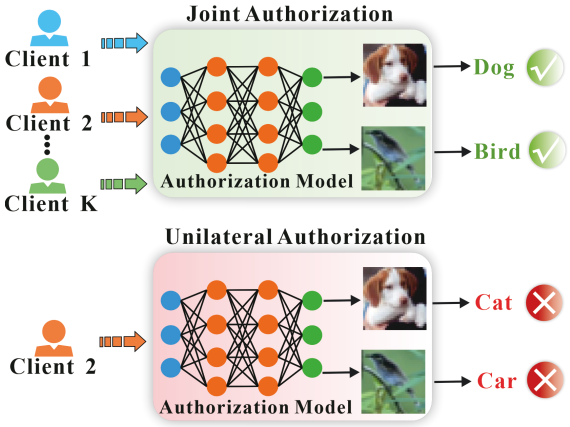



Figure 1: Consensus Learning Conceptual Diagram.

party is compromised, the model remains inaccessible without all valid authorizations. As shown in Figure 1, this design intrinsically resists misuse, enhances security and robustness, and establishes a trustworthy framework for secure multi-party collaboration.

To support the Consensus Learning paradigm, we introduce the Collaborative Perturbation Trigger Method (CPTM). CPTM enables each participant to sequentially inject identity-bound perturbations into shared data, producing Consensus Triggered Data (CTD) that encodes multi-party authorization. CTD serves as the access credential, tightly binding model execution to all involved parties.In a typical five-client federated setting (A-E), CPTM constructs perturbed inputs in a fixed order. Each participant applies affine transformations and nonlinear noise to embed personal identity features. The process begins with A and continues through B, C, and D, ending with E. The final image integrates perturbations from all parties, ensuring indivisible and verifiable access control.

The main contributions of this paper are as follows:

- **Consensus Learning Paradigm:** We propose Consensus Learning, a collaborative paradigm that binds model execution to jointly authorized data from all participants. Without modifying the model architecture, it enforces access control under full authorization, enhancing security and robustness in multi-party settings.
- **Collaborative Perturbation Trigger Method (CPTM):** We introduce CPTM, a chained perturbation injection method that generates Consensus Triggered Data (CTD) as identity-bound credentials. CTD binds model access to full multi-party authorization, enabling secure, distributed control.
- **Empirical Validation:** Extensive experiments across datasets and models show that our method enforces secure access with negligible performance loss, confirming its effectiveness and generality.

# Related Work

## Passive Protection Mechanisms

With the growing deployment of deep learning in vision and language tasks, models have become valuable digital assets, raising concerns over their security and IP protection (Lao et al. 2022; Zong et al. 2024; Li et al. 2022b). Existing works primarily focus on passive defenses that verify ownership post misuse. Uchida et al. proposed a weight-based watermarking method that embeds ownership into fully connected layer outputs during training, ensuring imperceptibility (Uchida et al. 2017). To enhance robustness, Chen et al. introduced a backpropagation-based uniform embedding that distributes watermark signals broadly in the weight space (Chen, Rohani, and Koushanfar 2018). Wang et al. proposed RIGA, embedding watermarks into redundant substructures and using trigger sets for verification, improving resistance to structural changes (Wang and Kerschbaum 2021).

Although effective for ownership verification, existing methods are post hoc and lack proactive control. Once stolen, models can still be reused or modified, highlighting the need for execution-restrictive protection.

## Unilateral Proactive Protection Mechanisms

To overcome the limitations of passive mechanisms in access control, recent studies have shifted toward proactive protection methods that embed control logic within the model itself. These approaches ensure that a model performs correctly only when specific authorization conditions are satisfied, thereby enabling real-time defense and usage restrictions. Chen et al. encrypted model weights to ensure functionality only with valid keys (Chen and Wu 2018). To enhance deployment-level restrictions, Alam et al. introduced DeepLock, which combines encryption with access control policies (Alam et al. 2024). However, such encryption-based solutions often impose significant computational and energy overhead, making them less suitable for resource-constrained environments such as edge devices. To mitigate this issue, Alam et al. later presented NN-Lock, a lightweight framework that incorporates simplified encryption schemes and authorization protocols to reduce overhead (Alam et al. 2022). In a different line of work, Li et al. proposed SecureNet, which embeds fixed image triggers into training data. After deployment, the model only generates valid outputs when inputs contain authorized triggers, enabling active verification and secure model operation (Li et al. 2024).

Prior proactive methods rely on single-party authorization. If triggers or keys are leaked, adversaries can bypass controls. Without multi-party binding, these methods are vulnerable to single-point failures and insider attacks, limiting their effectiveness in collaborative learning.

