MORPHEUS: MODALITY-AGNOSTIC ADVERSARIAL ATTACK FROM LEAKED MODELS IN FEDERATED LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning (FL) creates a distinctive attack surface by repeatedly exposing intermediate global models to participating clients. A malicious client can retain one such checkpoint as a surrogate and use it to craft adversarial examples (AEs) against the final deployed model, even though the model may undergo subsequent training. In this paper, we investigate an intriguing property, which we term temporal transferability: the ability of an AE crafted against an early checkpoint along a training trajectory to remain effective against the converged model. Although extensive prior work has studied transfer attacks, where AEs crafted against one model are transferred to another, existing approaches are often tailored to specific modalities and do not explicitly account for model evolution over time. We identify two fundamental ingredients that govern temporal transferability: (i) an AE should achieve a sufficiently large adversarial margin, and (ii) it should lie in a region of the leaked model's feature space that is insensitive to parameter changes, making it more likely to remain on the adversarial side of the decision boundary as training progresses. Motivated by these observations, we introduce the weight-space margin (WSM), which quantifies an AE's robustness to changes in model parameters and can be computed using only the leaked checkpoint. Building on this insight, we propose Morpheus, the first modality-agnostic temporal transfer attack. Morpheus optimizes each AE across randomly sampled subnetworks of the leaked checkpoint, reducing its dependence on the exact leaked weights while requiring only the checkpoint and the inputs to be attacked. Across three modalities, six datasets, and six architectures, Morpheus consistently outperforms the strongest of six baseline attacks. When the checkpoint is leaked after only 30% of training, Morpheus improves the attack success rate over the strongest baseline by up to 20.00%.

1 INTRODUCTION

Deep neural networks are vulnerable to adversarial examples (AEs) (Goodfellow et al., 2015; Madry et al., 2018; Carlini & Wagner, 2018; Qin et al., 2019; Xiang et al., 2019; Hamdi et al., 2020). In practice, however, an attacker rarely has access to the parameters of the model it attacks, the victim (also called the target model). A transfer attack instead crafts an AE on a model that the attacker can access, the source model (also called a surrogate), and then submits it to the victim (Waseda et al., 2023). The attack works only if the source model shares vulnerabilities with the victim, but such a source model is hard to obtain even with the victim's architecture and training data.

Federated learning (FL) changes this picture. It can hand the attacker an early copy of the victim itself. FL trains a shared model across many clients without collecting their data, and it has been adopted in settings where privacy, data ownership, or communication costs prevent centralized data collection. Examples include medical imaging across hospitals (Guo et al., 2021; Liu et al., 2021; Xu et al., 2022), speech and clinical audio, which reveal identity and health (Zhang et al., 2023a; Zhong et al., 2025; Wei et al., 2026), and LiDAR perception, whose scans reveal where a vehicle travels (Gan et al., 2026; Ali et al., 2025; Khan et al., 2024). As illustrated in Fig. 1a, a central server coordinates the training. ① In every round, it sends the current global checkpoint to a subset of clients, who update it on their private data and return the update (McMahan et al., 2017). This

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

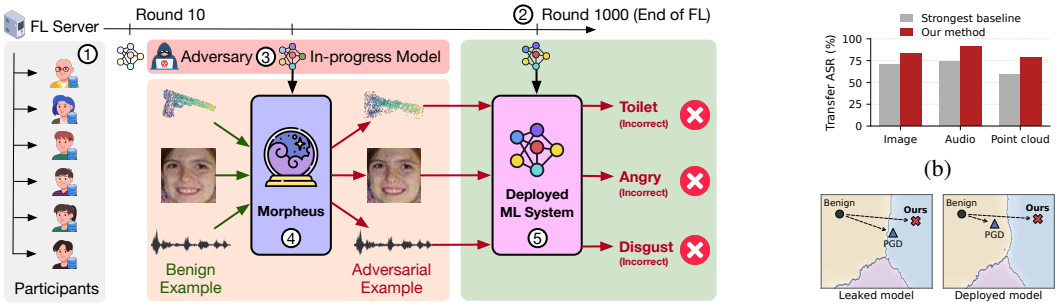


Figure 1: Model leakage in FL yields adversarial examples that survive continued training. (a) FL repeatedly exposes intermediate global models to selected clients before deployment (round numbers are illustrative). A malicious client can retain one such leaked checkpoint and use Morpheus to craft adversarial examples across different modalities that deceive the final deployed model. (b) Transfer ASR at 30% leakage of our method and the strongest baseline on one pair per modality. (c) Schematic of the decision boundary after deployment, where PGD (triangle) reverts and our method's example (cross) remains adversarial.

process repeats until training ends, and ② the server then deploys the final model. However, the same process exposes the in-progress model to clients throughout training, because every selected client must be able to read the checkpoint in order to train on it. A malicious client can therefore ③ keep the checkpoint it receives in an early round, ④ use it offline to craft AEs from its own benign inputs, and ⑤ submit them once the final model is deployed. It needs nothing beyond what FL hands to every selected client.

This paper studies how far a leaked checkpoint can be exploited by a client that holds only the checkpoint and its own inputs. Here the leaked checkpoint is the source model and the final deployed model is the victim. The victim is the source model after further training, so the two share an architecture and a training trajectory and differ only in their weights. AEs crafted on the checkpoint nevertheless lose their effect as training continues. For example, PGD AEs crafted on an early checkpoint of an image classifier (leaked after only 30% of the training rounds) fool this checkpoint on every input it classifies correctly. On the final deployed model, the same AEs fool only 50.54% of the inputs it classifies correctly. This fraction is the transfer attack success rate (transfer ASR). We define this transfer from a leaked checkpoint to the later deployed model as *temporal transfer*, and the ability of an AE to stay effective on the deployed model as its *temporal transferability*. A common way to make AEs transfer better is to transform the input, for example by resizing or translating an image (Athalye et al., 2018; Xie et al., 2019; Guo et al., 2025). Each modality, however, needs its own transformations, designed and tuned separately. Image attacks have been studied for more than a decade (Szegedy et al., 2014) and have many such transformations, whereas attacks on other modalities are much younger and have far fewer. Model-based alternatives need the full training set (Gubri et al., 2022; Li et al., 2023), which in FL is split across clients. This raises a question. *With only an early checkpoint, can a malicious client craft AEs that still work on the final model, for any modality?*

We show that the answer is yes, with one attack that contains nothing specific to any modality. The key insight is that only the weights change between leakage and deployment. An AE that sits just across the decision boundary reverts when training moves the boundary. Two ingredients therefore govern temporal transferability. First, an AE should cross the decision boundary by a large margin. Second, it should lie in a region where the prediction is insensitive to parameter changes, so it stays on the adversarial side as training moves the boundary (Fig. 1c). We introduce the weight-space margin (WSM), which quantifies both ingredients using only the leaked checkpoint. WSM tracks how well AEs transfer, both across attacks and across individual AEs. Building on these insights, we propose Morpheus, a weight-space smoothing attack. It crafts each AE against many subnetworks of the leaked checkpoint, each made by randomly dropping connections, so the AE does not depend on the exact leaked weights. Because Morpheus never transforms the input, it can be applied to any modality without modification, as we demonstrate on image, audio, and point-cloud tasks.

In summary, this paper makes the following contributions.

- We show that model leakage-induced temporal transfer is a modality-agnostic threat. One checkpoint leaked after 30% of training suffices to attack the final deployed model, as we demonstrate on three modalities.
- We identify two ingredients that govern temporal transferability, a large adversarial margin and insensitivity to parameter changes. The weight-space margin (WSM) quantifies both on the leaked checkpoint alone and tracks temporal transferability across attacks and across individual AEs.
- Building on these insights, we propose Morpheus, a modality-agnostic attack that needs only the leaked checkpoint and the inputs being attacked. Morpheus optimizes one perturbation across stochastic subnetworks of the checkpoint, sampled at multiple drop rates.

When leakage occurs after only 30% of training, Morpheus consistently outperforms the strongest of six baseline attacks across three modalities, six datasets, and six architectures. For example, a malicious client selected as early as round 31 reaches 82.80% transfer ASR against an image classifier deployed after more than a hundred rounds, while the strongest baseline yields 70.16%. The gain over the strongest baseline is up to 20.00% in transfer ASR.

2 Background and Related Work

Preliminaries. Federated learning trains a shared model $F_{\boldsymbol{\theta}}$ across distributed clients. At each round r the server sends the current global parameters $\boldsymbol{\theta}_r$ to a selected subset of clients, each client i optimizes them on its private dataset $\mathcal{D}^i$ and returns $\boldsymbol{\theta}_r^i$, and the server aggregates by federated averaging (FedAvg), $\boldsymbol{\theta}_{r+1} = \sum_i |\mathcal{D}^i|\, \boldsymbol{\theta}_r^i / \sum_j |\mathcal{D}^j|$ (McMahan et al., 2017). Training repeats until a fixed horizon or a validation-based stopping rule is met, which forms the global trajectory $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots$, and the selected checkpoint is deployed. An adversarial attack seeks $\boldsymbol{\delta}$ with $\|\boldsymbol{\delta}\|_p \leq \epsilon$ such that $F_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}) \neq y$, by gradient-based maximization of $\mathcal{L}(F_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\delta}), y)$ under white-box access (Goodfellow et al., 2015; Madry et al., 2018). A transfer attack optimizes $\boldsymbol{\delta}$ on the accessible source model $F_{\boldsymbol{\theta}_s}$ and submits it to the unseen target model $F_{\boldsymbol{\theta}_t}$ (Waseda et al., 2023; Shan et al., 2026).

Transfer-Based Adversarial Attacks. Transfer attacks rely on the transferability of AEs from a surrogate to unseen models, and they improve it by reducing surrogate-specific overfitting. Gradient-space methods refine or aggregate update directions, as in MI-FGSM (Dong et al., 2018), NI-FGSM (Lin et al., 2020), and PGN (Ge et al., 2023). Input-space methods average gradients over semantics-preserving views of the input, as in DIM (Xie et al., 2019), TIM (Dong et al., 2019), and SIA (Wang et al., 2023). Model-side methods expand or adapt the surrogate's weights, as in LGV (Gubri et al., 2022), MoreBayesian (Li et al., 2023), and AWT (Chen et al., 2025), and RaPA (Su et al., 2026) combines image transformations with parameter pruning. These methods fall short in our setting. Input-space methods and RaPA rely on image transformations that do not apply to other modalities. LGV and MoreBayesian need additional training data or checkpoints. We use those that can run from one leaked checkpoint as baselines. Morpheus instead targets the weight change itself, which is what FL training produces between leakage and deployment.

Model Leakage-Induced Threats in FL. The model sharing in FL exposes several attack surfaces (Kumar et al., 2024; Lyu et al., 2024). Internal evasion attacks exploit the model components that clients share in personalized FL (Kim et al., 2023b), and related transfer attacks train surrogates with data from compromised clients (Li et al., 2024b). Proteus (Shan et al., 2026) attacks leaked intermediate checkpoints through stochastic image transformations. However, Proteus is vision-only because its image transformations do not carry over to other modalities. To our knowledge, leaked-checkpoint adversarial attacks outside vision have not been studied. Morpheus drops these requirements and needs only what the standard FL protocol already gives every client.

