Pointillism: Probing-Based Model Compatibility for Robust Collaborative Machine Learning

Yang Li[1], **Adams Wai Kin Kong**[1]
[1]College of Computing and Data Science,
Nanyang Technological University, Singapore
{YANG047,AdamsKong}@ntu.edu.sg

Abstract

Assessing model compatibility is a key challenge in collaborative machine learning. Heterogeneous data distributions can lead to discrepancies among independently trained models that degrade performance when combined, while adversarial manipulation can introduce harmful behaviors. In both cases, reliable evaluation prior to model merging is essential. Existing approaches mostly rely on parameter-space statistics, which do not reflect model behavior and are often unreliable in non-IID settings. Inspired by the pointillism art style, where images are formed from small, structured dots of color, we propose *Pointillism*, a probing-based framework that evaluates model compatibility directly in function space without requiring task data. The method uses structured, randomized probes sampled from an out-of-distribution space to elicit model responses. From these responses, we construct compact model signatures that capture both global prediction characteristics and class-level features. Compatibility is assessed through feature consensus, measuring cross-model agreement on class-level representations via probe transfer. This provides a behavior-centric and fine-grained view of model alignment. We apply the framework to federated learning, where consensus-based selection identifies a self-consistent subset of client models for aggregation. Experiments in federated learning demonstrate improved robustness against both untargeted and backdoor attacks, outperforming existing defense methods. The code is available at `https://github.com/yangland/pointillism`.

1 Introduction

Assessing model compatibility is a critical problem in collaborative machine learning. This challenge arises in settings such as federated learning [1] and its decentralized variants [2–5], where model merging over weights is commonly used to combine knowledge across distributed participants. In practice, each participant contributes to model training under heterogeneous conditions; some contributions may be incompatible due to non-IID data, while others may be adversarial. Without careful evaluation, incompatible or malicious contributions can degrade performance or introduce hidden behaviors such as backdoors.

Existing approaches for assessing model compatibility primarily rely on model weights or update statistics, such as ℓ_1 distance [6], ℓ_2 distance [7, 8], cosine similarity [9, 10] or consistency between rounds [11]. While effective in certain settings, these methods exhibit fundamental limitations. First, under non-IID data, benign models can naturally diverge in parameter space, making such statistical signals unreliable [8, 12–14]. Second, model parameters do not fully capture model functionality; carefully crafted malicious models can appear statistically normal while embedding backdoor behaviors[15, 16]. Third, in high-dimensional parameter spaces, small but adversarially aligned perturbations can evade detection, making parameter-based signatures less discriminative [17,

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Pointillism artwork (b) Local magnification (c) $d = 4$ mask (d) $d = 4$ probe (e) $d = 8$ mask (f) $d = 8$ probe

Figure 1: Illustration of the pointillism-inspired probing design. (a) *L'Hirondelle Steamer on the Seine*, Paul Signac, 1901 [19]. (b) Local magnification on (a) showing dot-based structure. (c)–(f) Example binary masks and rendered probes at two grid resolutions.

18]. Together, these limitations indicate that parameter-space analysis provides an incomplete and potentially misleading characterization of model compatibility.

Despite these limitations, directly evaluating model behavior in collaborative settings remains challenging. In federated learning, the underlying task data is typically unavailable. As a result, model behavior cannot be reliably assessed. Consequently, prior approaches resort to parameter statistics as indirect proxies for model compatibility. This raises a key question: *Can model compatibility be assessed directly at the level of model behavior, prior to merging?*

In this work, we introduce *Pointillism*, a probing-based framework for assessing model compatibility through functional consensus without the need for task data. This name is inspired by the art technique *pointillism*, where images are formed from small, distinct dots of color (see Fig. 1), reflecting our use of structured probe patterns. The framework consists of three key components. First, *Pointillism Probing* generates structured, randomized inputs from an out-of-distribution space. These inputs are constructed using grid-based masks with controlled sparsity and color assignments. Second, *Model Signature* construction summarizes model responses into a compact representation that captures both global prediction behavior and class-level features. Third, *Compatibility Assessment* evaluates model alignment by measuring feature consensus on the class labels via cross-model probe transfer. We then show that this design provides a behavior-centric view of model alignment and extends naturally to multi-model settings in federated learning, where consensus-based selection identifies a self-consistent subset for aggregation. The contributions of this paper are:

1. We propose *Pointillism*, a novel probing-based framework for model compatibility assessment that departs from conventional parameter-space analysis and instead evaluates models through their functional responses.
2. We analyze model behaviors under carefully designed synthetic inputs, extracting compact and interpretable representations of model functionality that capture informative signals for compatibility assessment.
3. We evaluate the proposed framework in federated learning settings under heterogeneous and adversarial conditions, demonstrating improved robustness over existing methods, achieving up to 87.8% reduction in attack success rate under backdoor attacks and 5.0% improvement in accuracy under untargeted attacks.

2 Related Work

Existing approaches for assessing model compatibility in federated learning primarily rely on parameter-space statistics. Krum [7] and RFA [8] identify outliers via ℓ_2 distance between model updates. FLAME [9] measures similarity in update directions via cosine similarity, while FoolsGold [10] relies on cosine similarity to detect collusion. Norm Bound [20] constrains the ℓ_2 norm of updates, and Coordinate-wise Median [21] takes the median across clients for each parameter dimension. FLDetector [11] detects anomalies via temporal inconsistency across rounds, FCDC [22] uses Fisher information to capture parameter importance discrepancies, AlignIns [23] analyzes update alignment and sign consistency, and FPD [24] incorporates cosine-based filtering. These methods characterize models through geometric or statistical relations in parameter space.

An alternative direction is to analyze models via reconstructed inputs, as in model inversion methods [25–27]. However, for model compatibility assessment, reconstructing semantically meaningful

inputs is unnecessary. Instead, it suffices to extract discriminative behavioral signals that reflect cross-model consistency. Moreover, such optimization-based procedures are computationally expensive, making them unsuitable for scalable collaborative settings.

DeepSight [28], the closest probing-based baseline, probes models using random inputs and analyzes output distributions for anomaly detection in federated learning. However, its random probes are unstructured and are not designed to elicit class-level representations; moreover, it still relies on parameter-based filtering. In contrast, our approach constructs structured probes and derives class-level representations, relying solely on probing-based signals for compatibility assessment.

3 Method

3.1 Pointillism Probing

We design the probing process to elicit informative model responses by stimulating the model with diverse inputs. Exhaustively enumerating the input space is computationally infeasible, as even a small input such as 32×32 yields $|\mathcal{X}| = 256^{1024}$. Conversely, while purely random inputs (e.g., Gaussian noise) lack the structure to induce discriminative features, some degree of randomness remains necessary to explore the input space and avoid sampling bias. To balance these requirements, we propose a probing scheme based on randomized structured inputs, combining controlled geometric patterns with stochastic variation.

Let the input have height H, width W, and $\mathcal{C}$ channels. Let C denote the number of classes. We construct each probe on a grid of varying resolution $d \in \mathcal{D}$. For each grid size and ratio $\rho \in \Omega$, a binary mask partitions the grid into two groups by activating $w(d, \rho) = \text{round}(\rho d^2)$ cells. Each group is then assigned a color sampled from the normalized input space with minimum distance $d_{\min}$, and per-cell values are obtained by adding small Gaussian variations ($\sigma_{\text{fg}}, \sigma_{\text{bg}}$) around the assigned group colors. This color assignment is performed as described in Algorithm 1 (line 4), with example probes illustrated in Fig. 1. This design introduces stochasticity through random color assignment while preserving structured geometric patterns induced by the mask. From the probe pool $\mathcal{P}$, we construct per-class candidate sets by selecting the top-M responses for each class:

$$\mathcal{V}_c = \operatorname*{Top}_{M,\, x \in \mathcal{P}} f(x)_c. \tag{1}$$

Here, Top_M returns the top-M probes, and $f(x)_c$ denotes the model's softmax output for class c. We then define the Top-K probe set for class c as

$$\mathcal{T}_c = \text{Top}_K\big\{x \in \mathcal{V}_c \mid \arg\max_{c'=1,\dots,C} f(x)_{c'} = c\big\}, \tag{2}$$

where probes are ranked by $f(x)_c$ and $K \leq M$. These sets form a compact representation of model behavior at the class level. In practice, limited sampling may lead to insufficient valid probes for certain classes. When $|\mathcal{T}_c| < K$, we perform local refinement by applying bounded edits (up to T_{edit} operations) to high-scoring candidates in $\mathcal{V}_c$, generating additional probes to improve coverage. Each edit modifies the binary mask via bounded cell-level operations: state flips toggle selected cells; activation/deactivation add or remove active cells; and on/off swaps relocate activity by deactivating one active cell and activating another inactive cell. This refinement is triggered selectively and operates under a small budget. The full procedure is summarized in Algorithm 1.