# Methodology

## Threat Model

This study focuses on the problem of model access control in multi-party collaborative learning scenarios. We assume that

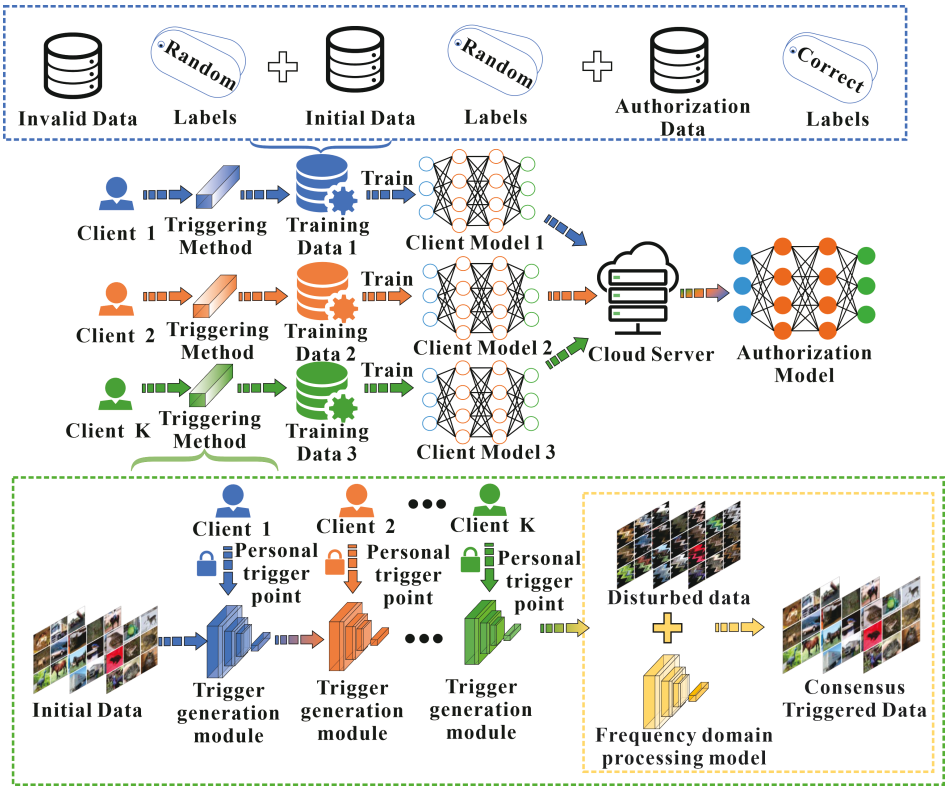



Figure 2: Overall Framework Diagram.

the trained model is jointly owned by multiple participants, and its usage should be strictly governed by consensus-based authorization from all involved parties. The adversary's objective is to bypass this multi-party authorization mechanism and activate the model without obtaining permission from all stakeholders, thereby enabling unauthorized usage, illicit deployment, or intellectual property theft.

**Adversary Capability Assumptions:** (1) Single-Party Information Leakage: The adversary may gain access to one participant's local trigger data, private samples, perturbation parameters, or generation protocols. (2) Model Acquisition: The adversary may obtain a complete copy of the trained model via transmission interception or server compromise. (3) Partial Trigger Forgery: The adversary may attempt to craft forged inputs by leveraging accessible image samples and known trigger patterns to elicit a response from the model.

**Security Assumptions:** (1) Local Perturbation Generation: Each participant independently generates and embeds perturbations on their local data without the need for centralized coordination. (2) Decentralized Perturbation Secrecy: Perturbation parameters and generation logic are neither stored on a central server nor shared through public communication channels.

**Defense Objective:** The system is designed to ensure that the model can only be activated upon receiving a complete and valid set of multi-party consensus-based trigger information. If any participant fails to authorize, or if the perturbation information is missing or forged, the model remains inactive and produces no prediction output. This guarantees fine-grained access control and enables proactive defense against unauthorized usage.

## Overall Framework

The overall framework of this study is illustrated in Figure 2 and consists of three core components: the Collaborative Perturbation Trigger Method (CPTM), dataset construction strategies, and the model training pipeline.

**Collaborative Perturbation Trigger Method (CPTM):** Each participant applies identity-bound perturbations sequentially based on their local private data. Through a chained collaborative process, Consensus Triggered Data (CTD) integrating multi-party features is generated. CTD exhibits two key properties: **intra-user diversity** (varying perturbations across different samples of the same user) and **inter-user uniqueness** (distinct perturbations on the same sample across different users). The resulting data is highly irreversible and identity-bound, serving as the sole valid credential for model access.