3 Threat Model

Adversary Goal. We consider a malicious FL client who aims to craft AEs that mislead the final deployed model. Using one early checkpoint from the trajectory $\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, \ldots$, the *source* $\boldsymbol{\theta}_s$, the adversary runs an attack algorithm $\mathcal{A}$ to construct $\boldsymbol{\delta} = \mathcal{A}(\boldsymbol{\theta}_s, \boldsymbol{x}, y)$ such that $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$ and $F_{\boldsymbol{\theta}_t}(\boldsymbol{x} + \boldsymbol{\delta}) \neq y$ for the later *target*, or deployed, checkpoint $\boldsymbol{\theta}_t$, where the source round r_s is far earlier than the target round r_t.

Adversary Capability. The adversary is selected in an early round, follows the prescribed FL protocol, and retains the global checkpoint $\boldsymbol{\theta}_s$ it receives. The adversary does not control when it is selected: in cross-device FL, each client is likely to participate no more than once in the training of a given model (Kairouz et al., 2021), so the adversary cannot count on receiving a later checkpoint. The attack is then generated offline and submitted to the deployed model at inference time. The attack and its hyperparameters are fixed before the first submission: the adversary cannot query the deployed model or use the outcomes of submitted AEs to revise them. The adversary also cannot poison client updates or alter server aggregation.

Adversary Knowledge. The adversary has white-box access to $F_{\boldsymbol{\theta}_s}$ and knows the attack input, label, task, and model architecture. In standard FL protocols, this access holds even under secure aggregation (Bonawitz et al., 2017), which hides each client's update from the server while every selected client still has to read the global checkpoint to train on it. The adversary does not observe $F_{\boldsymbol{\theta}_t}$, any subsequent checkpoint, or other clients' private data. The attack uses no training data, not even the adversary's own local data.

4 From Temporal Transferability to Morpheus

Temporal transfer succeeds only if an AE keeps fooling the model through the parameter changes between leakage and deployment. Since the deployed model is unavailable during attack construction, the attacker must judge an AE's temporal transferability from the leaked source model alone. This section first describes what makes an AE survive and how to measure it, and then turns the design principle into an attack objective over stochastic subnetworks of the leaked checkpoint.

4.1 What Makes an AE Survive?

AEs that fool the leaked source model, called source-successful AEs, differ in temporal transferability. We expect an AE to have strong temporal transferability when it combines two ingredients, both measurable on the leaked checkpoint alone. First, it crosses the source model's decision boundary by a large margin. Second, this margin stays large when the weights change slightly. A large margin alone is not enough. Even a wide margin can be fragile when small parameter changes substantially alter the prediction.

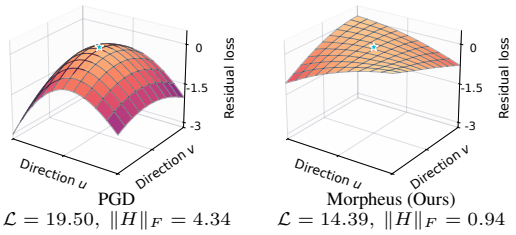


Figure 2: The Morpheus AE lies in a high-loss region with substantially lower local curvature than the PGD AE.

Fig. 2 illustrates in input space how an AE that fools the leaked model can still be fragile. The PGD panel plots the loss landscape around the AE that PGD crafted, where $\mathcal{L}$ is the loss at the AE. The AE sits on a sharp peak (Hessian Frobenius norm $\|H\|_F = 4.34$, a measure of how sharply the loss curves), so the loss drops quickly when the input moves away from it. Continued training changes the parameters rather than the input, so we next define the corresponding quantities in parameter space.

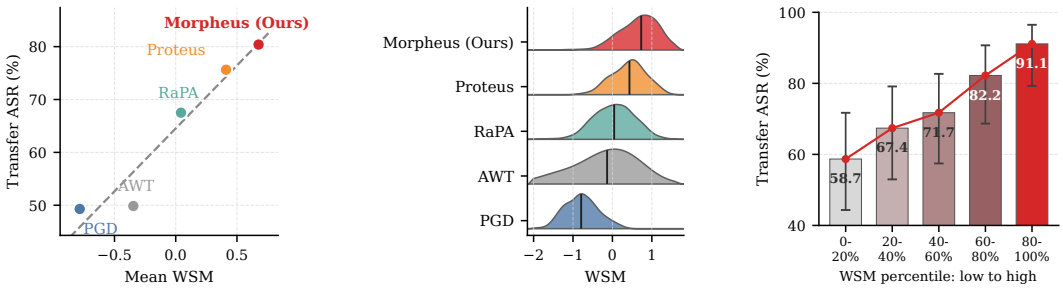
For an input $\boldsymbol{x}$ with true label y, let $z_c(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta})$ denote the logit of class c under weights $\boldsymbol{\theta}$. The *adversarial margin* measures how far the perturbed input crosses the decision boundary:

$$a_{\boldsymbol{\delta}}(\boldsymbol{\theta}) = \max_{c \neq y} z_c(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}) - z_y(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}). \tag{1}$$

The attack succeeds when $a_{\boldsymbol{\delta}}(\boldsymbol{\theta}) > 0$. However, the margin alone does not reveal how sensitive this crossing is to the model parameters.

To measure this sensitivity, we evaluate the same AE after small random perturbations to the source parameters $\boldsymbol{\theta}_s$. Let Q_ρ be a fixed parameter-perturbation distribution, where ρ controls its radius, and let $\boldsymbol{\xi}_\rho \sim Q_\rho$ denote one sampled perturbation. *Parameter sharpness* is the root-mean-square (RMS) change in adversarial margin:

$$S_\rho(\boldsymbol{\delta}) = \left(\mathbb{E}_{\boldsymbol{\xi}_\rho \sim Q_\rho}\left[\left(a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s+\boldsymbol{\xi}_\rho) - a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s)\right)^2\right]\right)^{1/2}. \tag{2}$$



(a) Method-Level Association (b) Per-Example Distribution (c) WSM-Stratified Transfer

Figure 3: Larger source-only WSM_ρ accompanies higher transfer ASR across attacks (a), in the shift of per-example WSM distributions (b), and across WSM percentile groups within Morpheus (c, with 95% Wilson intervals).

A small S_ρ means that nearby parameter changes leave the margin largely unchanged.

The *weight-space margin* (WSM) therefore scores a large margin and a small sharpness together:

$$\text{WSM}_\rho(\boldsymbol{\delta}) = \log \frac{a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s)}{S_\rho(\boldsymbol{\delta})}. \tag{3}$$

A higher WSM makes a random weight change less likely to break the AE. Formally, a weight perturbation from Q_ρ breaks the AE with probability at most $\exp(-2\,\text{WSM}_\rho)$ (Markov's inequality).

We next examine, with leakage at 30% progress, whether WSM is associated with survival under real training drift. Fig. 3 scores five attacks spanning gradient, input-transformation, and model-side families (Madry et al., 2018; Chen et al., 2025; Su et al., 2026; Shan et al., 2026), from which we draw three observations. First, mean WSM_ρ ranks the five attacks in the same order as their transfer ASR (Fig. 3a). Second, more transferable attacks shift the per-example WSM distribution toward larger values (Fig. 3b). Third, within a single attack, transfer ASR rises monotonically across WSM percentile groups (Fig. 3c, trend test $p < 10^{-3}$).

These observations suggest a design principle. An AE with strong temporal transferability should not depend on the exact weights of the leaked checkpoint. Based on these insights, we next describe how to boost temporal transferability and propose Morpheus.

4.2 Weight-Space Smoothing Through Stochastic Subnetworks

To make an AE less dependent on the exact leaked weights, Morpheus evaluates the attack loss on structurally different variants of these weights and favors perturbations whose loss stays high across them. We refer to this strategy as *weight-space smoothing*. As illustrated in Fig. 4, each iteration of Morpheus samples N fresh subnetworks (Secs. 4.2 and 4.3), evaluates the shared perturbation on each of them, averages their input gradients, applies momentum, and projects the update back into the perturbation budget (Sec. 4.4). Empirically, weight-space smoothing raises WSM, and WSM keeps rising with the number of sampled subnetworks N until it saturates beyond $N = 16$ (Fig. 5). This holds even though WSM perturbs the weights with additive Gaussian noise, a distribution that the attack never samples. Fig. 2 also shows the loss landscape around one AE that Morpheus produces, which is much flatter than the PGD AE's ($\|H\|_F = 0.94$ against 4.34). Parameter perturbations have also been used in optimization to find solutions that are less sensitive to nearby weights (Foret et al., 2021; Kwon et al., 2021; Zhuang et al., 2022; Li et al., 2024a). In contrast, Morpheus never updates the weights and optimizes only the input perturbation $\boldsymbol{\delta}$. The subnetworks serve only as different models on which $\boldsymbol{\delta}$ is evaluated.

Morpheus expresses weight variation through a relative quantity, the fraction of dropped connections. Specifically, let $\mathcal{E}_w$ index the eligible scalar connection weights in $\boldsymbol{\theta}_s$, every weight except biases and normalization parameters. For a connection-drop rate $b \in (0, 1)$, Morpheus drops each eligible connection independently with probability b:

$$\Pr[r_e = 0] = b, \quad \Pr[r_e = 1] = 1 - b, \quad e \in \mathcal{E}_w, \qquad \boldsymbol{\theta}^{(b,\boldsymbol{r})} = \mathcal{S}(\boldsymbol{\theta}_s; b, \boldsymbol{r}). \tag{4}$$

Here $\boldsymbol{r}$ is the binary retention mask, and $\mathcal{S}(\boldsymbol{\theta}_s; b, \boldsymbol{r})$ multiplies each eligible weight by its mask entry r_e and leaves all other parameters unchanged. Each resulting $\boldsymbol{\theta}^{(b,\boldsymbol{r})}$ is a temporary subnetwork, used

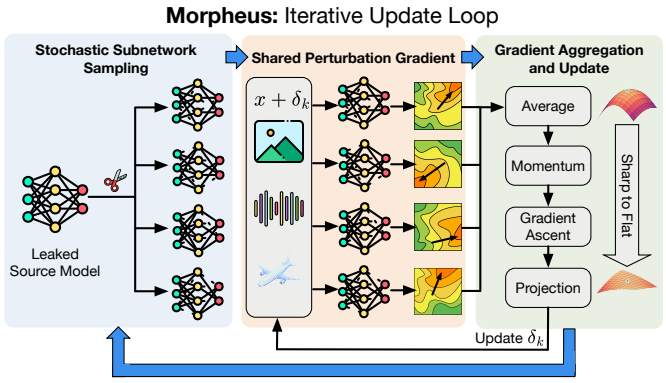


Figure 4: Morpheus iteratively samples stochastic subnetworks of the leaked checkpoint to search for perturbations that remain effective throughout subsequent training.

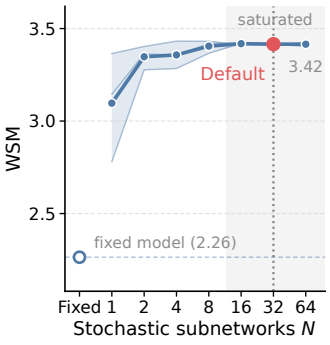


Figure 5: WSM rises with the subnetwork count N and saturates beyond 16 (mean and range over three attack seeds).

for one gradient evaluation and then discarded, so the source checkpoint stays unchanged. What remains is to choose the drop rate b.