3.2 Model Signature

From Pointillism probing, we obtain the probe pool $\mathcal{P}$ and the Top-K probe sets $\{\mathcal{T}_c\}_{c=1}^{C}$. The probe pool provides broad coverage but is dominated by random inputs, while $\mathcal{T}_c$ captures only high-confidence responses. To overcome these limitations and extract a more informative representation, we define a *model signature* $\mathcal{S} = (\mathcal{S}_g, \mathcal{S}_c)$, consisting of global response statistics ($\mathcal{S}_g$) and class-level features ($\mathcal{S}_c$).

Global response statistics. For each grid size $d \in \mathcal{D}$, let $\mathcal{P}_d \subseteq \mathcal{P}$ be the probes generated at resolution d. The class distribution at resolution d is

$$q_c^{(d)} = \frac{1}{|\mathcal{P}_d|} \sum_{x \in \mathcal{P}_d} \mathbf{1}\left[\arg\max_{c'=1,\dots,C} f(x)_{c'} = c\right], \quad c = 1, \dots, C. \tag{3}$$

Algorithm 1 Pointillism Probing

Require: Model f, grid set $\mathcal{D}$, active-ratio schedule Ω, Top-K size K, candidate pool size M, edit budget T_{edit}, per-resolution probe budget B, color separation $d_{\min}$, perturbation scales $\sigma_{\text{fg}}, \sigma_{\text{bg}}$

Initialize $\mathcal{P} \leftarrow \emptyset$
for $d \in \mathcal{D}$, $\rho \in \Omega$ until B exhausted for each d **do** ▷ global probing
  Sample mask m with $\text{round}(\rho d^2)$ active cells
  Sample foreground and background colors $\mu_{\text{fg}}, \mu_{\text{bg}} \in [0,1]^{\mathcal{C}}$ with $\|\mu_{\text{fg}} - \mu_{\text{bg}}\|_2 \geq d_{\min}$, and render x from m with $\sigma_{\text{fg}}, \sigma_{\text{bg}}$
  Evaluate and cache $f(x)$, and add x to $\mathcal{P}$
end for
for $c = 1, \ldots, C$ **do** ▷ Top-K selection
  Construct $\mathcal{V}_c$ as Top-M probes ranked by $f(x)_c$ via Eq. (1)
  Construct $\mathcal{T}_c$ as Top-K probes classified as c via Eq. (2)
  if $|\mathcal{T}_c| < K$ **then** ▷ refinement
    Refine probes from $\mathcal{V}_c$ with up to T_{edit} edits
    Update $\mathcal{P}$ and recompute $\mathcal{V}_c$, $\mathcal{T}_c$
  end if
end for
return $\mathcal{P}$, $\{\mathcal{T}_c\}_{c=1}^{C}$

The distribution $q^{(d)}$ captures how the model maps probes to class labels, providing a global summary of its functional behavior under probing. The global component is $\mathcal{S}_g = \{q^{(d)}\}_{d \in \mathcal{D}}$.

Class-level features. The class-level component captures both representative and boundary-sensitive behavior. The Top-K sets $\{\mathcal{T}_c\}_{c=1}^{C}$ provide high-confidence probes for each class. To characterize decision boundaries, we further extract edge-case probes based on softmax ambiguity. For each probe x, let $f(x)_{(1)}$ and $f(x)_{(2)}$ denote the largest and second-largest softmax values, and define the margin $m(x) = f(x)_{(1)} - f(x)_{(2)}$. Let $\pi_1(x)$ and $\pi_2(x)$ be their corresponding class indices. For each class pair (p, q), we consider probes with $\{\pi_1(x), \pi_2(x)\} = \{p, q\}$ and select the K_e samples with smallest margins:

$$\mathcal{E}_{pq} = \text{Top}_{K_e}\big\{x \in \mathcal{P} \mid \{\pi_1(x), \pi_2(x)\} = \{p, q\}\big\}, \tag{4}$$

ranked by ascending $m(x)$. The class-level component is $\mathcal{S}_c = \big(\{\mathcal{T}_c\}_{c=1}^{C},\ \{\mathcal{E}_{pq}\}_{1 \leq p < q \leq C}\big)$.

3.3 Model Compatibility

Given two models f_i and f_j, we assess their compatibility using the model signature $\mathcal{S} = (\mathcal{S}_g, \mathcal{S}_c)$, evaluating their functional consistency under probing. The overall probing, signature construction, and compatibility assessment pipeline is illustrated in Fig. 2.

3.3.1 Model Diversity

The global signature $\mathcal{S}_g$ provides a measure of the intrinsic diversity of each model under probing. In particular, it captures whether the model produces non-degenerate responses across the label space. For each grid size $d \in \mathcal{D}$, we measure the diversity of the predicted-label distribution $q^{(d)}$ using its normalized entropy. We then define the model-level diversity score as

$$r(f_i) = \frac{1}{|\mathcal{D}|} \sum_{d \in \mathcal{D}} \frac{H\left(q_i^{(d)}\right)}{\log C}, \quad H\left(q_i^{(d)}\right) = -\sum_{c=1}^{C} q_{i,c}^{(d)} \log q_{i,c}^{(d)}. \tag{5}$$

Here, $q_i^{(d)}$ denotes the probe-label distribution of model f_i at resolution d. This score is high when probe responses are distributed across the label space and low when the model collapses most probes to a small subset of labels.

For a pair of models (f_i, f_j), the pairwise weight is defined as

$$r(f_i, f_j) = \sqrt{r(f_i)\, r(f_j)}, \tag{6}$$

which downweights comparisons involving models with low prediction diversity.

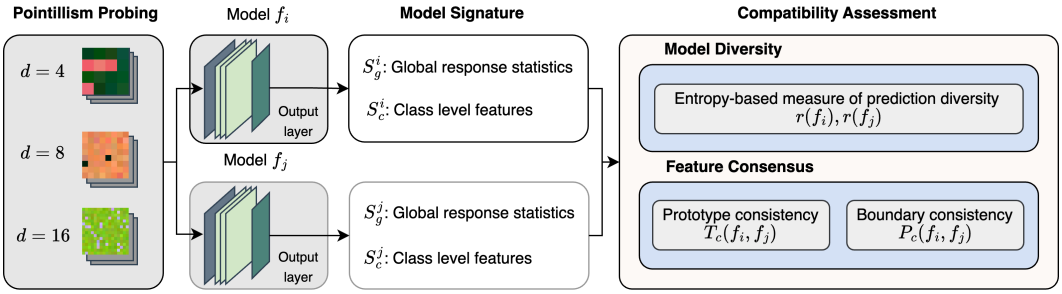


Figure 2: Overview of the proposed framework. Pointillism probing generates structured inputs to extract model signatures. Model feature compatibility is then assessed via (i) an entropy-based model diversity score capturing prediction diversity, and (ii) feature consensus measured by prototype and boundary consistency.

3.3.2 Feature Consensus

Feature consensus is evaluated by transferring probe sets between models and measuring whether their functional roles are preserved. For each class label c, we consider two complementary aspects: high-confidence prototypes and decision-boundary edge cases.

Prototype consistency. For each class c, the Top-K probe set $\mathcal{T}_c^{(f_i)}$ captures inputs that strongly activate class c under model f_i. We transfer these probes to f_j and measure whether they are still recognized as class c. The resulting score is defined as

$$T_c(f_i \to f_j) = \frac{1}{|\mathcal{T}_c^{(f_i)}|} \sum_{x \in \mathcal{T}_c^{(f_i)}} f_j(x)_c, \tag{7}$$

which quantifies how consistently representative features of class c are preserved across models.

Boundary consistency. We next examine whether decision boundaries are aligned. For each class pair (p, q), the edge-case set $\mathcal{E}_{pq}^{(f_i)}$ consists of probes that lie near the decision boundary between p and q under f_i. These probes are inherently ambiguous, simultaneously activating both classes. After transfer to f_j, we test whether this ambiguity is preserved—i.e., whether the probe remains jointly associated with classes p and q, rather than collapsing to a single class. We measure this using a pair-ambiguity score

$$A_{pq}^{(f_j)}(x) = \min\{f_j(x)_p,\ f_j(x)_q\}, \tag{8}$$

which is high only when both classes remain active. Averaging over edge-case probes yields the boundary preservation score

$$P_{pq}(f_i \to f_j) = \frac{1}{|\mathcal{E}_{pq}^{(f_i)}|} \sum_{x \in \mathcal{E}_{pq}^{(f_i)}} A_{pq}^{(f_j)}(x). \tag{9}$$

For each class c, we average the boundary preservation scores over available class pairs involving c to obtain the class-level boundary consistency $P_c(f_i \to f_j)$.

Symmetrization. We symmetrize both prototype and boundary scores as $T_c(f_i, f_j) = \frac{1}{2}\big(T_c(f_i \to f_j) + T_c(f_j \to f_i)\big)$ and $P_c(f_i, f_j) = \frac{1}{2}\big(P_c(f_i \to f_j) + P_c(f_j \to f_i)\big)$. The feature consensus between f_i and f_j is then represented as $\{\,(T_c(f_i, f_j),\ P_c(f_i, f_j)) \mid c = 1, \ldots, C\,\}$.