**Dataset Structuring:** To enhance the model's ability to distinguish valid perturbation patterns and improve robustness, we construct three contrasting types of training datasets. **(1)Invalid Set:** Contains images embedded with random or forged perturbations paired with incorrect labels. This set simulates scenarios where malicious attackers attempt to fabricate authorized inputs. **(2)Initial Set:** Com-

prises images without any perturbation information and with randomly assigned labels. It is used to train the model to remain inactive when exposed to clean, unmodified inputs. **(3)Authorized Set:** Consists of images embedded with valid perturbations collaboratively generated by all participants, accompanied by correct labels. This is the only set permitted to activate the model's predictive functionality.

**Model Training:** Each participant trains a local model on private data, with periodic parameter aggregation via a federated scheme. This yields a unified model that embeds multi-party perturbation signals while preserving structural and training consistency. The resulting model reliably recognizes Consensus Triggered Data (CTD) and enforces secure access control.

## Collaborative Perturbation Trigger Method

To support secure access in Consensus Learning, we propose a Collaborative Perturbation Trigger that integrates spatial perturbations with frequency-domain embeddings. This mechanism ensures federated dependency, irreversibility, and tamper resistance, enabling model activation only upon full-party consensus.The detailed procedure of this method is presented in Algorithm 1.

**Single-Stage Affine Transformation Mechanism.** The objective of the single-stage affine transformation is to construct a set of affine matrices based on three pairs of point correspondences in the image space, thereby applying spatial perturbations to the image.

Let the input image be denoted as $X \in \mathbb{R}^{H\times W\times C}$, and a client $A_i$ randomly selects three pairs of coordinate points.

$$\begin{cases} Source\ point\ set: S_i = \left\{p_1^i, p_2^i, p_3^i\right\} \\ Target\ point\ set: T_i = \left\{q_1^i, q_2^i, q_3^i\right\} \end{cases} \tag{1}$$

The affine matrix $M_i \in \mathbb{R}^{2\times 3}$ is computed based on these three pairs of points, satisfying:

$$\begin{bmatrix} q_1^i & q_2^i & q_3^i \end{bmatrix} = M_i \begin{bmatrix} p_1^i & p_2^i & p_3^i \\ 1 & 1 & 1 \end{bmatrix} \tag{2}$$

The affine matrix $M_i$ is applied to the image to obtain the perturbed image:

$$X' = \mathrm{A}(X, M_i) \tag{3}$$

Here, $\mathrm{A}(\cdot,\cdot)$ denotes the affine transformation operator.

**Multi-Stage Joint Affine Perturbation Mechanism.** In consensus learning, the model trigger must be collaboratively constructed by multiple clients. We introduce a multi-stage joint affine mechanism, in which the perturbations from all clients are sequentially applied to the image to generate a collaboratively triggered image.

Assume there are $N$ clients, denoted as $A_1, A_2, ..., A_N$. Let the image at each stage be defined as follows: the initial image is $X_0 = X$,the image at stage $i$ is $X_i = \mathrm{A}(X_{i-1}, M_i)$, and the final output image is:

$$\begin{aligned} X_N &= \mathrm{A}(X_{N-1}, M_N) \\ &= \mathrm{A}(\cdots \mathrm{A}(\mathrm{A}(X_0, M_1), M_2), \cdots, M_N) \end{aligned} \tag{4}$$

Algorithm 1: Collaborative Perturbation Trigger Generation

**Input**: Original image $X \in \mathbb{R}^{H\times W\times C}$, Number of clients $N$, Three pairs of affine control points per client: $\left\{\left(p_1^i, q_1^i\right), \left(p_2^i, q_2^i\right), \left(p_3^i, q_3^i\right)\right\}, i \in [1, N]$, Nonlinear perturbation parameters: $\alpha, \beta, \gamma, \phi$, High-frequency patch size: $h, w$, Fixed noise image $N_{fixed} \in \mathbb{R}^{H\times W\times C}$

**Output**: Consensus triggered image $X^{trigger}$

Initialize $X_0 \leftarrow X$.
**for** each client $i = 1$ to $N$ **do**
  Compute affine matrix: $M_i \leftarrow solve\_affine(p_1^i, p_2^i, p_3^i, q_1^i, q_2^i, q_3^i)$
  Apply affine transformation: $X_i \leftarrow affine\_transform(X_{i-1}, M_i)$
  **for** each pixel coordinate $(x, y)$ and channel $c \in \{1, \cdots, C\}$ **do**
    Compute nonlinear perturbation: $\delta \leftarrow \alpha \cdot \sin(\beta x + \gamma y + \phi)$
    Inject nonlinear noise: $X_i(x, y, c) \leftarrow X_i(x, y, c) + \delta$
    Add fixed noise component: $X_i(x, y, c) \leftarrow X_i(x, y, c) + N_{fixed}(x, y, c)$
  **end for**
**end for**
Obtain perturbed image: $X_N \leftarrow X_N$
Compute Fourier transforms: $\hat{X}_0 \leftarrow FFT(X_0), \hat{X}_N \leftarrow FFT(X_N)$
Extract high-frequency patch: $\hat{X}_{HF} \leftarrow \hat{X}_N[H-h:H, W-w:W]$
Replace frequency patches in $\hat{X}_0$:
$\hat{X}_0[0:h, 0:w] \leftarrow \hat{X}_{HF}$
$\hat{X}_0[H-h:H, 0:w] \leftarrow \hat{X}_{HF}$
$\hat{X}_0[0:h, W-w:W] \leftarrow \hat{X}_{HF}$
$\hat{X}_0[H-h:H, W-w:W] \leftarrow \hat{X}_{HF}$
Inverse Fourier transform to obtain final image: $X^{trigger} \leftarrow \Re\left(IFFT\left(\hat{X}_0\right)\right)$
**return** $X^{trigger}$