4.3 Multi-Rate Stochastic-Subnetwork Objective

A fixed drop rate sets one strength of weight-space smoothing. However, the drift between the leaked and the deployed model can be small or large, and the attacker cannot know which in advance. Tuning the rate on the deployed model would violate the source-only setting. Morpheus therefore samples a fresh rate for every subnetwork from a predefined range, $\log b \sim \mathcal{U}(\log b_{\text{lo}}, \log b_{\text{hi}})$. Log-uniform sampling gives equal weight to equal multiplicative ranges, so the sampled subnetworks cover both slightly and strongly altered versions of the source.

Every sampled subnetwork runs a forward pass on the same perturbed input. Let $\mathcal{L}$ denote the cross-entropy loss maximized by the untargeted attack. Under $\|\boldsymbol{\delta}\|_\infty \le \epsilon$ and $\boldsymbol{x} + \boldsymbol{\delta} \in \mathcal{X}$, Morpheus maximizes

$$J_{\text{M}}(\boldsymbol{\delta}) = \mathbb{E}_{b,\boldsymbol{r}}[\mathcal{L}(F_{\boldsymbol{\theta}^{(b,\boldsymbol{r})}}(\boldsymbol{x} + \boldsymbol{\delta}), y)]\,. \tag{5}$$

Intuitively, $\boldsymbol{\delta}$ is rewarded for raising the loss on average over the sampled subnetworks. A loss increase that relies on one source connection therefore contributes only in the subnetworks that keep it. Directions supported across many connection patterns contribute repeatedly.

4.4 Gradient Aggregation and Adversarial Update

To estimate the objective in Eq. 5, Morpheus samples fresh subnetworks at every attack iteration. At iteration k, let $\boldsymbol{\delta}_k$ be the current perturbation and $\boldsymbol{\theta}_i^{(k)}$ be the i-th of N independently sampled subnetworks. The input gradient from each subnetwork and their raw average are

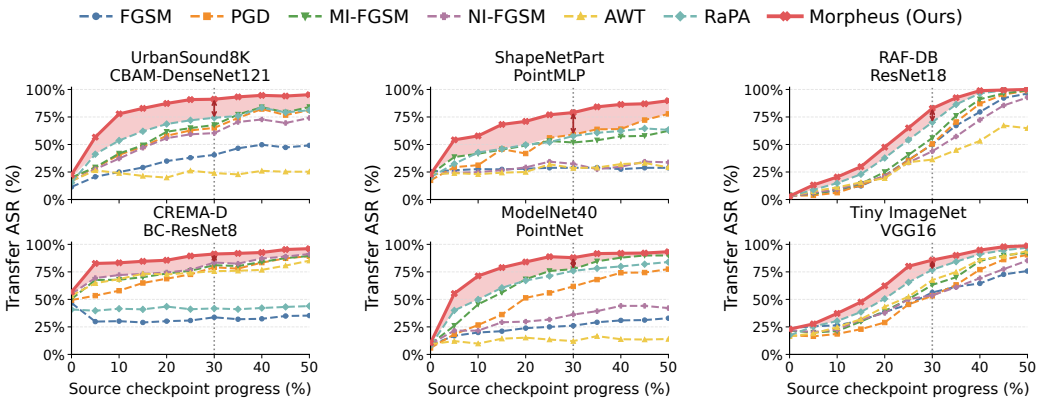
$$\boldsymbol{g}_i^{(k)} = \nabla_{\boldsymbol{\delta}} \mathcal{L}\Big(F_{\boldsymbol{\theta}_i^{(k)}}(\boldsymbol{x} + \boldsymbol{\delta}_k), y\Big)\,, \qquad \widehat{\boldsymbol{g}}_k = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{g}_i^{(k)}. \tag{6}$$

Averaging before normalization keeps the relative magnitudes of the per-subnetwork gradients, so subnetwork-specific components can cancel.

The aggregate gradient is then used in a projected momentum update. Let $\boldsymbol{q}_k$ be the momentum accumulator, μ the momentum coefficient, α the step size, and $\Pi_{\epsilon,\mathcal{X}}$ the projection onto the perturbation constraint and valid input domain. Starting from $\boldsymbol{q}_0 = \mathbf{0}$ and $\boldsymbol{\delta}_0 = \mathbf{0}$, for $k = 0, \ldots, K-1$, Morpheus updates

$$\boldsymbol{q}_{k+1} = \mu \boldsymbol{q}_k + \frac{\widehat{\boldsymbol{g}}_k}{\|\widehat{\boldsymbol{g}}_k\|_1}, \qquad \boldsymbol{\delta}_{k+1} = \Pi_{\epsilon,\mathcal{X}}\left(\boldsymbol{\delta}_k + \alpha\ \text{sign}(\boldsymbol{q}_{k+1})\right). \tag{7}$$

Each iteration costs N forward and backward passes, which can run in parallel. The WSM implementation and survival bound, the drop-rate moments, the estimator analysis, and the complete algorithm are in App. B and App. C.



(a) Audio (b) Point Cloud (c) Image

Figure 6: Morpheus achieves the strongest temporal transferability at most leakage stages across modalities.

5 Evaluation

We evaluate Morpheus for its broad applicability to different modalities, datasets, and neural architectures (Sec. 5.1), its robustness to FL configurations (Sec. 5.2), its resilience under defenses (Sec. 5.3), and the need for each of its components (Sec. 5.4).

Unless otherwise specified, we consider an FL setting in which 100 clients train one global model with FedAvg. In each round, 10 clients are randomly selected, and each performs five local epochs on its IID partition using SGD with a constant learning rate. Training terminates when validation accuracy shows no improvement for 10 consecutive rounds, and the checkpoint with the best validation accuracy is deployed. We measure leakage by training progress, an evaluation coordinate that the attacker does not observe. For a model deployed at round r_t, a leak at progress q gives the source round $r_s = \text{round}(qr_t)$, with $q \in \{0, 0.05, 0.10, \ldots, 1.00\}$ and a default of $q = 0.30$.

Each modality has two datasets and two architectures. Images use RAF-DB (Li et al., 2017) and TinyImageNet (Deng et al., 2009) with ResNet18 (He et al., 2016) and VGG16 (Simonyan & Zisserman, 2015), audio uses CREMA-D (Cao et al., 2014) and UrbanSound8K (Salamon et al., 2014) with BC-ResNet8 (Kim et al., 2021) and CBAM-DenseNet121 (Huang et al., 2017; Woo et al., 2018), and point clouds use ModelNet40 (Wu et al., 2015) and ShapeNetPart (Yi et al., 2016) with PointNet (Qi et al., 2017) and PointMLP (Ma et al., 2022), which gives 12 pairs of dataset and architecture in all (Table 3 in App. D).

We compare Morpheus against six modality-agnostic baselines, namely FGSM (Goodfellow et al., 2015), PGD (Madry et al., 2018), MI-FGSM (Dong et al., 2018), NI-FGSM (Lin et al., 2020), AWT (Chen et al., 2025), and RaPA (Su et al., 2026), which operate directly on the leaked checkpoint without modality-specific input transformations. All attacks use the same source and target checkpoints, evaluation subset, and perturbation budget for each pair. Morpheus uses $N = 32$ stochastic subnetworks per iteration, momentum $\mu = 1$, and connection-drop rates sampled log-uniformly from $[0.005, 0.05]$ on all 12 pairs. For RaPA, we disable its image-specific transformations on all 12 pairs so that one protocol covers every modality, and report the mean over three pre-specified pruning ratios. No attack reads the deployed checkpoint or its predictions while crafting AEs. We report the untargeted transfer ASR, the fraction of inputs with $F_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = y$ for which $F_{\boldsymbol{\theta}_t}(\boldsymbol{x} + \boldsymbol{\delta}) \neq y$. Methods are compared by the absolute difference in this rate. The formal metric, all remaining settings, and the temporal-transfer curves of the six pairs not shown in Fig. 6 are in App. D.

5.1 Temporal Transfer Across Modalities

In each of the 12 FL trajectories, we craft one set of AEs on the checkpoint leaked at each progress q and evaluate every set on the final deployed model of that trajectory. The 30% stage is the primary comparison point, following prior model leakage-induced attack evaluations (Shan et al., 2026).

Fig. 6 shows two pairs per modality that jointly cover both datasets and architectures. Across all 12 pairs, we draw two observations. First, leaked checkpoints remain exploitable even early in training. When leakage occurs after only 30% of training, Morpheus achieves the highest transfer ASR on all 12 pairs. On the six pairs in Fig. 6, it reaches 82.80% and 85.71% on images, 91.12% and 91.23% on audio, and 79.00% and 87.90% on point clouds. Their gains over the strongest baseline are 8.80% to 12.64% on images, 7.80% to 16.85% on audio, and 10.40% to 20.00% on point clouds. Second, the advantage extends over a wide range of leakage stages. Over all 12 pairs, Morpheus is the strongest attack in 81 of the 84 pair-stage cells between 10% and 40% of training, and its average lead over the strongest baseline peaks at 11.3% at 25% progress.

Fig. 7 does the reverse and measures temporal transferability directly. We fix the source at 30% progress of RAF-DB/ResNet18 and generate one set of 512 AEs per attack. The same AEs are evaluated unchanged on every later checkpoint through deployment, and the first column is the zero-drift white-box reference. Morpheus stays effective on every later checkpoint and retains 82.80% transfer ASR on the final deployed model, exceeding the strongest baseline by 12.64%. We additionally show one example per modality in Fig. 8. Morpheus uses the same perturbation budget as PGD, and its AE, which survives to the deployed model, stays close to the clean input.

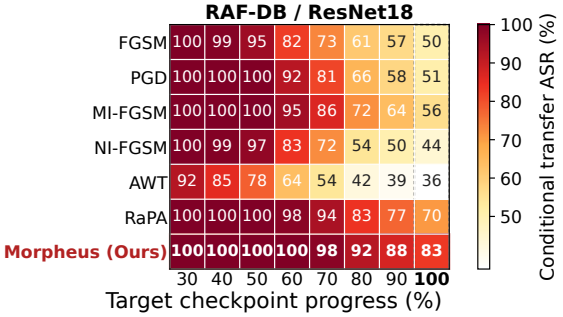


Figure 7: Morpheus AEs generated at 30% progress remain effective on every later checkpoint.

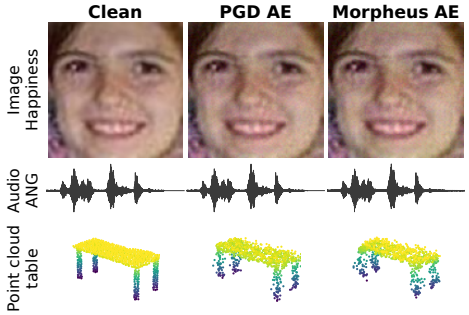
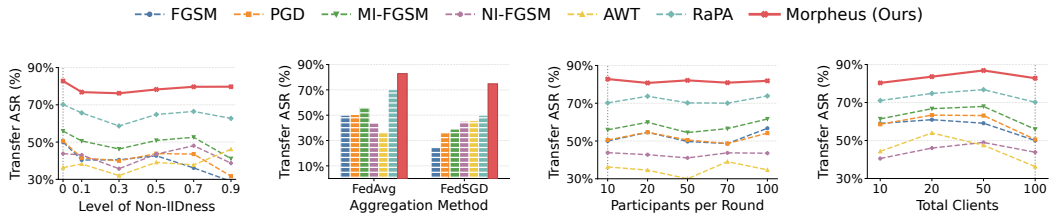


Figure 8: Morpheus AEs use the budget of PGD and transfer to the final model in all three modalities.