3.4 Application to Federated Learning: Consensus-Based Client Filtering

In federated learning, the server receives k client models $\{f_1, \ldots, f_k\}$ in each round. We extend the pairwise compatibility formulation to the multi-model setting and apply it as a consensus-based client filtering procedure.

Multi-model feature consensus. The multi-model feature consensus is represented as

$$\{(\mathbf{T}_c, \mathbf{P}_c) \mid c = 1, \ldots, C\}, \tag{10}$$

where $\mathbf{T}_c, \mathbf{P}_c \in \mathbb{R}^{k \times k}$ with (i, j)-th entries $T_c(f_i, f_j)$ and $P_c(f_i, f_j)$. At the class level, we define the class-wise agreement for each class and model pair (f_i, f_j) as

$$A_{ij}^{(c)} = r(f_i, f_j)\Big[1 - \big(1 - T_c(f_i, f_j)\big)\big(1 - P_c(f_i, f_j)\big)\Big]. \tag{11}$$

At the model level, we aggregate agreement uniformly across the label space as $\tilde{A}_{ij} = \frac{1}{C}\sum_{c=1}^{C} A_{ij}^{(c)}$, which favors agreement distributed across labels rather than concentrated on a small subset of labels.

Consensus-based objective. Given a candidate model subset $\mathcal{M} \subseteq \{1, \ldots, k\}$, we measure its internal consistency using

$$J_{\text{con}}(\mathcal{M}) = \frac{2}{|\mathcal{M}|(|\mathcal{M}| - 1)} \sum_{i<j,\, i,j \in \mathcal{M}} \tilde{A}_{ij}, \tag{12}$$

which computes the average pairwise agreement within the subset. The agreement $\tilde{A}_{ij}$ jointly captures output diversity and feature consensus, favoring subsets that are diverse in their predictions while maintaining consistent class-level representations. We select a subset of size $\lceil k/2 \rceil$ (at least half of the participating clients) by maximizing the consensus objective:

$$\mathcal{M}^\star = \arg \max_{\substack{\mathcal{M} \subseteq \{1,\ldots,k\} \\ |\mathcal{M}| = \lceil k/2 \rceil}} J_{\text{con}}(\mathcal{M}). \tag{13}$$

Cross-round probe memory. To improve probing efficiency and stability, we maintain a memory bank of high-quality probes across rounds. Specifically, we store: (i) Top-K probes per class label, and (ii) edge-case probes for class pairs, extracted from previously selected clients. These probes are merged with newly sampled probes in subsequent rounds, enabling progressive refinement of the probe pool and more stable estimation of feature consensus over time.

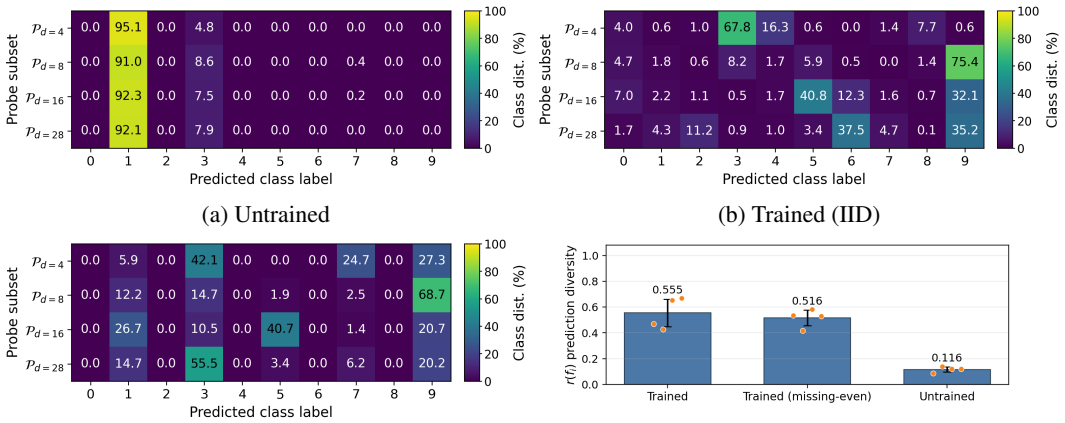
4 Experiments

4.1 Model Diversity and Feature Compatibility

We illustrate the Model Compatibility Assessment through a controlled case study with six representative models: one untrained model, two benign models, one label-skewed model missing even class labels, one backdoored model, and one different-domain model trained on a separate dataset. The untrained model retains random initialization. All other models are trained on disjoint IID data subsets. The two benign models are trained on disjoint IID subsets of MNIST [29]. The label-skewed model is trained on MNIST with labels of even numbers removed. The backdoored model is trained on MNIST, where 50% of samples in its subset are poisoned using a BadNets-style [30] 3×3 trigger placed at the bottom-right corner and targeting class 7. The different-domain model is trained on Fashion-MNIST (FMNIST) [31] with a dataset size matched to a single MNIST subset. All models use the same ResNet-20 architecture [32]. Except for the untrained model, all trained models are optimized for 20 epochs using 10% of the full training data, with SGD (learning rate 0.001, batch size 256, momentum 0.9, weight decay 5×10^{-4}). This setup yields benign models with accuracy around 95% on MNIST.

For Pointillism, we set the grid resolution $d \in \mathcal{D} = \{4, 8, 16, 28\}$. The two-group probe configuration samples group centers from $[0, 1]$ with a minimum separation constraint $d_{\min} = 0.4$, and uses per-cell perturbation $\sigma_{\text{fg}} = \sigma_{\text{bg}} = 0.05$. Probes are generated across multiple grid resolutions with activation ratios $\rho \in \Omega = \{0.08, 0.16, 0.32, 0.50\}$, where each probe activates $w(d, \rho) = \text{round}(\rho d^2)$ cells. The per-resolution budget $B = 8000$. We maintain Top-$M = 2048$ candidates per class, set $K = 512$ for Top-K selection, and retain $|\mathcal{E}_{pq}| = 512$ edge-case probes per class pair. Local refinement is enabled with up to two edit operations per probe.

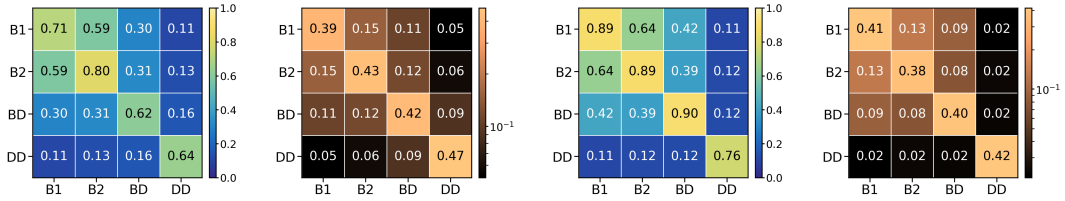
Model Diversity. Fig. 3 visualizes the predicted-label distributions ${q^{(d)}}_{d \in \mathcal{D}}$ together with the aggregated diversity scores $r(f_i)$. The untrained model exhibits a clear collapse, where the majority



(a) Untrained (b) Trained (IID)

(c) Trained (missing-even) (d) Prediction diversity $r(f_i)$

Figure 3: Visualization of the global signature $\mathcal{S}_g$ across representative models. The first three panels show class label distributions $\{q^{(d)}\}_{d\in\mathcal{D}}$ for untrained, IID-trained, and label-skewed (missing-even) models, respectively. The bottom-right panel reports the aggregated prediction diversity $r(f_i)$ across probe configurations.



(a) 5 epochs: $\bar{\mathbf{T}}$ (b) 5 epochs: $\bar{\mathbf{P}}$ (c) 20 epochs: $\bar{\mathbf{T}}$ (d) 20 epochs: $\bar{\mathbf{P}}$

Figure 4: Model-level summaries of feature consensus under transfer. Each cell shows mean symmetric agreement aggregated over class labels. Prototype consistency ($\bar{\mathbf{T}}$) and boundary consistency ($\bar{\mathbf{P}}$) at 5 and 20 training epochs. Labels: B1/B2 (benign), BD (backdoored), DD (different domain).

of probes are consistently mapped to a single label across resolutions, resulting in a low diversity score of 0.116. In contrast, the IID-trained model distributes predictions across all class labels, producing a balanced response pattern and a high diversity score of 0.555. The missing-even model shows structured sparsity: predictions on even-indexed classes are nearly zero, while mass is concentrated on the remaining labels, leading to a slightly lower diversity score of 0.516. These patterns demonstrate that $r(f_i)$ captures both complete collapse and structured label bias, assigning low scores to degenerate behaviors and higher scores to models with broader prediction support.

Feature Compatibility. We evaluate four trained models with full class label coverage: two benign models (B1, B2), one backdoored model (BD), and one cross-domain model (DD). To examine training progression, we consider snapshots at 5 and 20 epochs. At 5 epochs, benign models achieve 64% accuracy, while the backdoored model has 27% accuracy with attack success rate (ASR $= 100\%$). At 20 epochs, benign models reach 96% accuracy, while the backdoored model attains 90% accuracy with ASR remaining at 100%.