Here, $M_i$ represents the affine matrix constructed by client $A_i$ based on its selected coordinate points, $X_i$ denotes the image result at stage $i$, and $N$ is the total number of clients participating in consensus learning. This process constructs spatially asymmetric and cumulatively irreversible image perturbations.

The affine point sets $\{(S_i, T_i)\}_{i=1}^N$ individually selected by all clients collectively constitute the "Consensus Authorization Key" for trigger image generation, with security guarantees reflected in: **Distributed Confidentiality:** Each party retains access only to its own point set, remaining invisible to others. **Chained Coupling:** Each transformation stage depends on the output of the previous one, making the perturbation process indivisible. **Attack Resistance:** Full reconstruction requires access to all affine matrices $\{M_1, \cdots, M_n\}$; leakage from all parties is necessary, and the chain breaks if any remain private. **High Brute-forHigh Brute-force Complexity:** Even with only three point pairs

per client, the search space is vast. With $N$ clients applying sequential affine transformations, the joint space grows exponentially. For example, in a 32x32 image, the brute-force complexity can be expressed as:

$$Total = \left(C\left(1024, 3\right) \times C\left(1024, 3\right)\right)^{N} \tag{5}$$

For example, when $N = 5$, the brute-force complexity exceeds $2^{270}$, far surpassing the widely accepted $2^{64}$ security threshold, making brute-force attacks infeasible.

**Nonlinear Perturbation Injection Mechanism.** Affine transformations are linear operations that can be easily recognized or suppressed by models. To enhance the randomness and unpredictability of the trigger signals, we introduce nonlinear perturbations at the pixel level of the image. This improves the model's perception difficulty and robustness against attacks. For each pixel coordinate $(x, y)$, a sinusoidal perturbation is defined as follows:

$$\delta(x, y) = \alpha \cdot \sin(\beta x + \gamma y + \phi) \tag{6}$$

Here, $\alpha$ denotes the perturbation amplitude, controlling the intensity of the perturbation; $\beta$ and $\gamma$ are frequency factors that determine the periodicity of the perturbation texture; and $\phi$ is a phase constant used to randomize the perturbation pattern. This perturbation term is added to each channel of the image's pixel values, and the update is formulated as follows:

$$X_i(x, y, c) \leftarrow X_i(x, y, c) + \delta(x, y), \forall c \in \{1, 2, ..., C\} \tag{7}$$

To further enhance the irreversibility of the triggered image, we inject a fixed structured noise image $N_{fixed} \in \mathbb{R}^{H \times W \times C}$ into the perturbed output. This noise remains identical across all images, is independent of the original input. As a result, it serves as a content-agnostic "structural lock" embedded within the image.

$$X_i(x, y, c) \leftarrow X_i(x, y, c) + \delta(x, y) + N_{fixed}(x, y, c) \tag{8}$$

**Frequency-Domain Trigger Embedding Mechanism.** To improve security and reduce detectability, we introduce a frequency-domain embedding mechanism that injects high-frequency components of perturbed images into selected regions of the original image's spectrum, enabling cross-domain feature fusion and model-aware activation.

Let the input image be denoted as $X_0 \in \mathbb{R}^{H \times W \times C}$ and the final perturbed image as $X_N \in \mathbb{R}^{H \times W \times C}$. We first perform a two-dimensional Fourier transform on both images independently for each channel.

$$\begin{aligned} \hat{X}_0^{(c)} &= F[X_0^{(c)}] \\ &= \sum_{x=0}^{H-1} \sum_{y=0}^{W-1} X_0^{(c)}(x, y) \cdot e^{-2\pi i(\frac{ux}{H} + \frac{vy}{W})} \end{aligned} \tag{9}$$

The extracted high-frequency components are duplicated into the four corners of the original image's frequency spectrum. This operation preserves the primary frequency structure of the original image while superimposing the high-frequency texture patterns from the perturbed image. These patterns serve as identifiable cues for the model's internal trigger detection mechanism, as illustrated in Algorithm 1.