5.2 Temporal Transfer Across FL Configurations

FL deployments differ in data distribution, aggregation rule, and scale, which changes the training trajectory that an AE must survive. We therefore evaluate Morpheus on RAF-DB/ResNet18 under such variations, where each configuration is trained as a separate FL run. We first vary client-data heterogeneity, expressed as non-IIDness $1 - \alpha$, with $1 - \alpha = 0$ the IID anchor and the remaining points Dirichlet partitions with $\alpha \in \{0.9, 0.7, 0.5, 0.3, 0.1\}$. As shown in Fig. 9a, Morpheus ranks first throughout the sweep, leading the strongest baseline by 11.1% to 17.5%. We then compare FedAvg with FedSGD (McMahan et al., 2017), vary participating clients per round over $\{10, 20, 50, 70, 100\}$ with 100 total clients, and vary the total client population over $\{10, 20, 50, 100\}$ with 10 participants per round (Figs. 9b to 9d). The lead over the strongest baseline is 24.8% for FedSGD, 7.0% to 12.6% across participation rates, and 8.8% to 12.6% across client populations. Overall, Morpheus ranks first in all 14 distinct settings of this subsection (the default is shared across sweeps) and outperforms the strongest baseline by 12.6% on average.



(a) Data Distribution (b) FL Aggregation (c) FL Participants (d) FL Clients

Figure 9: Morpheus maintains the strongest temporal transferability across diverse FL configurations.

5.3 Temporal Transfer Under Defenses

We evaluate three training-free, modality-agnostic inference-time defenses, namely Gaussian logit smoothing, single-query Gaussian input randomization (Qin et al., 2021; Byun et al., 2022), and post-hoc randomized smoothing (Cohen et al., 2019). Each is set to the noise level that costs about 5 points of clean accuracy. We also evaluate the protocol-level defense FedSlice (Zhang et al., 2023b), which sends each client only a slice of the model. Fig. 10 reports the results on RAF-DB/ResNet18, from which we observe two findings. First, the three inference-time defenses preserve 66.43% to 67.37% clean accuracy but leave Morpheus at 80.06% to 81.84% ASR, at least 14.92% above every competing attack. Second, FedSlice reduces Morpheus to 4.27% ASR, but its 40.84% clean accuracy is only 2.22% above the 38.62% majority-class baseline. FedSlice therefore lowers transfer ASR only at the cost of a model that no longer works.

5.4 Ablation Study

We conduct three analyses on RAF-DB/ResNet18 with leakage at 30% progress. Table 1 compares no subnetwork sampling, the mean over three fixed drop rates, and linear- and log-uniform multi-rate sampling. Fig. 11 varies $N \in \{4, 8, 16, 32, 64\}$ under the full method, both as single runs on the same 512 inputs. Fig. 12 varies the sampled range $[b_{\text{lo}}, b_{\text{hi}}]$ itself and reports the mean over three seeds per setting, each seed drawing its own 512 inputs. All other attack settings remain fixed.

Table 1: Component ablation.

Variant	ASR (%)
w/o subnetworks	55.91
w/o multi-rate	75.36
w/o log-uniform	79.57
Full Morpheus	**82.80**

Removing subnetwork sampling, which reduces Morpheus to MI-FGSM, lowers transfer ASR from 82.80% to 55.91%. Fixed-rate subnetworks average 75.36% across the rates 0.005, 0.016, and 0.05, while linear- and log-uniform sampling reach 79.57% and 82.80%. Thus log-uniform sampling improves over the fixed-rate mean by 7.44% without selecting a rate using the unknown deployed model. Increasing N from 4 to 8 raises ASR from 79.30% to 81.45%, and ASR plateaus at 82.80% from $N = 32$ onward (Fig. 11).

Transfer is insensitive to the exact range. Varying its width from 2.5× to 200× or shifting it by an octave moves transfer ASR by at most 3.49%, from 78.47% to 81.96%, whereas a nearly fixed rate spans 17.47%, from 64.50% at a rate of 0.05 to 81.97% at 0.016 (Fig. 12). A single rate matches the best sampled range only if it is the best rate. In our experiments, that rate is identified only by evaluating transfer on the deployed model, and the threat model withholds this feedback.

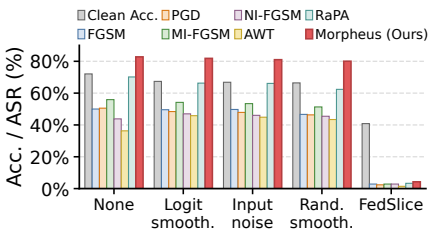


Figure 10: Only FedSlice cuts ASR, but it collapses clean accuracy.

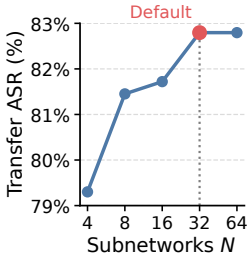


Figure 11: ASR plateaus from N=32.

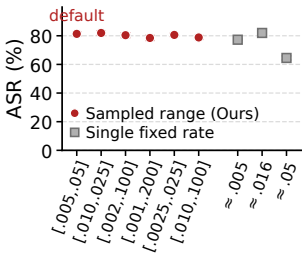


Figure 12: Ranges are robust, fixed rates not.

6 Conclusions

We presented Morpheus, a modality-agnostic adversarial attack for model leakage-induced temporal transfer in FL. The weight-space margin diagnoses temporal transferability from the leaked checkpoint alone, and Morpheus widens that margin through weight-space smoothing over stochastic subnetworks. Morpheus AEs from early checkpoints stay effective across three modalities. On RAF-DB/ResNet18, they also stay effective across the FL configurations we varied and under inference-time defenses. No evaluated defense reduces temporal transfer without impairing clean utility, so defending against model leakage-induced attacks remains an open problem.

AI USE STATEMENT

In this work, we used generative AI tools for text polishing, proofreading, adjusting figure sizes, LaTeX formatting, and organizing the anonymized code release. The authors have manually verified all AI-assisted edits and take full responsibility for the final manuscript.

ETHICS STATEMENT

This paper describes an attack that a malicious FL client can mount with a checkpoint it receives during ordinary training. We study this attack to help build more robust FL systems, since a defense can be designed and tested only against a known threat. Sec. 5.3 evaluates inference-time and protocol-level defenses against it. No evaluated defense stops the attack without a large loss of clean accuracy, so defending against this threat remains an open problem. All experiments use public research datasets under their license terms and FL models that we trained ourselves in simulation. RAF-DB contains face images and CREMA-D contains recorded speech of actors, and we use both only for emotion recognition, without any attempt to identify individuals. No deployed system or real user was attacked. We release the code so that future defenses can be evaluated against this attack.

REPRODUCIBILITY STATEMENT

Code for Morpheus is available at `https://anonymous.4open.science/r/morpheus-1E79/`. Algorithm 1 in App. C states the full attack, and App. C.4 gives its optimization details. App. B.1 derives the WSM survival bound stated in Sec. 4.1. App. D lists the training recipe, deployment checkpoint, and perturbation budget of each pair (Table 3), the evaluation subsets, the formal transfer ASR, and the settings of every attack (App. D.4), and the settings of the FL-configuration, ablation, and defense experiments (Apps. D.5 to D.7).

REFERENCES

Asad Ali, Huang Jianjun, and Ayesha Jabbar. Recent advances in federated learning for connected autonomous vehicles: Addressing privacy, performance, and scalability challenges. *IEEE Access*, 13:80637–80665, 2025. doi: 10.1109/ACCESS.2025.3562128.

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 284–293. PMLR, 2018.

Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 1175–1191, 2017.

Junyoung Byun, Hyojun Go, and Changick Kim. On the effectiveness of small input noise for defending against query-based black-box attacks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3051–3060, 2022.

Houwei Cao, David G. Cooper, Michael K. Keutmann, Ruben C. Gur, Ani Nenkova, and Ragini Verma. CREMA-D: Crowd-sourced emotional multimodal actors dataset. *IEEE Transactions on Affective Computing*, 5(4):377–390, 2014.

Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7. IEEE, 2018.

Guangke Chen, Yedi Zhang, Zhe Zhao, and Fu Song. QFA2SR: Query-Free adversarial transfer attacks to speaker recognition systems. In *Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23)*, pp. 2437–2454. USENIX Association, 2023.

Jiahao Chen, Zhou Feng, Rui Zeng, Yuwen Pu, Chunyi Zhou, Yi Jiang, Yuyou Gan, Jinbao Li, and Shouling Ji. Enhancing adversarial transferability with adversarial weight tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pp. 2061–2069, 2025.

Jeremy Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320. PMLR, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. IEEE, 2009.

Jiahua Dong, Duzhen Zhang, Yang Cong, Wei Cong, Henghui Ding, and Dengxin Dai. Federated incremental semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3934–3943, 2023.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9185–9193, 2018.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4312–4321, 2019.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations (ICLR)*, 2021.

Fuquan Gan and Yan Wo. Boosting the transferability of adversarial examples through gradient aggregation. *IEEE Transactions on Information Forensics and Security (TIFS)*, 20:5563–5576, 2025. doi: 10.1109/TIFS.2025.3574989.

Maolin Gan, Li Liu, Samiul Alam, Jiancheng Liu, Mi Zhang, Huacheng Zeng, and Zhichao Cao. 3D-FAD: Informativeness-guided multimodal 3D object detection for federated autonomous driving. *ACM Transactions on Sensor Networks*, 22(4):1–29, 2026.

Liang Gao, Huazhu Fu, Li Li, Yingwen Chen, Ming Xu, and Cheng-Zhong Xu. FedDC: Federated learning with non-IID data via local drift decoupling and correction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10112–10121, 2022.

Zhijin Ge, Hongying Liu, Xiaosen Wang, Fanhua Shang, and Yuanyuan Liu. Boosting adversarial transferability by achieving flat local maxima. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Martin Gubri, Maxime Cordy, Mike Papadakis, Yves Le Traon, and Koushik Sen. LGV: Boosting adversarial example transferability from large geometric vicinity. In *European Conference on Computer Vision (ECCV)*, pp. 603–618. Springer, 2022.

Pengfei Guo, Puyang Wang, Jinyuan Zhou, Shanshan Jiang, and Vishal M. Patel. Multi-institutional collaborations for improving deep learning-based magnetic resonance image reconstruction using federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2423–2432, 2021.

Ping Guo, Tiantian Zhang, Xi Lin, Xiang Li, Zhi-Ri Tang, and Qingfu Zhang. Few-for-many personalized federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17515–17524, 2026.

Yu Guo, Weiquan Liu, Qingshan Xu, Shijun Zheng, Shujun Huang, Yu Zang, Siqi Shen, Chenglu Wen, and Cheng Wang. Boosting adversarial transferability through augmentation in hypothesis space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19175–19185, 2025.

Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. AdvPC: Transferable adversarial perturbations on 3D point clouds. In *European Conference on Computer Vision (ECCV)*, pp. 241–257. Springer, 2020.

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 (CVPR)*, pp. 770–778, 2016.

Mengze Hong, Yi Gu, Di Jiang, Hanlin Gu, Chen Jason Zhang, Lu Wang, and Zhiyang Su. Federated heterogeneous language model optimization for hybrid automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 18477–18481. IEEE, 2026. doi: 10.1109/ICASSP55912.2026.11460729.