Fig. 4 summarizes the aggregated prototype and boundary consistency matrices $\bar{\mathbf{T}}$ and $\bar{\mathbf{P}}$ (averaged over $\mathbf{T}_c$ and $\mathbf{P}_c$ in Eq. 10 across class labels). The benign pair (B1, B2) exhibits strong alignment, with $\bar{\mathbf{T}}$ increasing from 0.59 to 0.64 and $\bar{\mathbf{P}}$ remaining stable around 0.15–0.13 across training. In contrast, the backdoored model shows reduced compatibility with benign models: at 20 epochs, $\bar{\mathbf{T}}$ between BD and B1/B2 remains moderate (0.42/0.39), while $\bar{\mathbf{P}}$ drops to 0.09/0.08, reflecting misalignment in decision boundaries despite improved accuracy. The different-domain model (DD) exhibits consistently low agreement with others, with $\bar{\mathbf{T}}$ values around 0.11–0.12 and $\bar{\mathbf{P}}$ near 0.02, indicating a lack of shared feature structure. These results show that feature consensus distinguishes compatible models from both backdoored and different-domain models beyond standard performance metrics.

4.2 Federated Learning Defense

4.2.1 Experimental Setup

We evaluate Pointillism in a standard federated learning setting on FMNIST [31], CIFAR-10 [33], and GTSRB [34]. The system consists of $N = 100$ clients, with $k = 10$ clients sampled per communication round. Following [35, 36], the malicious population ratio is fixed at 30%, corresponding to $N_m = 30$ compromised clients. To ensure controlled and reproducible evaluation, we fix the per-round malicious proportion to match the population-level ratio, enforcing exactly 3 malicious clients among the selected participants in each round. This avoids randomness in client sampling and prevents attacker-majority rounds that could introduce high variance across defense methods.

Client data is distributed in a non-IID manner using a Dirichlet distribution [37], with $\alpha = 0.5$ for FMNIST and GTSRB, and $\alpha = 0.9$ for CIFAR-10. Each compromised client performs an all-to-one backdoor attack [16], poisoning 50% of its local data while using the same training procedure as benign clients. The global model architecture is ResNet-20 [32]. Each selected client trains for one local epoch using SGD with batch size 64, momentum 0.9, and weight decay 5×10^{-4}. The learning rate is set to 0.002 for FMNIST, 0.005 for GTSRB and CIFAR-10. Training runs for 120 rounds on FMNIST and 200 rounds on CIFAR-10 and GTSRB. The probing configuration for Pointillism follows the same setting as described in Sec. 4.1, with grid resolutions $d \in \mathcal{D} = \{4, 8, 16, 28\}$ for FMNIST and $d \in \mathcal{D} = \{4, 8, 16, 32\}$ for CIFAR-10 and GTSRB.

Defense baselines: We compare Pointillism against six representative federated learning defenses: FedAvg [1], Norm Bound [20], RFA [8], Multi-Krum (Krum) [7], FLAME [9], and DeepSight [28].

Attack baselines: We include two untargeted attacks. Label Perturbation [14] shifts labels cyclically as $c \mapsto (c + 1) \bmod C$ (e.g., $0 \to 1, \ldots, 9 \to 0$). Reverse Gradient [10] flips the sign of the gradient after local training. Both attacks apply a scaling factor of $\frac{N}{N_m}$ to the model update norm. We further evaluate five backdoor (targeted) attacks: BadNets [30], DBA [38], Neurotoxin [39], Scaling attack (Scale) [35], and A3FL [40]. A3FL performs trigger optimization, while DBA follows its original trigger configuration. For fixed-trigger attacks, BadNets uses a 3×3 bottom-right trigger as in Sec. 4.1, while Scale follows the same trigger injection but scales the model update norm with the same factor $\frac{N}{N_m}$. All attack baselines are implemented within the same experimental framework to ensure consistent comparison.

Evaluation metrics: Following [40, 39, 38], we report test accuracy (ACC) and attack success rate (ASR). Each experiment is repeated three times, and results are averaged across runs.

4.2.2 Main Results

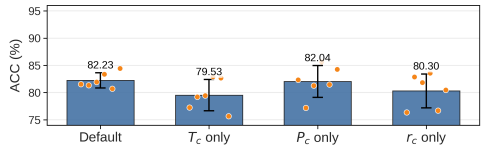
As shown in Tab. 1, Pointillism achieves the highest average ACC across all datasets under untargeted attacks. On FMNIST, it attains 81.22, compared to 76.22 from the strongest baseline (FLAME). On GTSRB, it reaches 82.68, exceeding 78.65 from FLAME. On CIFAR-10, it achieves 55.82, compared to 52.39 from FLAME, corresponding to an improvement of up to 5.0%. Under backdoor attacks, Pointillism reports the lowest average ASR on all datasets. On FMNIST, the ASR is 1.10, compared to 9.01 from FLAME. On GTSRB, it achieves 0.73, compared to 7.38 from FLAME. On CIFAR-10, the ASR is 10.70, compared to 15.63 from Krum. The reduction on FMNIST corresponds to a relative decrease of 87.8%. Across both settings, Pointillism maintains higher ACC than competing methods under untargeted attacks and lower ASR under backdoor attacks.

4.2.3 Ablation and Sensitivity Study.

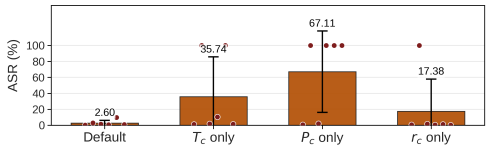
We evaluate the contribution of each component in Eq. (11) on FMNIST with 40% malicious clients under Dirichlet non-IID partitioning with $\alpha = 0.5$. We compare the default Pointillism configuration against variants that use only prototype consistency T_c, boundary consistency P_c, or model diversity $r(f_i, f_j)$ for client filtering. As shown in Fig. 5, the default design achieves higher ACC under Label Perturbation and lower ASR under BadNets. In contrast, using a single component leads to larger variance and degraded robustness, especially for ASR, indicating that combining T_c, P_c, and $r(f_i, f_j)$ provides more stable client selection. Sensitivity analysis across Dirichlet α compares Pointillism with the strongest baseline FLAME. Pointillism consistently achieves higher ACC and lower ASR, particularly in highly non-IID settings (low α) with a high malicious ratio. Additional analyses and discussion, including computational costs, are provided in Appendix A–E.

Table 1: Defense performance under untargeted and backdoor attacks. We report ACC (%) for untargeted attacks and ASR/ACC (%) for backdoor attacks. Best results are bolded.

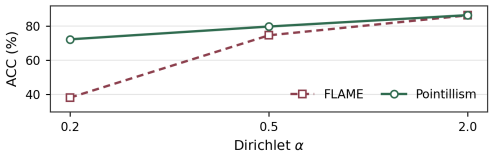
Dataset	Attack	FedAvg	Norm Bound	RFA	Krum	FLAME	DeepSight	Pointillism (ours)
FMNIST	LabelPerturb	27.90	47.77	32.91	71.21	79.94	77.59	**80.73**
	ReverseGrad	10.00	67.60	73.23	76.63	72.49	64.27	**81.70**
	Avg ACC ↑	18.95	57.69	53.07	73.92	76.22	70.93	**81.22**
	A3FL	100.0/81.17	100.0/82.85	100.0/82.82	21.07/79.49	4.63/80.04	**0.19**/82.34	1.70/81.70
	BadNets	100.0/84.26	100.0/80.02	100.0/82.12	41.00/81.42	2.15/82.45	100.0/82.37	**1.75**/83.66
	DBA	12.74/81.78	5.48/79.96	6.85/78.13	65.89/73.13	30.67/76.59	5.52/83.56	**0.12**/79.86
	Neurotoxin	99.04/68.91	100.0/71.95	100.0/68.78	100.0/80.91	7.26/76.18	100.0/84.15	**1.80**/80.93
	Scale	100.0/85.11	100.0/82.42	100.0/81.92	6.48/76.03	0.33/80.76	100.0/81.64	**0.13**/81.21
	Avg ASR ↓	82.36	81.10	81.37	46.89	9.01	61.14	**1.10**
GTSRB	LabelPerturb	25.39	67.75	74.24	83.93	80.63	78.80	**85.64**
	ReverseGrad	5.23	53.67	59.25	71.16	76.67	47.89	**79.71**
	Avg ACC ↑	15.31	60.71	66.75	77.55	78.65	63.35	**82.68**
	A3FL	100.0/85.87	100.0/88.27	100.0/87.29	16.22/83.52	2.65/80.17	**0.27**/77.07	0.45/81.98
	BadNets	95.10/82.13	94.86/81.58	95.49/79.57	84.33/72.05	5.19/82.77	93.45/80.97	**0.55**/81.01
	DBA	4.05/74.62	4.71/85.65	1.93/85.75	11.43/75.56	8.42/81.20	2.41/82.23	**0.23**/83.30
	Neurotoxin	94.70/81.03	95.51/83.51	95.42/83.79	80.06/84.16	24.40/83.97	22.04/81.81	**1.22**/73.50
	Scale	91.33/85.16	95.90/85.34	95.41/83.81	3.38/83.10	3.59/83.35	26.89/79.31	**1.95**/81.71
	Avg ASR ↓	80.48	73.33	69.21	32.79	7.38	29.01	**0.73**
CIFAR-10	LabelPerturb	15.15	41.60	39.78	46.16	**58.56**	49.29	55.89
	ReverseGrad	10.00	35.66	34.39	51.21	46.21	40.11	**55.75**
	Avg ACC ↑	12.58	38.63	37.09	48.69	52.39	44.70	**55.82**
	A3FL	100.0/51.42	100.0/57.28	100.0/51.99	**2.32**/57.80	2.53/57.11	100.0/49.33	27.25/52.63
	BadNets	85.26/41.81	59.30/25.29	61.69/36.48	3.74/35.79	1.24/43.90	96.87/58.08	**1.14**/37.00
	DBA	46.50/57.48	34.89/60.63	39.20/59.90	69.80/45.21	78.74/38.57	**0.50**/52.55	10.64/55.67
	Neurotoxin	93.41/56.70	94.13/59.37	93.06/61.03	**1.31**/55.73	4.65/46.36	97.22/59.14	4.37/54.66
	Scale	96.97/47.99	96.57/56.81	95.04/51.11	**1.00**/50.01	2.06/57.88	97.20/60.49	10.07/54.39
	Avg ASR ↓	84.43	76.98	77.80	15.63	17.84	78.36	**10.70**