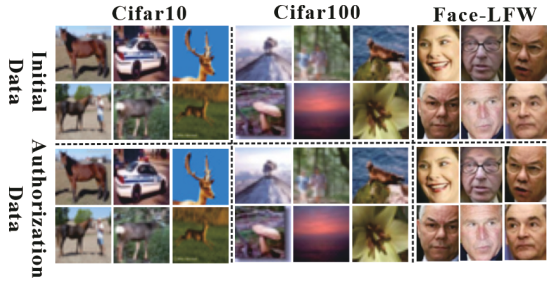



Figure 3: Comparison between Consensus Triggered Data and Original Data.

Finally, the frequency-domain image is transformed back into the spatial domain by applying the inverse Fourier transform. The corresponding formulation is as follows:

$$\begin{aligned} X^{trigger}(x, y, c) &= F^{-1}[\hat{X}_0^{(c)}] \\ &= \frac{\sum_{\mathrm{u}=0}^{H-1} \sum_{v=0}^{W-1} \hat{X}_0^{(c)}(u, v) \cdot e^{2\pi i(\frac{ux}{H} + \frac{vy}{W})}}{HW} \end{aligned} \tag{10}$$

Let $F^{-1}[\cdot]$ denote the inverse Fourier transform, and $X^{trigger}$ represent the final output image, referred to as the Consensus Triggered Data (CTD).

# Experiment

## Experimental Setup

All experiments were conducted using PyTorch on a high-performance platform with multiple NVIDIA Tesla V100 GPUs. Evaluations used diverse public datasets such as MNIST, CIFAR-10, CIFAR-100, and Face-LFW, covering tasks from digit classification to face recognition to ensure robustness. We used common architectures including VGG16, ResNet18, DenseNet121, and WideResNet50-2 to assess the method's generalizability across varied network structures.

## Model Performance Evaluation

To systematically evaluate the effectiveness of the proposed method in terms of data stealthiness and access control capability, we conducted a series of comparative experiments. First, to assess whether the visual differences between the generated Consensus Triggered Data and the original data are perceptible to the human eye, we designed a visualization experiment based on image comparison, as shown in Figure 3. This experiment aims to determine whether the perturbations introduced by the trigger mechanism at the image level are sufficiently imperceptible to prevent adversaries from identifying the model authorization method through visual inspection or heuristic detection.

As shown in the experimental results in Figure 3, the authorized (consensus-triggered) images exhibit no significant

visual differences compared to the original input across various datasets. This indicates that the generated consensus triggered data effectively preserves visual indistinguishability while enforcing model access control. Such indistinguishability makes it difficult for external observers or adversaries to detect or infer the presence of embedded triggers. These findings validate the effectiveness of the proposed method in ensuring data stealthiness.

To evaluate model performance under varied authorization and validate the Collaborative Perturbation Trigger Mechanism (CPTM), we design a multi-party experiment involving five participants. This setup assesses model accuracy on different datasets under collective access control. Acc-Clean is the accuracy of the unprotected model on clean data; Acc-Pro is the protected model's accuracy on consensus triggered data; Acc-Origin is its accuracy on original clean data.Acc-Origin is the accuracy of the protected model on the original clean dataset.

The experimental results (Table 1) show that Acc-Pro achieves performance comparable to or even better than Acc-Clean, indicating that consensus-triggered data enhances feature learning. In contrast, Acc-Origin remains below 10%, demonstrating strong suppression of unauthorized inputs. These results highlight the effectiveness of the proposed collaborative perturbation trigger in improving authorized access while blocking unauthorized interference, with strong potential for secure multi-party model deployment.

To evaluate the effectiveness of our multi-party fused perturbation trigger, we conduct t-SNE visualization on ResNet-18 trained with CIFAR-10. As shown in Figure 4, features from consensus-authorized data form well-separated clusters, while unauthorized data appears randomly scattered. This demonstrates the method's robustness and reliability in access control.

| Dataset | Networks | Acc-Clean | Acc-Pro | Acc-Origin |
|---|---|---|---|---|
| MNIST | VGG16 | 99.30 | **99.38** | 11.35 |
| | ResNet18 | 99.44 | **99.47** | 11.21 |
| | DenseNet | **99.57** | 99.06 | 10.37 |
| | WideResNet | **99.52** | 99.21 | 9.97 |
| CIFAR-10 | VGG16 | **89.01** | 88.84 | 1.31 |
| | ResNet18 | 93.08 | **93.33** | 0.56 |
| | DenseNet | 95.39 | **93.45** | 0.69 |
| | WideResNet | 94.40 | **95.31** | 0.39 |
| CIFAR-100 | VGG16 | **70.19** | 67.63 | 0.53 |
| | ResNet18 | 75.34 | **75.41** | 0.77 |
| | DenseNet | **78.72** | 76.31 | 0.17 |
| | WideResNet | **80.67** | 79.81 | 0.15 |
| Face-LFW | VGG16 | **95.70** | 94.05 | 0.87 |
| | ResNet18 | **95.04** | 94.17 | 0.99 |
| | DenseNet | **97.36** | 97.35 | 0.76 |
| | WideResNet | 96.53 | **97.19** | 0.85 |