Wenjie Hou, Tianxiang Chen, Feng Wang, Tiantong Wu, Zhiming Zheng, Shaoting Tang, and Wei Yang Bryan Lim. FedAdamom: Adaptive momentum for improved generalization in federated optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36354–36364, 2026.

Qianjiang Hu, Daizong Liu, and Wei Hu. Exploring the devil in graph spectral domain for 3D point cloud attacks. In *European Conference on Computer Vision (ECCV)*, pp. 229–248. Springer, 2022.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017.

Qidong Huang, Xiaoyi Dong, Dongdong Chen, Hang Zhou, Weiming Zhang, and Nenghai Yu. Shape-invariant 3D adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15335–15344, 2022.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations (ICLR)*, 2020.

Peter Kairouz, H. Brendan McMahan, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

Woonsang Kang, Joohyung Lee, Seungjun Kim, Jungchan Cho, and Yoonseon Oh. Communication-efficient module-wise federated learning for grasp pose detection in cluttered environments. *IEEE Robotics and Automation Letters*, 11(2):1234–1241, 2026. doi: 10.1109/LRA.2025.3641101.

Abdullah Aman Khan, Khwaja Mutahir Ahmad, Sidra Shafiq, Waqas Amin, and Rajesh Kumar. 3DFFL: Privacy-preserving federated few-shot learning for 3D point clouds in autonomous vehicles. *Scientific Reports*, 14(1):19589, 2024.

Byeonggeun Kim, Simyung Chang, Jinkyu Lee, and Dooyong Sung. Broadcasted residual learning for efficient keyword spotting. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 4538–4542, 2021.

Hoki Kim, Jinseong Park, and Jaewook Lee. Generating transferable adversarial examples for speech classification. *Pattern Recognition*, 137:109286, 2023a.

Taejin Kim, Shubhranshu Singh, Nikhil Madaan, and Carlee Joe-Wong. Characterizing internal evasion attacks in federated learning. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 206 of *Proceedings of Machine Learning Research*, pp. 907–921. PMLR, 2023b.

Kummari Naveen Kumar, Chalavadi Krishna Mohan, and Linga Reddy Cenkeramaddi. The impact of adversarial attacks on federated learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 46(5):2672–2691, 2024.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5905–5914. PMLR, 2021.

Huy Q. Le, Loc X. Nguyen, Yu Qiao, Seong Tae Kim, Eui-Nam Huh, and Choong Seon Hong. FedDAP: Domain-aware prototype learning for federated learning under domain shift. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3390–3399, 2026.

Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10713–10722, 2021.

Qizhang Li, Yiwen Guo, Wangmeng Zuo, and Hao Chen. Making substitute models more Bayesian can enhance transferability of adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2023.

Shan Li, Weihong Deng, and JunPing Du. Reliable crowdsourcing and deep locality-preserving learning for expression recognition in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2852–2861, 2017.

Tao Li, Pan Zhou, Zhengbao He, Xinwen Cheng, and Xiaolin Huang. Friendly sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5631–5640, 2024a.

Yijiang Li, Ying Gao, and Haohan Wang. Towards understanding adversarial transferability in federated learning. *Transactions on Machine Learning Research (TMLR)*, 2024b.

Chenchen Lin, Wenhao Yuan, Zhengji Xu, and Xuehe Wang. Domain sensitive federated learning with Fisher-informed pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17535–17544, 2026.

Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2020.

Qinliang Lin, Cheng Luo, Zenghao Niu, Xilin He, Weicheng Xie, Yuanbo Hou, Linlin Shen, and Siyang Song. Boosting adversarial transferability across model genus by deformation-constrained warping. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pp. 3459–3467, 2024.

Quande Liu, Cheng Chen, Jing Qin, Qi Dou, and Pheng-Ann Heng. FedDG: Federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1013–1023, 2021.

Xingyu Lyu, Shixiong Li, Ning Wang, Tao Li, Danjue Chen, and Yimin Chen. Adversarial attacks on federated learning revisited: A client-selection perspective. In *2024 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9. IEEE, 2024.

Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual MLP framework. In *International Conference on Learning Representations (ICLR)*, 2022.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

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 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1273–1282. PMLR, 2017.

Jiaxu Miao, Zongxin Yang, Leilei Fan, and Yi Yang. FedSeg: Class-heterogeneous federated learning for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8042–8052, 2023.

Geofrey Owino, Bernard Shibwabo Kasamani, Ahmed M. Abdelmoniem, and Edem Wornyo. Federated primitive-preserving audio transformers for non-identifiable infant cry classification. *IEEE Access*, 14:63644–63660, 2026. doi: 10.1109/ACCESS.2026.3686735.

Shuchao Pang, Zhenghan Chen, Shen Zhang, Liming Lu, Siyuan Liang, Anan Du, and Yongbin Zhou. Towards a 3D transfer-based black-box attack via critical feature guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 26912–26922, 2025.

Krishna Pillutla, Kshitiz Malik, Abdel-Rahman Mohamed, Mike Rabbat, Maziar Sanjabi, and Lin Xiao. Federated learning with partial model personalization. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17716–17758. PMLR, 2022.

Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 652–660, 2017.

Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5231–5240. PMLR, 2019.

Zeyu Qin, Yanbo Fan, Hongyuan Zha, and Baoyuan Wu. Random noise defense against query-based black-box attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 7650–7663, 2021.

Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia (ACM MM)*, pp. 1041–1044, 2014.

Junjie Shan, Yue Zhang, Ziqi Zhao, and Ka-Ho Chow. Proteus: Model leakage-induced adversarial attack in federated learning. In *European Conference on Computer Vision (ECCV)*, pp. 225–243. Springer, 2026. doi: 10.1007/978-3-032-37356-4_13.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Tongrui Su, Qingbin Li, Shengyu Zhu, Wei Chen, and Xueqi Cheng. RaPA: Enhancing transferable targeted attacks via random parameter pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6538–6548, 2026.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Zilin Tian, Yunfei Long, Liguo Zhang, and Jiahong Zhao. Enhancing transferability of audio adversarial example for both frequency- and time-domain. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6263–6271, 2025.

Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-57959-7.

Kunyu Wang, Xuanran He, Wenxuan Wang, and Xiaosen Wang. Boosting adversarial transferability by block shuffle and rotation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24336–24346, 2024.

Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16158–16167, 2021.

Xiaosen Wang, Zeliang Zhang, and Jianping Zhang. Structure invariant transformation for better adversarial transferability. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4607–4619, 2023.

Futa Waseda, Sosuke Nishikawa, Trung-Nghia Le, Huy H. Nguyen, and Isao Echizen. Closer look at the transferability of adversarial examples: How they fool different models differently. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 1360–1368, 2023.

Xiao Wei, Bin Wen, Yuqin Lin, Kai Li, Mingyang Gu, Xiaobao Wang, Longbiao Wang, and Jianwu Dang. Breaking data efficiency dilemma: A federated and augmented learning framework for Alzheimer's disease detection via speech. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 19147–19151. IEEE, 2026.

Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1912–1920, 2015.

Chong Xiang, Charles R. Qi, and Bo Li. Generating 3D adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9136–9144, 2019.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2730–2739, 2019.

An Xu, Wenqi Li, Pengfei Guo, Dong Yang, Holger R. Roth, Ali Hatamizadeh, Can Zhao, Daguang Xu, Heng Huang, and Ziyue Xu. Closing the generalization gap of cross-silo federated medical image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20866–20875, 2022.

Anshun Xue, Xinghua Zhu, Jianzong Wang, and Jing Xiao. Diversified point cloud classification using personalized federated learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021.

Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics (TOG)*, 35(6):210:1–210:12, 2016. doi: 10.1145/2980179.2980238.

Tuo Zhang, Tiantian Feng, Samiul Alam, Sunwoo Lee, Mi Zhang, Shrikanth S. Narayanan, and Salman Avestimehr. FedAudio: A federated learning benchmark for audio tasks. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023a.

Ziqi Zhang, Yuanchun Li, Bingyan Liu, Yifeng Cai, Ding Li, Yao Guo, and Xiangqun Chen. FedSlice: Protecting federated learning models from malicious participants with model slicing. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pp. 460–472. IEEE, 2023b.

Tao Zhong, Mengzhe Geng, Shujie Hu, Guinan Li, and Xunying Liu. Regularized federated learning for privacy-preserving dysarthric and elderly speech recognition. In *Annual Conference of the International Speech Communication Association (INTERSPEECH)*, pp. 2103–2107, 2025.

Hegui Zhu, Yuchen Ren, Xiaoyan Sui, Lianping Yang, and Wuming Jiang. Boosting adversarial transferability via gradient relevance attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4741–4750, 2023.

Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C. Dvornek, Sekhar Tatikonda, James S. Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations (ICLR)*, 2022.

Chen-Chen Zong and Sheng-Jun Huang. Federated active learning under extreme non-IID and global class imbalance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24534–24544, 2026.

A Background and Additional Related Work

FL deployments across modalities. FL is used across vision, audio, and 3D sensing, where privacy, data ownership, and communication costs often prevent centralized data collection (Gao et al., 2022; Li et al., 2021; Guo et al., 2026; Le et al., 2026; Hou et al., 2026; Lin et al., 2026; Zong & Huang, 2026). Beyond medical imaging under data-protection rules (Voigt & von dem Bussche, 2017), vision deployments include scene segmentation across data silos (Miao et al., 2023; Dong et al., 2023). Federated audio systems cover keyword spotting, emotion recognition, and sound event classification (Zhang et al., 2023a), speech recognition (Pillutla et al., 2022; Hong et al., 2026), and infant cry analysis (Owino et al., 2026). In 3D sensing, FL supports shape understanding and robotic grasping without centralizing raw scans (Xue et al., 2021; Kang et al., 2026).

Gradient and input-space diversification. Gradient-only attacks improve or aggregate update directions without semantic input transformations. Representative methods include MI-FGSM (Dong et al., 2018), NI-FGSM (Lin et al., 2020), PGN (Ge et al., 2023), GRA (Zhu et al., 2023), and GAA (Gan & Wo, 2025). Input-space attacks instead optimize over transformed examples. This family includes DIM (Xie et al., 2019), TIM (Dong et al., 2019), SIM (Lin et al., 2020), Admix (Wang et al., 2021), SIA (Wang et al., 2023), BSR (Wang et al., 2024), and DeCoWA (Lin et al., 2024). Their resize, translation, mixing, block, or deformation views are defined through the structure and semantics of the input modality.

Transfer attacks beyond images. Audio transfer has been studied for speech classification, speaker recognition, and joint frequency- and time-domain optimization (Kim et al., 2023a; Chen et al., 2023; Tian et al., 2025). Point-cloud attacks improve cross-model transfer through reconstruction objectives, shape-aware constraints, graph-spectral structure, and critical-feature guidance (Hamdi et al., 2020; Huang et al., 2022; Hu et al., 2022; Pang et al., 2025). These results establish transfer threats outside image classification, while relying on modality-specific waveform, time-frequency, or 3D geometric structure.

Model-side diversification. LGV (Gubri et al., 2022) and MoreBayesian (Li et al., 2023) build surrogate ensembles by continuing to train the surrogate, which needs training data and yields multiple checkpoints. AWT (Chen et al., 2025) tunes the surrogate with generated AEs to promote flat local maxima and model smoothness. RaPA (Su et al., 2026) combines image transformations with random pruning of parameters in batch-normalization and linear layers. These methods motivate model-side diversification, but their full protocols assume additional data, multiple checkpoints, surrogate tuning, or modality-specific transformations. In our evaluation, AWT and RaPA run from the leaked checkpoint and the attacked inputs alone, with RaPA's image transformations removed. Temporal transfer instead requires a construction that operates from one leaked checkpoint without target-side selection.