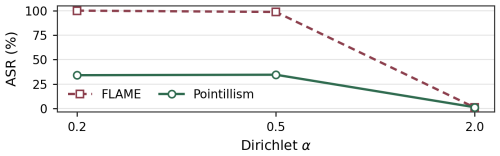
(a) Ablation: ACC (Label Perturbation)



(b) Ablation: ASR (BadNets)



(c) Sensitivity: ACC vs Dirichlet α



(d) Sensitivity: ASR vs Dirichlet α

Figure 5: Ablation and sensitivity study of Pointillism on FMNIST with 40% malicious clients.

5 Conclusion

We propose *Pointillism* to address model compatibility assessment in collaborative machine learning. Our framework evaluates models directly in function space using structured probes generated from an out-of-distribution space. From probe responses, the framework constructs compact model signatures based on global statistics and class-level features. Compatibility is assessed through model diversity and feature consensus derived from cross-model probe transfer. Experiments in federated learning show that our approach enables reliable model selection and improves robustness under adversarial attacks.

Acknowledgments and Disclosure of Funding

This research is supported by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority.

The authors declare no competing interests.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1273–1282. PMLR, 2017.

[2] Anousheh Gholami, Nariman Torkzaban, and John S Baras. Trusted decentralized federated learning. In *2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC)*, pages 1–6. IEEE, 2022.

[3] Shivam Kalra, Junfeng Wen, Jesse C Cresswell, Maksims Volkovs, and Hamid R Tizhoosh. Decentralized federated learning through proxy model sharing. *Nature communications*, 14(1):2899, 2023.

[4] Minghong Fang, Zifan Zhang, Hairi, Prashant Khanduri, Jia Liu, Songtao Lu, Yuchen Liu, and Neil Gong. Byzantine-robust decentralized federated learning. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2874–2888, 2024.

[5] Christodoulos Pappas, Dimitris Chatzopoulos, Spyros Lalis, and Manolis Vavalis. Ipls: A framework for decentralized federated learning. In *2021 IFIP Networking Conference (IFIP Networking)*, pages 1–6. IEEE, 2021.

[6] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *International conference on machine learning*, pages 5650–5659. Pmlr, 2018.

[7] Peva Blanchard, El Mahdi El Mhamdi, Rachid Guerraoui, and Julien Stainer. Machine learning with adversaries: Byzantine tolerant gradient descent. *Advances in neural information processing systems*, 30, 2017.

[8] Cong Xie, Oluwasanmi Koyejo, and Indranil Gupta. Fall of empires: Breaking byzantine-tolerant sgd by inner product manipulation. In *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115, pages 261–270. PMLR, 22–25 Jul 2020.

[9] Thien Duc Nguyen, Phillip Rieger, Huili Chen, Hossein Yalame, Helen Möllering, Hossein Fereidooni, Samuel Marchal, Markus Miettinen, Azalia Mirhoseini, Shaza Zeitouni, Farinaz Koushanfar, Ahmad-Reza Sadeghi, and Thomas Schneider. FLAME: Taming backdoors in federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1415–1432, Boston, MA, August 2022. USENIX Association.

[10] Clement Fung, Chris JM Yoon, and Ivan Beschastnikh. The limitations of federated learning in sybil settings. In *23rd International symposium on research in attacks, intrusions and defenses (RAID 2020)*, pages 301–316, 2020.

[11] Zaixi Zhang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Fldetector: Defending federated learning against model poisoning attacks via detecting malicious clients. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 2545–2555, 2022.

[12] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 634–643. PMLR, 09–15 Jun 2019.

[13] Virat Shejwalkar and Amir Houmansadr. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*, 2021.

[14] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning. In *Proceedings of the 29th USENIX Conference on Security Symposium*, SEC'20, USA, 2020. USENIX Association.

[15] Xiaoting Lyu, Yufei Han, Wei Wang, Jingkai Liu, Bin Wang, Jiqiang Liu, and Xiangliang Zhang. Poisoning with cerberus: Stealthy and colluded backdoor attack against federated learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(7):9020–9028, Jun. 2023.

[16] Haomin Zhuang, Mingxian Yu, Hao Wang, Yang Hua, Jian Li, and Xu Yuan. Backdoor federated learning by poisoning backdoor-critical layers, 2024.

[17] El-Mahdi El-Mhamdi, Rachid Guerraoui, and Sébastien Rouault. The hidden vulnerability of distributed learning in Byzantium. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3521–3530. PMLR, 10–15 Jul 2018.

[18] Gilad Baruch, Moran Baruch, and Yoav Goldberg. A little is enough: Circumventing defenses for distributed learning. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[19] Paul Signac. L'Hirondelle Steamer on the Seine. Wikimedia Commons, 1901. Public domain. Accessed 2026-04-27.

[20] Ziteng Sun, Peter Kairouz, Ananda Theertha Suresh, and H Brendan McMahan. Can you really backdoor federated learning? *arXiv preprint arXiv:1911.07963*, 2019.

[21] Dong Yin, Yudong Chen, Ramchandran Kannan, and Peter Bartlett. Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5650–5659, 10–15 Jul 2018.

[22] Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, and Bo Du. Parameter disparities dissection for backdoor defense in heterogeneous federated learning. *Advances in Neural Information Processing Systems*, 37:120951–120973, 2024.

[23] Jiahao Xu, Zikai Zhang, and Rui Hu. Detecting backdoor attacks in federated learning via direction alignment inspection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20654–20664, 2025.

[24] Wei Wan, Shengshan Hu, Minghui Li, Jianrong Lu, Longling Zhang, Leo Yu Zhang, and Hai Jin. A four-pronged defense against byzantine attacks in federated learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 7394–7402, New York, NY, USA, 2023. Association for Computing Machinery.

[25] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8715–8724, 2020.

[26] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261, 2020.

[27] Zhuowen Yuan, Fan Wu, Yunhui Long, Chaowei Xiao, and Bo Li. Secretgen: Privacy recovery on pre-trained models via distribution discrimination. In *European Conference on Computer Vision*, pages 139–155. Springer, 2022.

[28] Phillip Rieger, Thien Duc Nguyen, Markus Miettinen, and Ahmad-Reza Sadeghi. Deepsight: Mitigating backdoor attacks in federated learning through deep model inspection. *arXiv preprint arXiv:2201.00763*, 2022.

[29] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[30] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *Ieee Access*, 7:47230–47244, 2019.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[33] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[34] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288, 2013.

[35] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR, 26–28 Aug 2020.

[36] Ye Li, Yanchao Zhao, Chengcheng Zhu, and Jiale Zhang. Infighting in the dark: Multi-label backdoor attack in federated learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 25770–25779, 2025.

[37] Thomas Minka. Estimating a dirichlet distribution, 2000.

[38] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. Dba: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations (ICLR)*, 2019.

[39] Zhengming Zhang, Ashwinee Panda, Linyue Song, Yaoqing Yang, Michael Mahoney, Prateek Mittal, Ramchandran Kannan, and Joseph Gonzalez. Neurotoxin: Durable backdoors in federated learning. In *International conference on machine learning*, pages 26429–26446. PMLR, 2022.

[40] Hangfan Zhang, Jinyuan Jia, Jinghui Chen, Lu Lin, and Dinghao Wu. A3fl: Adversarially adaptive backdoor attacks to federated learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 61213–61233. Curran Associates, Inc., 2023.

Appendix

A Why OOD Probes Reveal Model Compatibility

We add a controlled experiment to test whether Pointillism probe disagreement is associated with a poisoning-induced change in representations of real task images. We construct a binary MNIST task using digits 0 and 3 and train four models with the same architecture and initialization within each repetition:

- B_1 and B_2: balanced, disjoint clean subsets;
- B_{imb}: a clean subset with the same effective $800/1200$ class proportion as the malicious model;
- M: BadNets training, where 20% of class-0 samples receive a 3×3 trigger and are relabeled as class 3.