Table 1: Prediction Results Comparison Table

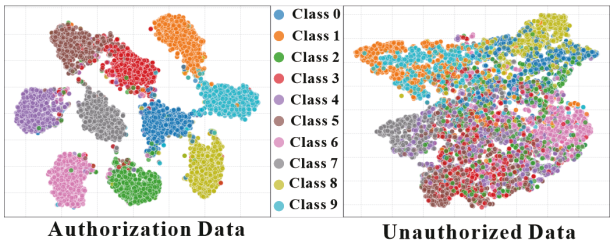



Figure 4: t-SNE Visualization Analysis.

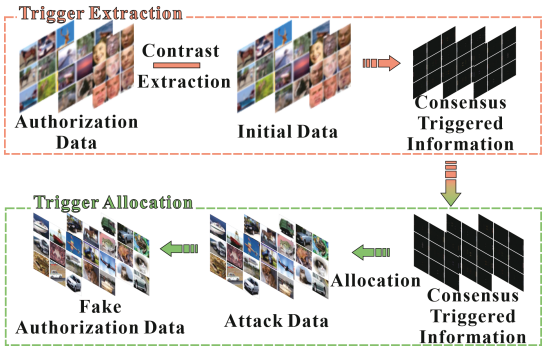



Figure 5: Trigger Extraction Attack Flowchart.

## Evaluation of Model Robustness

**Trigger Extraction Attack Experiments.** To evaluate the robustness of the proposed Collaborative Perturbation Trigger Method, we designed an adversary extraction experiment to assess the model's defense capability under mismatched authorization conditions. In this experiment, the adversary is assumed to have access to partial consensus triggered data and original data, attempting to extract perturbation features and apply them to their attack test samples to bypass the model's access control (the attack procedure is illustrated in Figure 5). The adversary extracts trigger features from the training set and applies them onto their own data in an attempt to illicitly activate the model.

As shown in Figure 6, across all four models and datasets, the attacker's prediction accuracy remains consistently low. Although the attacker attempts to extract and transfer consensus trigger information, the strong binding between the trigger and the original data features prevents any significant improvement in model response. These results demonstrate that the proposed multi-party fused perturbation mechanism exhibits strong robustness and resistance to transferability, effectively preventing unauthorized data from activating the model and thereby enhancing the security of access control.

**Robustness Experiments.** To further evaluate the robustness of the proposed method, we design multiple attack scenarios simulating adversaries with varying degrees of authorization leakage. Specifically, Acc-single denotes the case where the attacker possesses the perturbation information from a single client to generate trigger data, while Acc-fusion simulates an attacker acquiring partial information (e.g., from three out of five clients) and attempting to synthesize fused trigger samples. Experimental results demonstrate

| Dataset | Networks | Acc-Origin | Acc-Single | Acc-Fusion |
|---|---|---|---|---|
| MNIST | VGG16 | 11.35 | 11.60 | 10.34 |
| | ResNet18 | 11.21 | 12.17 | 10.32 |
| | DenseNet | 10.37 | 17.16 | 10.42 |
| | WideResNet | 9.97 | 15.58 | 10.38 |
| Cifar-10 | VGG16 | 1.31 | 5.12 | 15.08 |
| | ResNet18 | 0.56 | 1.90 | 4.35 |
| | DenseNet | 0.69 | 2.26 | 5.63 |
| | WideResNet | 0.39 | 2.83 | 7.29 |
| Cifar-100 | VGG16 | 0.53 | 1.67 | 1.41 |
| | ResNet18 | 0.77 | 1.89 | 1.10 |
| | DenseNet | 0.17 | 1.27 | 1.02 |
| | WideResNet | 0.15 | 1.00 | 1.27 |
| Face-LFW | VGG16 | 0.87 | 0.91 | 2.71 |
| | ResNet18 | 0.99 | 1.16 | 3.55 |
| | DenseNet | 0.76 | 1.27 | 1.16 |
| | WideResNet | 0.85 | 1.13 | 1.73 |