Margin and sharpness measures. Margin- and sharpness-based quantities are widely studied as predictors of generalization for trained networks (Bartlett et al., 2017; Jiang et al., 2020; Foret et al., 2021). WSM applies the same two ingredients to a single adversarial example instead of a trained model: the adversarial margin measures how far the example crosses the decision boundary, and parameter sharpness measures how much this margin changes under weight perturbations of the leaked checkpoint.

B Weight-Space Margin Analysis

B.1 Detailed derivation of the WSM survival bound

Fix a source-successful AE and abbreviate its positive source margin by

$$A = a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s) > 0. \tag{8}$$

For one random parameter perturbation $\boldsymbol{\xi}_\rho \sim Q_\rho$, define the corresponding margin change and failure event as

$$D(\boldsymbol{\xi}_\rho) = a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s + \boldsymbol{\xi}_\rho) - A, \qquad \mathcal{F} = \{a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s + \boldsymbol{\xi}_\rho) \le 0\}. \tag{9}$$

On $\mathcal{F}$, substituting $a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s + \boldsymbol{\xi}_\rho) = A + D(\boldsymbol{\xi}_\rho)$ gives

$$A + D(\boldsymbol{\xi}_\rho) \le 0 \implies D(\boldsymbol{\xi}_\rho) \le -A \implies |D(\boldsymbol{\xi}_\rho)| \ge A. \tag{10}$$

Hence $D(\boldsymbol{\xi}_\rho)^2/A^2 \geq 1$ whenever failure occurs. Outside $\mathcal{F}$, the failure indicator is zero. The following pointwise inequality therefore holds for every sampled perturbation:

$$\mathbf{1}_{\mathcal{F}} \leq \frac{D(\boldsymbol{\xi}_\rho)^2}{A^2}. \tag{11}$$

Taking expectations preserves the inequality and yields

$$\begin{aligned} \Pr_{Q_\rho}(\mathcal{F}) &= \mathbb{E}_{Q_\rho}[\mathbf{1}_{\mathcal{F}}] && (12)\\ &\leq \frac{\mathbb{E}_{Q_\rho}[D(\boldsymbol{\xi}_\rho)^2]}{A^2} = \frac{S_\rho(\boldsymbol{\delta})^2}{A^2} && (13)\\ &= \exp\left(-2\log\frac{A}{S_\rho(\boldsymbol{\delta})}\right) = \exp(-2\mathrm{WSM}_\rho(\boldsymbol{\delta}))\,. && (14) \end{aligned}$$

The second line substitutes the RMS definition in Eq. 2, and the final line substitutes Eq. 3. The argument requires neither differentiability nor local linearity.

The bound concerns failure under weight perturbations drawn from the fixed procedure Q_ρ and makes no statement about training drift. The temporal experiments separately evaluate whether the resulting source-side score reflects survival along real FL trajectories.

B.2 Computing and interpreting WSM

The random parameter perturbations used for measurement are fixed before transfer evaluation. Let $\mathcal{W}$ index the connection-bearing weight tensors. For every $\ell \in \mathcal{W}$, independently draw $\boldsymbol{u}^{(\ell)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and define

$$\boldsymbol{\xi}_\rho^{(\ell)} = \rho\|\boldsymbol{\theta}_s^{(\ell)}\|_2 \frac{\boldsymbol{u}^{(\ell)}}{\|\boldsymbol{u}^{(\ell)}\|_2}. \tag{15}$$

We use radius $\rho = 0.02$ and hold biases, normalization parameters, and running statistics fixed. We use additive perturbations rather than the attack's stochastic subnetworks so that the score is defined independently of any attack's own randomization. Every attack for a dataset–architecture pair is evaluated with the same radius and sampled directions, which are drawn once before any deployed checkpoint is evaluated, so the score is neither attack-specific nor selected on the deployed model.

For N_{diag} independent directions $\boldsymbol{\xi}_{\rho,1}, \ldots, \boldsymbol{\xi}_{\rho,N_{\mathrm{diag}}} \sim Q_\rho$, the directly executable estimator of Eq. 2 is

$$\begin{aligned} d_{\rho,k}(\boldsymbol{\delta}) &= a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s + \boldsymbol{\xi}_{\rho,k}) - a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s),\\ \widehat{S}_{\rho,N_{\mathrm{diag}}}(\boldsymbol{\delta}) &= \left(\frac{1}{N_{\mathrm{diag}}}\sum_{k=1}^{N_{\mathrm{diag}}} d_{\rho,k}(\boldsymbol{\delta})^2\right)^{1/2}. \end{aligned} \tag{16}$$

Writing $\widehat{S}^2_{\rho,N_{\mathrm{diag}}} = N_{\mathrm{diag}}^{-1}\sum_k d^2_{\rho,k}$, independence gives the intermediate estimator properties

$$\mathbb{E}[\widehat{S}^2_{\rho,N_{\mathrm{diag}}}] = S^2_\rho, \qquad \mathrm{Var}[\widehat{S}^2_{\rho,N_{\mathrm{diag}}}] = \frac{\mathrm{Var}[d^2_{\rho,1}]}{N_{\mathrm{diag}}}. \tag{17}$$

Thus the squared estimator is unbiased and its variance decreases as $1/N_{\mathrm{diag}}$. The square root is a consistent plug-in estimator.

Define $A(\boldsymbol{\delta}) = a_{\boldsymbol{\delta}}(\boldsymbol{\theta}_s)$. The practical per-example score is

$$\widehat{\mathrm{WSM}}_{\rho,N_{\mathrm{diag}}}(\boldsymbol{\delta}) = \log\big(A(\boldsymbol{\delta}) + \varepsilon\big) - \log\big(\widehat{S}_{\rho,N_{\mathrm{diag}}}(\boldsymbol{\delta}) + \varepsilon\big), \tag{18}$$

where ε is a fixed numerical stabilizer. Only source-successful AEs, for which $A(\boldsymbol{\delta}) > 0$, enter the WSM analysis. Equation 18 is the raw log-ratio corresponding to the theoretical definition.

Fig. 3 compares heterogeneous attacks whose log-margin and log-sharpness ranges differ. For this empirical visualization, let $u_{m,i} = \log(A_{m,i} + \varepsilon)$ and $v_{m,i} = \log(\widehat{S}_{\rho,N_{\mathrm{diag}},m,i} + \varepsilon)$ for source-successful example i from attack m. Over the displayed set $\mathcal{H} = \{$PGD, AWT, RaPA, Proteus,

Morpheus} of Fig. 3, the source-only equal-method moments are

$$\mu_u = \frac{1}{|\mathcal{H}|}\sum_{m\in\mathcal{H}}\frac{1}{n_m}\sum_{i=1}^{n_m} u_{m,i},$$
$$\sigma_u^2 = \frac{1}{|\mathcal{H}|}\sum_{m\in\mathcal{H}}\frac{1}{n_m}\sum_{i=1}^{n_m}(u_{m,i}-\mu_u)^2, \tag{19}$$

with μ_v and σ_v defined analogously. The normalized empirical score is

$$\widetilde{\mathrm{WSM}}_\rho(m,i) = \frac{u_{m,i}-\mu_u}{\sigma_u} - \frac{v_{m,i}-\mu_v}{\sigma_v}. \tag{20}$$

Each attack contributes equal weight to the normalizers. The three fixed RaPA ratios are pooled within the single RaPA group, so their additional samples do not give RaPA greater method-level weight. Fig. 3 uses $N_{\mathrm{diag}} = 32$ and $\rho = 0.02$. It and Table 2 come from a matched AE set generated separately for this diagnostic at 30% progress of the RAF-DB/ResNet18 run, with AWT at SAM radius 1.5×10^{-3} and RaPA at step $\epsilon/20$.

Fig. 5 reports the raw log-ratio of Eq. 18, without standardization, averaged over the source-successful AEs of each configuration on the RAF-DB/ResNet18 pair. It uses the same 512 inputs, three attack seeds, $\rho = 0.02$ and $N_{\mathrm{diag}} = 32$, and no target predictions.

All terms in Eqs. 16 through 20, including the normalizers, use only the leaked checkpoint and source-successful examples. Target predictions enter only after the score is frozen. Panel (a) uses the standard target-clean transfer ASR. Panel (c) first ranks source-successful, target-clean Morpheus examples by their frozen source-only score, divides them into five equal-sized groups, and then uses target predictions only to compute each group's transfer ASR.

Trend test for Fig. 3c. The five groups contain 46, 46, 46, 45, and 45 examples, of which 27, 31, 33, 37, and 41 transfer. A Cochran–Armitage trend test over the group index gives $Z = 3.88$ (two-sided $p = 1.1 \times 10^{-4}$), treating the 228 examples as independent draws from one set.

Margin and sharpness separately. Margin alone mis-orders the attacks. On the RAF-DB/ResNet18 pair of Fig. 5, PGD AEs have the larger mean adversarial margin, 18.6 logits against 14.7 for Morpheus with $N = 32$, but their margin fluctuates 1.98 logits under weight perturbations against 0.48 for Morpheus, which gives mean per-example raw WSM values of 2.26 and 3.42.

Outcome split. Table 2 reports, for each attack, the number n of source-successful AEs, the transfer ASR over the 357 inputs that the target classifies correctly, and the median normalized WSM of the source-successful AEs on such inputs, split into AEs that fail and AEs that transfer to this target. RaPA pools its three pruning ratios. Within every attack the transferred AEs carry the higher median.

Table 2: Source-successful AEs (n), transfer ASR over the inputs classified correctly by the target, and median normalized WSM of failed and transferred AEs among the source-successful AEs on those inputs, on RAF-DB/ResNet18 at the 30% leakage point.

Attack	n	Transfer ASR	Failed	Transferred
PGD	255	0.493	−0.821	−0.707
AWT	252	0.499	−0.343	0.149
RaPA	765	0.675	−0.087	0.153
Proteus	255	0.756	0.309	0.532
Morpheus	255	0.804	0.542	0.780

C Morpheus Details

C.1 Source-only measurability

Let ζ collect the attack's internal randomness, including drop rates $\{b_i^{(k)}\}$ and retention vectors $\{\boldsymbol{r}_i^{(k)}\}$. Let $\sigma(\boldsymbol{\theta}_s, \boldsymbol{x}, y, \zeta)$ be the σ-algebra generated by the leaked checkpoint, the input, its label,

and this randomness. The adversary of Sec. 3 observes none of the future trajectory $\{\boldsymbol{\theta}_r : r > r_s\}$, the deployed $\boldsymbol{\theta}_t$, the client updates, the private federated data, or any output from the deployed model during attack construction.

Claim. The Morpheus perturbation $\boldsymbol{\delta}_K$ of Algorithm 1 is measurable with respect to $\sigma(\boldsymbol{\theta}_s, \boldsymbol{x}, y, \zeta)$.

Proof. We use induction on the joint state $(\boldsymbol{\delta}_k, \boldsymbol{q}_k)$ and show that it is $\sigma(\boldsymbol{\theta}_s, \boldsymbol{x}, y, \zeta)$-measurable at every step.

Base case. $(\boldsymbol{\delta}_0, \boldsymbol{q}_0) = (\mathbf{0}, \mathbf{0})$ is constant, hence measurable.