The two clean comparisons control for representational differences caused by disjoint training data and class imbalance. All analyses use a separate MNIST test set and the same fixed Pointillism probe pool across five random seeds.

A.1 Real-image representation difference

Let $\mathcal{D}_c$ be the set of $n_c = |\mathcal{D}_c|$ images with class label c in the MNIST test set, and let i and j index two models. We feed the same images in $\mathcal{D}_c$ to both models. If the penultimate layer has p feature dimensions, then $H_{i,c} \in \mathbb{R}^{n_c \times p}$ denotes the feature matrix produced by model i, with one row per image. We use linear centered kernel alignment (CKA) because it compares the pairwise geometry of two representations while being invariant to an orthogonal change of feature basis and to isotropic scaling.

We define their classwise representation distance as

$$d_c(i, j) = 1 - \text{CKA}(H_{i,c}, H_{j,c}).$$

Here, $d_c(i, j)$ measures how differently models i and j represent the same real images from class c: a value near 0 indicates highly similar representation geometry, while a larger value indicates greater representational mismatch. The real-image CKA distances are reported in Table A1.

Table A1: Real-image representation differences between clean and malicious models.

Model comparison	Real-image CKA distance
B_1–B_2 (clean-clean)	0.151
B_1–B_{imb} (clean-imbalance)	0.141
B_1–M (clean-malicious)	**0.216**

Table A1 reports each pair's CKA distance averaged over classes 0 and 3 and then over five seeds. The clean-malicious distance is larger than both control distances, and this ordering holds in every seed. This provides evidence that the mismatch involving the malicious model extends to its representations of real task images, rather than appearing only in its responses to synthetic probes.

A.2 Trigger effect on real-image features

Let $h_i(x)$ denote the penultimate-layer feature vector produced by model i for input x. For each class $c \in \{0, 3\}$, we define model i's real-image class prototype as the mean feature vector over the images of class c in the MNIST test set:

$$\mu_{i,c} = \frac{1}{n_c} \sum_{x \in \mathcal{D}_c} h_i(x).$$

We then define the target-versus-source affinity of x as

$$a_i(x) = \cos(h_i(x), \mu_{i,3}) - \cos(h_i(x), \mu_{i,0}).$$

Table A2: Trigger-induced target-affinity shifts on real MNIST images.

Model	Trigger-induced target shift
B_1	0.012 ± 0.013
B_2	0.004 ± 0.012
B_{imb}	0.002 ± 0.002
M	$\mathbf{1.291 \pm 0.248}$

Table A3: Target-versus-source affinity of malicious-only Pointillism probes.

Model	Target-versus-source affinity
B_1	-0.247 ± 0.096
B_2	-0.572 ± 0.224
B_{imb}	-0.366 ± 0.106
M	$\mathbf{+0.487 \pm 0.120}$

We use cosine similarity because it measures alignment with each class prototype while being insensitive to the magnitude of the feature vector. The difference between the two similarities therefore indicates whether an input is relatively more target-affine or source-affine.

Let T denote the operation that adds the backdoor trigger to an image. We measure model i's average trigger-induced target-affinity shift over class-0 images in the MNIST test set as

$$A_i = \frac{1}{n_0} \sum_{x \in \mathcal{D}_0} [a_i(T(x)) - a_i(x)].$$

The trigger-induced target-affinity shifts are shown in Table A2. The actual trigger produces a large target-oriented shift only in M, while all clean models remain near zero. This is consistent with backdoor training inducing a representation change that moves triggered class-0 images toward the prototype of real class-3 images.

A.3 The same task-related change is exposed by Pointillism probes

Let p denote a Pointillism probe and $\hat{y}_i(p)$ the class predicted for it by model i. We select the malicious-only target-probe set

$$\mathcal{P}_M = \{p : \hat{y}_M(p) = 3,\ \hat{y}_{B_1}(p) = 0,\ \hat{y}_{B_2}(p) = 0\}.$$

Every model evaluates the same selected probes by comparing their features with the mean feature representations of class-0 and class-3 images from the MNIST test set. For model i, the target-versus-source affinity of probe p is

$$S_i(p) = \cos(h_i(p), \mu_{i,3}) - \cos(h_i(p), \mu_{i,0}).$$

A positive $S_i(p)$ indicates greater affinity to the real class-3 prototype, whereas a negative value indicates greater affinity to the real class-0 prototype. Table A3 reports $S_i(p)$ averaged over the selected probes.

The same selected probes are source-affine on average in all three clean models but target-affine on average in M. Thus, the observed disagreement among the models is not merely an arbitrary difference in the labels they assign to OOD inputs. In M, the probe features are more aligned with the prototype constructed from real target-class images; in the clean models, they are more aligned with the real source-class prototype.

Together, these results provide direct evidence linking poisoned training to changes in representations of real task images and to disagreement on Pointillism probes. In this experiment, the malicious model exhibits both a trigger-induced shift of real class-0 images toward the class-3 prototype and target-affine representations of the selected probes. The probes therefore do not need to resemble samples from the task distribution. By evaluating the same probes across models, Pointillism exposes differences in their learned class-specific representations.

Table B1: Pointillism under a white-box adaptive attack with 20% malicious clients.

Dataset	Model	Malicious clients (%)	Dirichlet α	Final ACC	Final ASR
CIFAR-10	ResNet-20	20	0.9	$57.50 \pm 2.53\%$	$86.35 \pm 19.56\%$
CIFAR-10	MobileNetV2	20	0.9	$55.88 \pm 1.75\%$	$25.78 \pm 29.61\%$
FMNIST	ResNet-20	20	0.5	$80.94 \pm 3.87\%$	$4.33 \pm 4.02\%$

B Additional Robustness Analysis

B.1 White-box adaptive attack

We evaluate Pointillism with a white-box adaptive attack. The adversary knows the full defense, including probe generation, signatures, consensus, and subset selection. At round t, each selected malicious client $i \in \mathcal{N}_m^t$ trains two models from the same global initialization: a clean reference $\theta_{i,\text{clean}}^t$ and a malicious endpoint $\theta_{i,\text{mal}}^t$ using a BadNets poisoning ratio of 0.5. It constructs candidates by parameter interpolation:

$$\theta_{i,\lambda}^t = (1-\lambda)\theta_{i,\text{clean}}^t + \lambda\theta_{i,\text{mal}}^t, \quad \lambda \in \Lambda = \{0.1, 0.25, 0.5, 1.0\}.$$

Using an independently generated private probe pool, the attacker computes Pointillism's pairwise-consensus matrix A^{clean} over the clean reference models of the selected malicious clients. It defines client i's clean consensus score as its mean pairwise Pointillism consensus with the other selected malicious clients:

$$C_i^{\text{clean}} = \frac{1}{|\mathcal{N}_m^t| - 1} \sum_{\substack{j \in \mathcal{N}_m^t \\ j \neq i}} A_{ij}^{\text{clean}}, \qquad \tau_C = \text{Median}_{i \in \mathcal{N}_m^t} C_i^{\text{clean}}.$$

Thus, τ_C is the median clean mean-pairwise-consensus score and serves as the benign-looking threshold. To score a candidate $\theta_{i,\lambda}^t$, the attacker replaces only client i's clean model with that candidate, retains the other malicious clients' clean references, and recomputes the pairwise-consensus matrix from the cached private-probe outputs. Denoting this matrix by $A^{(i,\lambda)}$, the candidate score is

$$C_i^\lambda = \frac{1}{|\mathcal{N}_m^t| - 1} \sum_{\substack{j \in \mathcal{N}_m^t \\ j \neq i}} A_{ij}^{(i,\lambda)}.$$

The attacker evaluates each interpolated model using its private probe pool and selects the candidate with the highest ASR that remains sufficiently consistent with the clean reference models:

$$\lambda_i^\star = \arg\max_{\lambda \in \Lambda} \text{ASR}_i(\theta_{i,\lambda}^t) \quad \text{s.t.} \quad C_i^\lambda \geq \tau_C.$$

If no candidate satisfies the constraint, the client submits its clean reference. The server's probes remain private. The adaptive-attack results are shown in Table B1. The adaptive attack follows the main experimental setting of 100 total clients, with 10 clients sampled per round. Under the 20% malicious-client setting, 2 of the 10 clients sampled in each round are malicious.

With 20% malicious clients, Pointillism can be compromised by the white-box adaptive attack on CIFAR-10 with ResNet-20, where the attack achieves an ASR of 86.35%. In contrast, Pointillism remains robust on FMNIST, limiting the ASR to 4.33%. On CIFAR-10, using the larger MobileNetV2 substantially improves robustness, reducing the ASR to 25.78%. This stronger robustness with a larger model is consistent with the capacity effect discussed in Sec. C.