Table 2: Results Table of Robustness Experiments

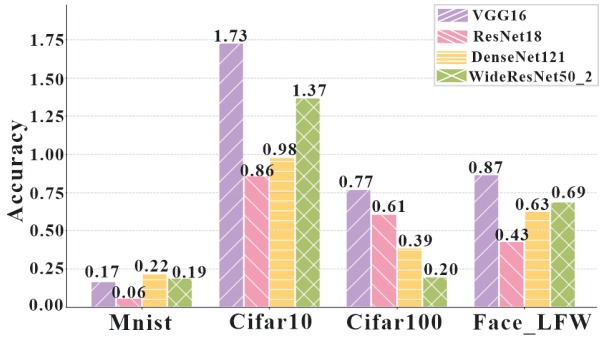


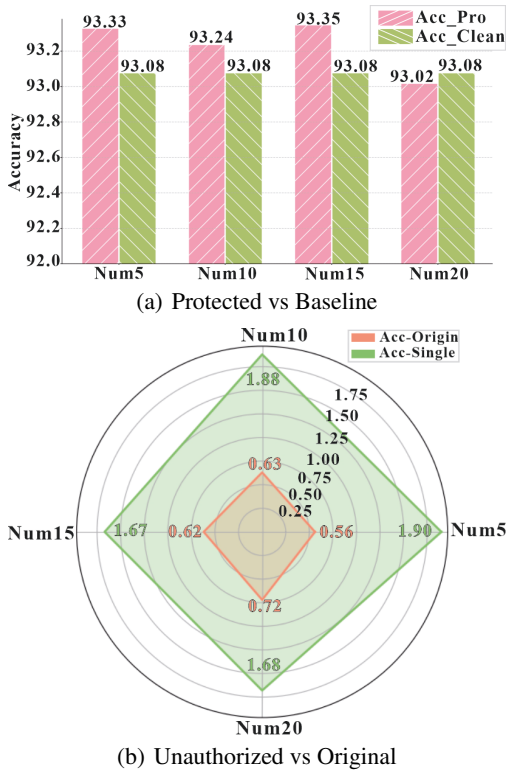

Figure 6: Prediction Results of Trigger Extraction Attack.

that the model maintains strong defense performance under partial leakage, validating the effectiveness and security of the proposed multi-party fusion perturbation mechanism in access control scenarios.

As shown in Table 2, Acc-Origin, Acc-Single, and Acc-Fusion remain consistently low, suggesting that partial authorization offers limited effect. The model effectively suppresses unauthorized triggers and fails to recover predictions without full-party input, confirming the robustness of the proposed fusion mechanism against partial leakage.

### Impact of the Number of Participating Clients on Model Performance

To assess the scalability and robustness of the proposed multi-party fusion perturbation trigger mechanism, we conducted experiments using the ResNet-18 model on the CIFAR-10 dataset with 5, 10, 15, and 20 clients. As shown in Table 3 and Figure 7, the model maintains a prediction accuracy (Acc-Pro) comparable to the baseline accuracy on clean data (Acc-Clean), even as the number of participating clients increases. This demonstrates that our method effectively preserves the model's performance while significantly resisting unauthorized access, highlighting its strong adapt-




(a) Protected vs Baseline

(b) Unauthorized vs Original

Figure 7: Effect of Client Number on Model Performance.

ability and security in collaborative environments.

| Num | Acc-Pro | Acc-Origin | Acc-Single | Acc-clean |
|---|---|---|---|---|
| 5 | 93.33 | 0.56 | 1.90 | 93.08 |
| 10 | 93.24 | 0.63 | 1.88 | 93.08 |
| 15 | 93.35 | 0.62 | 1.67 | 93.08 |
| 20 | 93.02 | 0.72 | 1.68 | 93.08 |

Table 3: Performance under multi-client settings

## Conclusion

This study addresses the challenges of model theft and unauthorized usage in multi-party collaborative deep learning environments by proposing a consensus learning paradigm. Based on this paradigm, we design a Collaborative Perturbation Triggering Mechanism (CPTM) to achieve secure access control. The mechanism relies on multi-party joint authorization, ensuring that the model is activated only when all participants reach unanimous agreement, thereby effectively preventing unilateral misuse and unauthorized access. Experimental results demonstrate that the proposed method exhibits strong robustness and security against partial authorization leakage, as well as notable scalability and stability across varying numbers of participants. This work provides novel insights and practical solutions for secure model access control and intellectual property protection in multi-party collaborative scenarios.

## Acknowledgments

This research was supported by the Purple Mountain Laboratories for Network and Communication Security. We thank the Big Data Computing Center of Southeast University for providing us with the supercomputing platform and the anonymous reviewers for their comments on the submission version of this work.

## References

Alam, M.; Saha, S.; Mukhopadhyay, D.; and Kundu, S. 2022. NN-Lock: A Lightweight Authorization to Prevent IP Threats of Deep Learning Models. *J. Emerg. Technol. Comput. Syst.*, 18(3).

Alam, M.; Saha, S.; Mukhopadhyay, D.; and Kundu, S. 2024. Deep-Lock: Secure Authorization for Deep Neural Networks. arXiv:2008.05966.