Inductive step. Assume $(\boldsymbol{\delta}_k, \boldsymbol{q}_k)$ is measurable. Each random draw $(b_i^{(k)}, \boldsymbol{r}_i^{(k)})$ builds the temporary subnetwork $\boldsymbol{\theta}_i^{(k)} = \mathcal{S}(\boldsymbol{\theta}_s; b_i^{(k)}, \boldsymbol{r}_i^{(k)})$ and forms $\boldsymbol{g}_i^{(k)} = \nabla_{\boldsymbol{\delta}} \mathcal{L}(F_{\boldsymbol{\theta}_i^{(k)}}(\boldsymbol{x} + \boldsymbol{\delta}_k), y)$. With the implementation's fixed subgradient convention, these are Borel-measurable functions of the measurable inputs, as is their average $\widehat{\boldsymbol{g}}_k$. Define $h(\boldsymbol{g}) = \boldsymbol{g}/\|\boldsymbol{g}\|_1$ for $\boldsymbol{g} \neq \mathbf{0}$ and $h(\mathbf{0}) = \mathbf{0}$. Both h and sign are Borel-measurable, while momentum addition and projection onto the input-budget box are continuous. Therefore $\boldsymbol{q}_{k+1} = \mu \boldsymbol{q}_k + h(\widehat{\boldsymbol{g}}_k)$ and $\boldsymbol{\delta}_{k+1} = \Pi_{\epsilon,\mathcal{X}}(\boldsymbol{\delta}_k + \alpha \operatorname{sign}(\boldsymbol{q}_{k+1}))$ are jointly measurable.

Conclusion. By induction $\boldsymbol{\delta}_K$ is $\sigma(\boldsymbol{\theta}_s, \boldsymbol{x}, y, \zeta)$-measurable. No step reads $\boldsymbol{\theta}_t$, a future round $\boldsymbol{\theta}_r$ with $r > r_s$, a client update, the private data, or deployed-model feedback, so no functional of the withheld trajectory enters the construction.

This formalizes the sense in which Morpheus is a source-side attack for fixed hyperparameters. Its only inputs are the leaked checkpoint, the input and label being attacked, and the attacker's own randomness. It requires no auxiliary or private training data.

C.2 Log-uniform drop-rate moments

Under $b \sim \log \mathcal{U}(b_{\mathrm{lo}}, b_{\mathrm{hi}})$ the log-rate $u = \log b$ is uniform on $[\log b_{\mathrm{lo}}, \log b_{\mathrm{hi}}]$, with constant density $p_u(u) = 1/\log(b_{\mathrm{hi}}/b_{\mathrm{lo}})$. Changing variables $b = e^u$, so $du/db = 1/b$, the density of b is

$$p(b) = p_u(\log b) \left| \tfrac{du}{db} \right| = \frac{1}{\log(b_{\mathrm{hi}}/b_{\mathrm{lo}})} \cdot \frac{1}{b}, \qquad b \in [b_{\mathrm{lo}}, b_{\mathrm{hi}}]. \tag{21}$$

For any $c > 1$ and $[a, ca] \subseteq [b_{\mathrm{lo}}, b_{\mathrm{hi}}]$,

$$\Pr(a \leq b \leq ca) = \int_a^{ca} p(b)\, db = \frac{\log c}{\log(b_{\mathrm{hi}}/b_{\mathrm{lo}})}, \tag{22}$$

which is independent of the interval location a. This is the precise sense in which the bounded law is scale-neutral: equal multiplicative drop-rate intervals receive equal probability. For any $p \neq 0$, the p-th moment is then a direct integral,

$$\mathbb{E}[b^p] = \int_{b_{\mathrm{lo}}}^{b_{\mathrm{hi}}} b^p\, p(b)\, db = \frac{1}{\log(b_{\mathrm{hi}}/b_{\mathrm{lo}})} \int_{b_{\mathrm{lo}}}^{b_{\mathrm{hi}}} b^{p-1}\, db \tag{23}$$

$$= \frac{1}{\log(b_{\mathrm{hi}}/b_{\mathrm{lo}})} \left[\tfrac{b^p}{p} \right]_{b_{\mathrm{lo}}}^{b_{\mathrm{hi}}} = \frac{b_{\mathrm{hi}}^p - b_{\mathrm{lo}}^p}{p\, \log(b_{\mathrm{hi}}/b_{\mathrm{lo}})}, \tag{24}$$

and at $p = 1$ this is $\mathbb{E}[b] = (b_{\mathrm{hi}} - b_{\mathrm{lo}})/\log(b_{\mathrm{hi}}/b_{\mathrm{lo}})$. Let $\boldsymbol{\phi}_s$ collect the source values of the eligible weights. A subnetwork displaces them by $\boldsymbol{\xi}_{b,\boldsymbol{r}} = \boldsymbol{\phi}_s \odot (\boldsymbol{r} - \mathbf{1})$, and each coordinate moves by $-\phi_{s,j}$ with probability b, so the per-rate energy is $\mathbb{E}[\|\boldsymbol{\xi}_{b,\boldsymbol{r}}\|_2^2 \mid b] = b\|\boldsymbol{\phi}_s\|_2^2$. The tower rule gives the marginal energy

$$\mathbb{E}\|\boldsymbol{\xi}_{b,\boldsymbol{r}}\|_2^2 = \mathbb{E}_b\big[b\|\boldsymbol{\phi}_s\|_2^2\big] = \mathbb{E}_b[b]\, \|\boldsymbol{\phi}_s\|_2^2 = \frac{b_{\mathrm{hi}} - b_{\mathrm{lo}}}{\log(b_{\mathrm{hi}}/b_{\mathrm{lo}})}\, \|\boldsymbol{\phi}_s\|_2^2. \tag{25}$$

Thus the connection-drop rate sets the normalized squared parameter displacement, and within $[b_{\mathrm{lo}}, b_{\mathrm{hi}}]$ the log-uniform law gives equal probability to equal multiplicative bands of the drop rate, and hence of the expected displacement energy.

C.3 Monte Carlo estimator: variance, concentration, and normalization order

Write one random subnetwork configuration as $\boldsymbol{\omega} = (b, \boldsymbol{r})$ with $\log b \sim \mathcal{U}(\log b_{\mathrm{lo}}, \log b_{\mathrm{hi}})$ and each r_e equal to 0 with probability b and to 1 otherwise, independently. Let $\boldsymbol{\theta}(\boldsymbol{\omega}) = \mathcal{S}(\boldsymbol{\theta}_s; b, \boldsymbol{r})$ and $\boldsymbol{h}(\boldsymbol{\omega}; \boldsymbol{\delta}) = \nabla_{\boldsymbol{\delta}} \mathcal{L}(F_{\boldsymbol{\theta}(\boldsymbol{\omega})}(\boldsymbol{x} + \boldsymbol{\delta}), y)$ be its input gradient. For the corresponding stochastic-subnetwork objective $J_{\mathrm{M}}(\boldsymbol{\delta}) = \mathbb{E}_{\boldsymbol{\omega}}[\mathcal{L}(F_{\boldsymbol{\theta}(\boldsymbol{\omega})}(\boldsymbol{x} + \boldsymbol{\delta}), y)]$, $\nabla_{\boldsymbol{\delta}} J_{\mathrm{M}}(\boldsymbol{\delta}) = \mathbb{E}_{\boldsymbol{\omega}}[\boldsymbol{h}(\boldsymbol{\omega}; \boldsymbol{\delta})]$ by dominated convergence when the input gradient exists almost everywhere and is bounded uniformly in $\boldsymbol{\omega}$ on the compact ℓ_∞ ball.

Fresh-subnetwork estimator. At each iteration, Algorithm 1 draws $\boldsymbol{\omega}_1^{(k)}, \ldots, \boldsymbol{\omega}_N^{(k)}$ independently from the joint drop-rate-and-retention distribution. Conditional on the current iterate $\boldsymbol{\delta}_k$, linearity and independence give

$$\mathbb{E}[\widehat{\boldsymbol{g}}_k \mid \boldsymbol{\delta}_k] = \nabla_{\boldsymbol{\delta}} J_{\mathrm{M}}(\boldsymbol{\delta}_k), \qquad \mathrm{Cov}[\widehat{\boldsymbol{g}}_k \mid \boldsymbol{\delta}_k] = \tfrac{1}{N} \mathrm{Cov}_{\boldsymbol{\omega}}[\boldsymbol{h}(\boldsymbol{\omega}; \boldsymbol{\delta}_k)], \tag{26}$$

so the raw-gradient estimation error decreases as $O(N^{-1/2})$.

Why normalization follows aggregation. The raw average in Eq. 6 estimates $\nabla_{\boldsymbol{\delta}} J_{\mathrm{M}}$. Normalizing each sampled gradient first would instead estimate a different quantity:

$$\mathbb{E}_{\boldsymbol{\omega}}\left[\frac{\boldsymbol{h}(\boldsymbol{\omega}; \boldsymbol{\delta})}{\|\boldsymbol{h}(\boldsymbol{\omega}; \boldsymbol{\delta})\|_1}\right] \neq \frac{\mathbb{E}_{\boldsymbol{\omega}}[\boldsymbol{h}(\boldsymbol{\omega}; \boldsymbol{\delta})]}{\|\mathbb{E}_{\boldsymbol{\omega}}[\boldsymbol{h}(\boldsymbol{\omega}; \boldsymbol{\delta})]\|_1} \quad \text{in general.}$$

Per-subnetwork normalization also assigns equal weight to gradients with different magnitudes. Morpheus therefore averages the raw gradients so their relative magnitudes are retained and subnetwork-specific components can cancel. Normalizing the aggregate afterward changes its scale but not its direction before momentum accumulation.

Loss concentration and the sign update. Let $Z_i = \mathcal{L}(F_{\boldsymbol{\theta}(\boldsymbol{\omega}_i)}(\boldsymbol{x} + \boldsymbol{\delta}), y) \in [a_0, a_1]$ be independent subnetwork losses and $\hat{J}_N = \frac{1}{N} \sum_i Z_i$. Hoeffding's inequality gives, for any $t > 0$,

$$\Pr\left(|\hat{J}_N - J_{\mathrm{M}}(\boldsymbol{\delta})| \geq t\right) \leq 2 \exp\Big(- \tfrac{2Nt^2}{(a_1 - a_0)^2} \Big).$$

This bound concerns the Monte Carlo error of the loss estimate at a fixed $\boldsymbol{\delta}$. Unbiasedness holds for the raw Monte Carlo gradient $\widehat{\boldsymbol{g}}_k$. The subsequent ℓ_1 normalization, momentum accumulation, and sign operation make the applied update direction a biased, bounded function of $\widehat{\boldsymbol{g}}_k$, as in momentum sign-ascent attacks generally.