The adaptive attack also incurs substantial computation: constructing the adaptive updates for the malicious-client group takes 34.45 seconds, whereas one benign client's local training takes only 0.29 seconds. Under practical federated learning round deadlines, these malicious clients would therefore be likely to time out or be treated as stragglers before their updates could be accepted. Thus, although the white-box results characterize performance under an unrestricted attacker model, executing such an attack may be challenging in a time-constrained deployment.

Table B2: Defense performance when 50% of participating clients are malicious.

Dataset	Metric	FedAvg	Norm Bound	RFA	Krum	FLAME	DeepSight	Pointillism
CIFAR-10	ACC (%) ↑	59.10	56.71	56.57	54.75	57.84	59.79	54.01
	ASR (%) ↓	97.66	97.41	96.43	97.72	96.93	97.43	96.74
FMNIST	ACC (%) ↑	77.76	82.80	81.20	81.06	82.76	80.49	81.78
	ASR (%) ↓	100.00	100.00	100.00	100.00	100.00	100.00	100.00

Table B3: False-negative and false-positive rates of client selection.

Dataset	Attack	Pointillism FNR/FPR	Krum FNR/FPR	FLAME FNR/FPR
FMNIST	BadNets	**11.76**/19.33	21.11/37.62	23.61/**17.74**
	DBA	**18.15/22.06**	69.17/58.21	81.11/38.45
	ReverseGrad	**5.28/16.55**	12.78/34.05	21.94/19.17
CIFAR-10	BadNets	**7.50**/17.50	**7.50**/31.79	8.83/**12.79**
	DBA	**29.72/27.02**	75.50/60.93	85.50/42.21
	ReverseGrad	9.89/18.52	**0.00**/28.57	0.33/**11.86**

B.2 Malicious-majority setting

Task-data-free federated learning defenses, including Pointillism, assume a benign majority and cannot reliably defend when malicious clients comprise at least half of the participants, as shown in Table B2.

B.3 Benign-client selection errors and fairness

We further evaluate the client-selection behavior of Pointillism using the false-negative rate (FNR) and false-positive rate (FPR). A false negative is a malicious client retained for aggregation, whereas a false positive is a benign client incorrectly excluded. Lower values are preferable for both metrics. These evaluations use the same settings as the main experiments: 30% of the client population is malicious, and the Dirichlet non-IID parameter is $\alpha = 0.5$ for FMNIST and $\alpha = 0.9$ for CIFAR-10. The FNR and FPR results are shown in Table B3.

Pointillism obtains the lowest FNR and FPR under DBA on both datasets and under ReverseGrad on FMNIST. Across all six settings, Pointillism has a lower FPR than Krum and a lower or equal FNR in five settings. Overall, Pointillism achieves a better balance between retaining benign clients and excluding malicious clients than Krum and FLAME.

To further assess fairness among benign clients, we report the normalized entropy of the false-positive distribution, which measures whether exclusions are concentrated on particular benign clients. The metric is averaged over FMNIST and CIFAR-10, and higher values indicate that false-positive exclusions are distributed more evenly among benign clients. The results are shown in Table B4.

Pointillism maintains consistently high normalized entropy across all three attacks, indicating that its false positives are not disproportionately concentrated on a small subset of benign clients. Together with its low FPR, this demonstrates a favorable balance between limiting benign-client exclusion and distributing unavoidable exclusions more evenly.

C Generalization and Probe Sensitivity

C.1 Generalization across architectures

We evaluated Pointillism with SmallCNN, ResNet-20, and MobileNetV2 on CIFAR-10 under BadNets. They contain 61,338, 272,474, and 2,236,682 parameters, respectively. We report clean test accuracy (ACC) and attack success rate (ASR) in Table C1.

Performance varies substantially by architecture. Pointillism performs poorly with SmallCNN, ResNet-20 provides much stronger mitigation, and MobileNetV2 achieves the lowest ASR with comparable ACC.

Table B4: Normalized entropy of false-positive exclusions among benign clients.

Defense	BadNets	DBA	ReverseGrad	Overall mean
Pointillism	86.86	88.53	84.41	86.60
Krum	89.88	96.01	87.40	91.10
FLAME	84.91	93.86	77.98	85.58

Table C1: Pointillism performance across model architectures on CIFAR-10 under BadNets.

Metric	SmallCNN	ResNet-20	MobileNetV2
ACC (%)	54.32 ± 5.56	57.47 ± 5.94	53.96 ± 4.06
ASR (%)	68.03 ± 44.03	14.36 ± 16.46	$\mathbf{5.67 \pm 3.41}$

Let C be the number of classes and let q_c be the fraction of probes that a model predicts as class c, so that $q = (q_1, \ldots, q_C)$ is its probe-label distribution. We compute its entropy as

$$H(q) = -\sum_{c=1}^{C} q_c \log q_c.$$

We average this entropy separately over benign and malicious clients, denoting the two averages by H_B and H_M, respectively, and track the gap $H_B - H_M$ over training. A larger positive gap indicates more diverse benign predictions and stronger malicious label concentration. Kendall's τ measures the monotonic association between the gap and the communication round, while the Theil–Sen slope estimates the gap's change per 100 rounds. These metrics are reported in Table C2.

SmallCNN has the lowest mean gap and exhibits almost no overall trend. In contrast, ResNet-20 shows a positive and more consistent trend, while MobileNetV2 achieves the largest mean gap, Kendall's τ, and growth rate, consistent with its lowest ASR. Together, these results indicate that insufficient model capacity can weaken the probe-based compatibility signal, while sufficiently expressive architectures allow the benign-malicious separation to become clearer over training.

C.2 Fine-grained classification

We also evaluate Pointillism on a fine-grained five-class task derived from the CIFAR-100 *large carnivores* superclass (bear, leopard, lion, tiger, and wolf; 2,500 training and 500 test images). We train ResNet-20 for 100 rounds with 20 clients, one local epoch, and a Dirichlet partition ($\alpha = 0.9$). Each round samples 10 clients, 20% of whom are malicious. BadNets uses a 3×3 bottom-right trigger, 50% local poisoning, and bear as the target. The results are shown in Table C3. Pointillism also remains effective on this fine-grained task, achieving the highest average ACC and the lowest average ASR.

C.3 Generalization to audio classification

Pointillism is most suitable for continuous structured domains in which semantic information is preserved under coarse-grained representations. For example, objects such as cars or apples remain recognizable in low-resolution images, while speech or music can remain identifiable after substantial reductions in audio resolution or fidelity. This property enables low-dimensional probes to capture meaningful model behavior without representing the full complexity of the original input space.

Standard NLP classification does not directly satisfy this condition because words and subwords are discrete symbolic tokens: coarsening a token sequence may disrupt or remove its semantic content rather than preserve it. Audio, by contrast, can be represented as a continuous two-dimensional time-frequency structure in which coarse acoustic patterns remain meaningful. Pointillism can therefore extend from spatial image grids to audio spectrograms by constructing probes along the frequency and time dimensions.

Table C2: Probe-label entropy separation across model architectures.

Metric	SmallCNN	ResNet-20	MobileNetV2
Mean gap	0.36 ± 0.13	0.49 ± 0.02	$\mathbf{0.76 \pm 0.07}$
Kendall's τ	-0.01 ± 0.18	0.41 ± 0.14	$\mathbf{0.53 \pm 0.09}$
Slope / 100 rounds	-0.00 ± 0.10	0.24 ± 0.08	$\mathbf{0.42 \pm 0.05}$

Table C3: Defense performance on the fine-grained CIFAR-100 large-carnivores task.

Defense	Runs	Average ACC (%)	Average ASR (%)
FedAvg	3	36.53 ± 8.86	37.00 ± 34.44
FLAME	3	41.47 ± 11.49	70.92 ± 28.14
Pointillism	3	$\mathbf{42.33 \pm 11.50}$	$\mathbf{16.42 \pm 5.01}$

We evaluate this extension on Google Speech Commands v0.02 using ten commands. Each one-second waveform is sampled at 16 kHz and converted into a normalized log-mel representation of size 40×101.

The federated system contains 100 clients, with 10 clients participating in each round, as in the main experiment. Client data are partitioned using a balanced Dirichlet distribution with $\alpha = 0.9$. We designate 20% of the clients as malicious and use fixed-rate sampling, such that exactly two malicious clients participate in each round. Training lasts for 100 communication rounds, with one local epoch per round, a batch size of 64, and Adam with a learning rate of 10^{-3}.

We use an all-to-one Audio BadNets attack with up as the target class. On each malicious client, 20% of the eligible non-target training examples are poisoned. After log-mel extraction and normalization, the last four frequency bins and last four time frames are overwritten with a value of 3.0, producing a 4×4 high-frequency, late-time trigger. The poisoned examples are relabeled as the target class. During evaluation, the same trigger is applied to non-target test examples, and ASR is the percentage of triggered examples classified as up.

Pointillism constructs probes on an 8×25 frequency-time grid. Active and inactive cells are assigned distinct values sampled from $\{-1.5, 0, 1.5, 3.0\}$, and the coarse probe is resized to 40×101 using nearest-neighbor interpolation. Probe refinement, compatibility measurement, client selection, and robust aggregation remain unchanged. The results are shown in Table C4.