Cao, X.; Jia, J.; and Gong, N. Z. 2021. Provably Secure Federated Learning against Malicious Clients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8): 6885–6893.

Chen, H.; Rohani, B. D.; and Koushanfar, F. 2018. DeepMarks: A Digital Fingerprinting Framework for Deep Neural Networks. arXiv:1804.03648.

Chen, M.; and Wu, M. 2018. Protect Your Deep Neural Networks from Piracy. In *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–7.

Hao, Z.; Zhang, B.; and Li, H. 2025. DCHM: Dynamic Collaboration of Heterogeneous Models Through Isomerism Learning in a Blockchain-Powered Federated Learning Framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(16): 17077–17084.

He, X.; Xu, Q.; Lyu, L.; Wu, F.; and Wang, C. 2022. Protecting Intellectual Property of Language Generation APIs with Lexical Watermark. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10): 10758–10766.

Lao, Y.; Zhao, W.; Yang, P.; and Li, P. 2022. DeepAuth: A DNN Authentication Framework by Model-Unique and Fragile Signature Embedding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9): 9595–9603.

Li, P.; Huang, J.; Wu, H.; Zhang, Z.; and Qi, C. 2024. SecureNet: Proactive intellectual property protection and model security defense for DNNs based on backdoor learning. *Neural Networks*, 174: 106199.

Li, Y.; Zhu, L.; Jia, X.; Jiang, Y.; Xia, S.-T.; and Cao, X. 2022a. Defending against Model Stealing via Verifying Embedded External Features. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2): 1464–1472.

Li, Y.; Zhu, L.; Jia, X.; Jiang, Y.; Xia, S.-T.; and Cao, X. 2022b. Defending against Model Stealing via Verifying Embedded External Features. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(2): 1464–1472.

Li, Z.; Bao, H.; Guan, M.; Pan, H.; Huang, C.; and Dai, H.-N. 2025. EBS-CFL: Efficient and Byzantine-robust Secure Clustered Federated Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(17): 18593–18601.

Liao, D.; Gao, X.; Xu, Y.; and Xu, C.-Z. 2025. Progressive Distribution Matching for Federated Semi-Supervised Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(5): 5191–5199.

Ma, Y.; Yao, Y.; and Xu, X. 2024. PPIDSG: A Privacy-Preserving Image Distribution Sharing Scheme with GAN in Federated Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13): 14272–14280.

Mu, X.; Wang, Y.; Huang, Z.; Lai, J.; Zhang, Y.; Wang, H.; and Yu, Y. 2024. EncryIP: A Practical Encryption-Based Framework for Model Intellectual Property Protection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19): 21438–21445.

Peigné, P.; Kniejski, M.; Sondej, F.; David, M.; Hoelscher-Obermaier, J.; Schroeder de Witt, C.; and Kran, E. 2025. Multi-Agent Security Tax: Trading Off Security and Collaboration Capabilities in Multi-Agent Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(26): 27573–27581.

Ren, G.; Wu, J.; Li, G.; Li, S.; and Guizani, M. 2024. Protecting Intellectual Property With Reliable Availability of Learning Models in AI-Based Cybersecurity Services. *IEEE Transactions on Dependable and Secure Computing*, 21(2): 600–617.

Shen, Y.; Zhuang, Z.; Yuan, K.; Nicolae, M.-I.; Navab, N.; Padoy, N.; and Fritz, M. 2025. Medical Multimodal Model Stealing Attacks via Adversarial Domain Alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(7): 6842–6850.

Uchida, Y.; Nagai, Y.; Sakazawa, S.; and Satoh, S. 2017. Embedding Watermarks into Deep Neural Networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, ICMR '17, 269–277. New York, NY, USA: Association for Computing Machinery. ISBN 9781450347013.

Wang, T.; and Kerschbaum, F. 2021. RIGA: Covert and Robust White-Box Watermarking of Deep Neural Networks. In *Proceedings of the Web Conference 2021*, WWW '21, 993–1004. New York, NY, USA: Association for Computing Machinery.

Xue, M.; Wu, Z.; Zhang, Y.; Wang, J.; and Liu, W. 2023. AdvParams: An Active DNN Intellectual Property Protection Technique via Adversarial Perturbation Based Parameter Encryption. *IEEE Transactions on Emerging Topics in Computing*, 11(3): 664–678.

Yi, L.; Yu, H.; Ren, C.; Wang, G.; Liu, X.; and Li, X. 2025. pFedES: Generalized Proxy Feature Extractor Sharing for Model Heterogeneous Personalized Federated Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(21): 22146–22154.

Zong, W.; Chow, Y.-W.; Susilo, W.; Baek, J.; Kim, J.; and Camtepe, S. 2024. IPRemover: A Generative Model Inversion Attack against Deep Neural Network Fingerprinting and Watermarking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7): 7837–7845.