Pointillism achieves the best ACC and ASR, with 83.26% ACC and 10.15% ASR. Krum also substantially reduces ASR, but its ACC decreases to 44.38%. These results demonstrate that Pointillism extends effectively to time-frequency inputs, suppressing the audio backdoor while preserving clean classification performance.

C.4 Probe-grid size

The effect of probe-grid size depends on the task. The grid-size sensitivity results are shown in Table C5. On CIFAR-10, the coarse 4×4 grid is most effective. Under a fixed probe budget, finer grids create a much larger search space, making it harder for a limited number of probes to cover meaningful structured variations. Fashion-MNIST is comparatively insensitive: all tested grids retain ASR below 1% with similar ACC. Because the server has no access to task data for selecting a task-specific resolution, our main experiments use grid sizes $d \in \mathcal{D} = \{4, 8, 16, 32\}$, including coarse grids that provide stronger coverage on difficult datasets.

D Cost Analysis in Federated Learning

Let k be the number of selected client models, C the number of classes, and $|\mathcal{P}|$ the total number of probes. Let C_f denote the cost of one forward pass. Evaluating the probe pool on all client models has computational cost $O(k\,|\mathcal{P}|\,C_f)$. Constructing the model signature involves computing predicted-label distributions and selecting Top-K and edge-case probes, which can be implemented with partial selection in $O(k\,C\,|\mathcal{P}|)$ time. In cases where $|\mathcal{T}_c| < K$, additional probes are generated

Table C4: Defense performance on Google Speech Commands under Audio BadNets.

Metric	FedAvg	Norm Bound	RFA	Krum	FLAME	DeepSight	Pointillism
ACC (%) ↑	78.69	80.66	77.39	44.38	81.64	78.18	**83.26**
ASR (%) ↓	84.89	84.67	82.67	12.44	36.00	62.67	**10.15**

Table C5: Sensitivity to probe-grid size on CIFAR-10 and FMNIST.

Dataset	Input	Probe grid	ACC (%) ↑	ASR (%) ↓
CIFAR-10	32×32	4×4	55.96 ± 3.38	$\mathbf{3.16 \pm 2.65}$
	32×32	8×8	55.35 ± 3.19	42.18 ± 44.78
	32×32	16×16	57.24 ± 4.46	65.47 ± 44.48
	32×32	32×32	$\mathbf{59.17 \pm 1.94}$	96.39 ± 2.41
FMNIST	28×28	3×3	79.89 ± 4.93	0.68 ± 0.37
	28×28	7×7	79.49 ± 4.71	0.60 ± 0.17
	28×28	14×14	$\mathbf{82.30 \pm 2.58}$	0.75 ± 0.27
	28×28	28×28	81.15 ± 4.90	$\mathbf{0.58 \pm 0.24}$

via local refinement with up to T_{edit} edits per candidate. This introduces a bounded overhead to probe generation and may increase $|\mathcal{P}|$, but is triggered selectively and does not change the overall complexity order.

After signature construction, Pointillism computes pairwise compatibility across the k models. For each model pair, prototype transfer evaluates CK probes (Top-K per class), while boundary transfer evaluates $C(C-1)K_e$ edge-case probes across class pairs. This results in $O\big(k^2(CK+C(C-1)K_e)\big)$ forward passes, yielding a total cost of $O\big(k^2(CK+C(C-1)K_e)\, C_f\big)$. Computing the corresponding agreement matrices $\{\mathbf{T}_c, \mathbf{P}_c\}_{c=1}^{C}$ and aggregating them into pairwise scores adds $O(k^2C)$ cost.

The dominant cost arises from forward passes during probe evaluation and cross-model probe transfer, giving an overall complexity of

$$\underbrace{O(kC_f|\mathcal{P}|)}_{\text{probe evaluation}} + \underbrace{O\big(k^2(CK + C(C-1)K_e)\, C_f\big)}_{\text{cross-model probe transfer}}$$

All additional steps, including probe generation, signature construction, and consensus computation, are lightweight in comparison. Since Pointillism operates entirely on the server side, it introduces no additional client-side computation or communication overhead beyond standard federated learning.

In our experiments, we set the per-resolution probe budget to $B = 8000$ with $|\mathcal{D}| = 4$ grid resolutions (e.g., $d \in \{4, 8, 16, 28\}$ or $\{4, 8, 16, 32\}$), resulting in a total probe pool size $|\mathcal{P}| \geq 32000$. For cross-model probe transfer, we use $K = 512$ prototype probes per class and $K_e = 512$ edge-case probes per class pair, yielding up to $KC + K_eC(C-1)$ cross-evaluations per model pair (not all terms are populated in practice). These forward evaluations dominate the runtime in our implementation.

D.1 Wall-clock time

We measured CUDA-synchronized wall-clock time per communication round on CIFAR-10 under BadNets using a single NVIDIA RTX A5000 GPU (24 GB). The measurement includes local training, server-side defense computation, and evaluation. The results are shown in Table D1.

Although search-based auditing incurs additional computation compared with aggregation-only defenses, Pointillism requires 17.26 seconds per round, slightly less than the 17.67 seconds required by DeepSight, another probing-based defense. This shows that Pointillism achieves competitive runtime while providing the substantial improvements in backdoor mitigation demonstrated in our main evaluation. The dominant components of Pointillism's per-round defense cost are shown in Table D2.

Table D1: Wall-clock time per communication round on CIFAR-10 under BadNets.

Method	Time / round (s)	Relative to FedAvg
FedAvg	5.54	$1.00\times$
Norm Bound	5.83	$1.05\times$
RFA	6.16	$1.11\times$
Krum	6.00	$1.08\times$
FLAME	6.03	$1.09\times$
DeepSight	17.67	$3.19\times$
Pointillism	17.26	$3.12\times$

Table D2: Breakdown of Pointillism's per-round defense cost.

Component	Time / round (s)	Share of defense
Probe-pool construction	1.70	15.6%
Base probe rendering and scoring	6.39	58.6%
Client-signature construction	1.18	10.8%
Other Pointillism steps	1.63	15.0%
Total Pointillism defense	**10.90**	**100.0%**

D.2 Scaling with clients and classes

We also measured scaling with client and class counts, as shown in Table D3. The number of classes has only a modest effect on defense time: using 5 rather than 10 CIFAR-10 classes changes it from 10.90 to 10.49 seconds, while GTSRB with 43 classes requires 14.41 seconds. Base probe rendering and scoring remain the dominant component. The defense time is 17.87 seconds with 20 participating clients, reflecting the additional probe evaluations and pairwise cross-model transfers. As future work, a two-stage strategy could improve scalability to larger client populations. Clients would first be grouped using a small probe set, after which each group could undergo full compatibility analysis in parallel across multiple GPUs.

Table D3: Runtime scaling with the numbers of participating clients and classes.

Case	Classes	Clients	Defense (s)	Relative	Total (s)	Relative
CIFAR-10 baseline	10	10	10.90 ± 0.85	$1.00\times$	17.26 ± 0.85	$1.00\times$
CIFAR-10 5-class subset	5	10	10.49 ± 0.45	$0.96\times$	15.41 ± 0.71	$0.89\times$
20 clients / round	10	20	17.87 ± 1.68	$1.64\times$	27.44 ± 1.69	$1.59\times$
GTSRB	43	10	14.41 ± 2.32	$1.32\times$	24.00 ± 2.43	$1.39\times$

E Discussion and Future Work

Pointillism introduces additional server-side computation from probe evaluation and cross-model compatibility analysis. In federated learning, where model updates per round are incremental and compatibility evaluation is performed prior to each aggregation step, this overhead accumulates across training rounds and may become noticeable in long-running processes. The grid-based two-group probe design is best suited to continuous structured domains where coarse-grained representations preserve semantic information.

As for future work, several extensions of the proposed framework are worth exploring. First, the probing-based evaluation can be naturally applied to decentralized or peer-to-peer federated learning, where clients directly assess neighboring models. Second, the framework may be particularly suitable for one-shot federated learning, where the relative cost of evaluation is reduced and compatibility assessment is performed on converged models.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims are stated in the abstract and introduction, and are supported by the experimental results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Key limitations are discussed in Appendices B, D, and E.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: The paper does not include formal theoretical results requiring proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides detailed descriptions of datasets, model architectures, training procedures, and evaluation protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The anonymized source code and instructions for reproducing the experiments are provided in the supplementary material.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details, hyperparameters, datasets, and evaluation protocols are described in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to computational constraints, main experiments are reported with multiple runs average with no error bars, but consistent trends are observed across settings. Sensitivity study and ablation study report error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix D reports wall-clock runtime using a single NVIDIA RTX A5000 GPU (24 GB).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS Code of Ethics and does not involve sensitive data or human subjects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: While the work improves robustness in collaborative learning, the paper does not explicitly discuss potential negative impacts such as misuse in adversarial settings or fairness considerations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not introduce high-risk models or datasets requiring safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Public datasets and baseline methods are properly cited, and their usage follows standard licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The paper provides source code with documentation in the supplementary material to support reproducibility.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as part of the core methodology.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.