C.4 Morpheus optimization details

The optimizer used with Eq. 6 is standard momentum sign-ascent. Given the Monte Carlo gradient estimate $\widehat{\boldsymbol{g}}_k$, for $k = 0, \ldots, K-1$ the accumulator and perturbation are updated as

$$\boldsymbol{q}_{k+1} = \mu\, \boldsymbol{q}_k + \frac{\widehat{\boldsymbol{g}}_k}{\|\widehat{\boldsymbol{g}}_k\|_1}, \qquad \boldsymbol{\delta}_{k+1} = \Pi_{\epsilon, \mathcal{X}}\big(\boldsymbol{\delta}_k + \alpha\, \mathrm{sign}(\boldsymbol{q}_{k+1})\big), \tag{27}$$

where $\Pi_{\epsilon, \mathcal{X}}$ projects onto the intersection of $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$ and the valid input range $\boldsymbol{x} + \boldsymbol{\delta} \in \mathcal{X}$, and we adopt the convention $\widehat{\boldsymbol{g}}_k / \|\widehat{\boldsymbol{g}}_k\|_1 := \mathbf{0}$ when $\widehat{\boldsymbol{g}}_k = \mathbf{0}$ so the update is defined at a zero gradient and each iterate stays a measurable function of the attack randomness (App. C.1). With one subnetwork and $b = 0$ (a degenerate case outside the sampled range), Eq. 6 reduces to the source-model gradient, giving the momentum sign-ascent control used in the evaluation.

Across all pairs, $\mathcal{E}_w$ indexes every scalar weight of every connection-bearing weight tensor of the architecture. Normalization affine parameters, biases, and running statistics are never included.

D Experimental Details

D.1 Dataset and architecture pairs

Table 3 gives the exact FL trajectory and perturbation budget used for each of the 12 pairs. The datasets and architectures follow representative federated and adversarial-learning evaluations in

Algorithm 1: Morpheus with multi-rate stochastic subnetworks

input : surrogate $\boldsymbol{\theta}_s$; eligible connection set $\mathcal{E}_w$ excluding normalization and biases; input $\boldsymbol{x}$, label y; input budget ϵ, step α, iterations K, momentum μ; sampled subnetworks N, connection-drop-rate range $[b_{\mathrm{lo}}, b_{\mathrm{hi}}]$

output: adversarial perturbation $\boldsymbol{\delta}$ with $\|\boldsymbol{\delta}\|_\infty \le \epsilon$

$\boldsymbol{\delta} \leftarrow \mathbf{0}$; $\quad \boldsymbol{q} \leftarrow \mathbf{0}$

for $k \leftarrow 1$ **to** K **do**

  for $i \leftarrow 1$ **to** N **do**

    draw $\log b_i^{(k)} \sim \mathcal{U}(\log b_{\mathrm{lo}}, \log b_{\mathrm{hi}})$

    for each $e \in \mathcal{E}_w$, set $r_{i,e}^{(k)} = 0$ with probability $b_i^{(k)}$ and $r_{i,e}^{(k)} = 1$ otherwise

    form $\boldsymbol{\theta}_i^{(k)} \leftarrow \mathcal{S}(\boldsymbol{\theta}_s; b_i^{(k)}, \boldsymbol{r}_i^{(k)})$

    $\boldsymbol{g}^{(i)} \leftarrow \nabla_{\boldsymbol{\delta}} \mathcal{L}\big(F_{\boldsymbol{\theta}_i^{(k)}}(\boldsymbol{x}+\boldsymbol{\delta}), y\big)$

  $\widehat{\boldsymbol{g}} \leftarrow \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{g}^{(i)}$ `// raw subnetwork-gradient mean`

  $\boldsymbol{q} \leftarrow \mu\,\boldsymbol{q} + \widehat{\boldsymbol{g}}/\|\widehat{\boldsymbol{g}}\|_1$ `// momentum`

  $\boldsymbol{\delta} \leftarrow \Pi_{\epsilon,\mathcal{X}}\big(\boldsymbol{\delta} + \alpha\,\mathrm{sign}(\boldsymbol{q})\big)$ `// ℓ∞ sign step and projection`

return $\boldsymbol{\delta}$

Table 3: Training recipe, deployment checkpoint, and perturbation budget of each pair. R is the training horizon, r_t the fixed deployment checkpoint, B the client batch size, η the client learning rate, and λ weight decay. P10 denotes the best-validation checkpoint under early stopping with patience 10, whereas Best denotes the best-validation checkpoint within $R = 200$. The perturbation budget ϵ is an ℓ_∞ radius in the corresponding normalized input domain.

Modality	Dataset / architecture	R	r_t	Rule	B	η	λ	ϵ
Image	RAF-DB / ResNet18	500	104	P10	32	0.01	5×10^{-4}	$8/255$
Image	RAF-DB / VGG16	500	148	P10	32	0.05	5×10^{-4}	$8/255$
Image	TinyImageNet / ResNet18	500	117	P10	32	0.01	5×10^{-4}	$8/255$
Image	TinyImageNet / VGG16	500	224	P10	32	0.01	5×10^{-4}	$8/255$
Audio	CREMA-D / BC-ResNet8	200	190	Best	32	0.01	5×10^{-4}	0.005
Audio	CREMA-D / CBAM-DenseNet121	200	199	Best	32	0.01	5×10^{-4}	0.005
Audio	UrbanSound8K / BC-ResNet8	200	186	Best	32	0.10	5×10^{-4}	0.005
Audio	UrbanSound8K / CBAM-DenseNet121	200	195	Best	32	0.01	5×10^{-4}	0.005
Point cloud	ModelNet40 / PointNet	300	95	P10	32	0.05	1×10^{-4}	0.06
Point cloud	ModelNet40 / PointMLP	300	147	P10	32	0.10	1×10^{-4}	0.17
Point cloud	ShapeNetPart / PointNet	300	68	P10	32	0.05	1×10^{-4}	0.10
Point cloud	ShapeNetPart / PointMLP	300	150	P10	32	0.10	1×10^{-4}	0.20

vision (Gao et al., 2022; Li et al., 2021), audio (Zhang et al., 2023a; Pillutla et al., 2022; Kim et al., 2023a; Chen et al., 2023), and point clouds (Xue et al., 2021; Khan et al., 2024; Hamdi et al., 2020; Hu et al., 2022).

Evaluation metric. For evaluation set $\mathcal{D}_{\mathrm{eval}}$, let $\mathcal{D}_c = \{(\boldsymbol{x}, y) \in \mathcal{D}_{\mathrm{eval}} : F_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = y\}$ be the inputs correctly classified by the deployed model, and let $\boldsymbol{\delta}_{\boldsymbol{x},y} = \mathcal{A}(\boldsymbol{\theta}_s, \boldsymbol{x}, y)$ denote the perturbation generated for input $\boldsymbol{x}$ with label y. The untargeted transfer attack success rate is $\mathrm{ASR}_{\mathrm{transfer}} = |\mathcal{D}_c|^{-1} \sum_{(\boldsymbol{x},y)\in\mathcal{D}_c} \mathbf{1}[F_{\boldsymbol{\theta}_t}(\boldsymbol{x} + \boldsymbol{\delta}_{\boldsymbol{x},y}) \neq y]$.

FL training. All pairs use FedAvg with $M = 100$ clients, $C = 10$ clients per round, $E = 5$ local epochs, IID client partitions, SGD, a constant learning rate, and one saved global checkpoint per round. For a selected target r_t, the source checkpoints are $r_s = \mathrm{round}(q r_t)$ for $q \in \{0, 0.05, 0.10, \ldots, 1.00\}$. For image and point-cloud pairs, training terminates when validation accuracy shows no improvement for 10 consecutive rounds (P10), and the best-validation checkpoint is deployed. Audio trajectories are trained for $R = 200$ rounds, and the best-validation checkpoint is deployed. Target selection is completed before any transfer attack is evaluated.

FL recipes. The table reports the training hyperparameters that differ across the 12 trajectories and the perturbation budget of each pair. No target-model information is used to tune an attack. Within each pair, all methods use the same source and target checkpoints, clean evaluation subset, and perturbation budget.

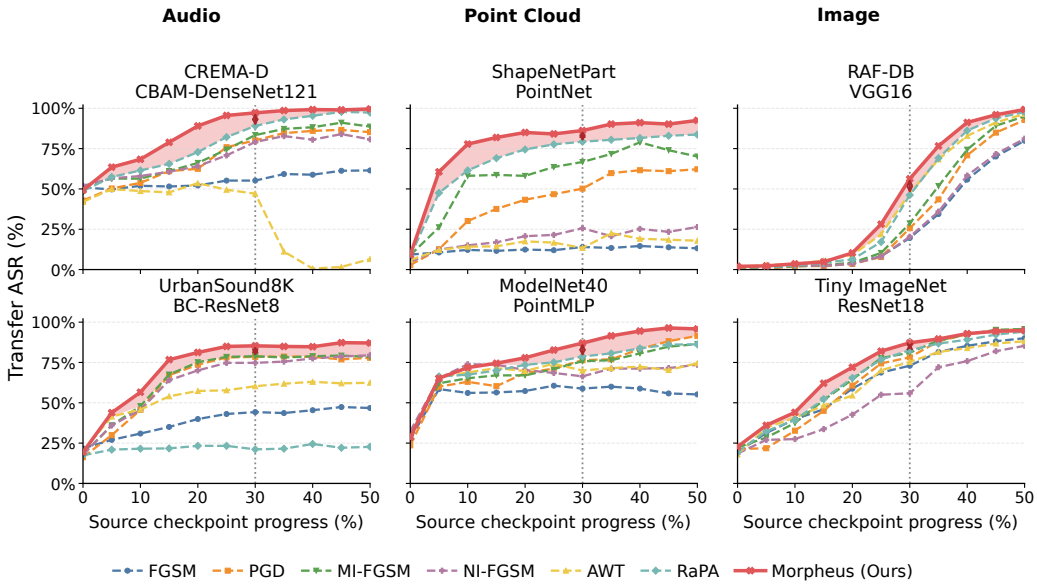


Figure 13: **Morpheus achieves the strongest temporal transferability at the 30% leakage stage across the complementary six pairs.** Each column contains the two pairs from one modality not shown in Fig. 6, and the dotted line marks the leakage stage used for the primary comparison. The source axis ends at 50%, while every attack is evaluated against its pair's fixed 100%-progress deployed target.

Table 4: **Morpheus achieves the highest transfer ASR at the 30% leakage stage on all 12 pairs.** Values are percentages computed using each pair's 30% source checkpoint and deployed target. The best result in each row is bold.

Dataset / architecture	FGSM	PGD	MI-FGSM	NI-FGSM	AWT	RaPA	Morpheus
Image							
RAF-DB / ResNet18	50.00	50.54	55.91	43.82	36.29	70.16	**82.80**
RAF-DB / VGG16	19.60	25.63	28.64	20.35	47.24	46.23	**56.28**
TinyImageNet / ResNet18	72.99	78.20	81.04	55.92	75.36	82.62	**87.20**
TinyImageNet / VGG16	56.28	53.25	63.20	53.68	67.53	76.91	**85.71**
Audio							
CREMA-D / BC-ResNet8	33.72	79.14	81.09	83.43	75.63	41.78	**91.23**
CREMA-D / CBAM-DenseNet121	55.17	80.12	83.37	79.11	47.06	89.05	**97.16**
UrbanSound8K / BC-ResNet8	44.23	78.47	78.86	74.76	60.27	21.14	**85.32**
UrbanSound8K / CBAM-DenseNet121	40.58	65.08	67.69	60.34	24.04	74.27	**91.12**
Point cloud							
ModelNet40 / PointNet	26.10	61.80	77.50	36.20	12.10	75.93	**87.90**
ModelNet40 / PointMLP	58.80	76.40	75.80	66.40	70.00	78.70	**87.00**
ShapeNetPart / PointNet	14.00	50.20	66.70	25.60	13.40	79.27	**86.20**
ShapeNetPart / PointMLP	28.80	59.00	51.80	32.40	28.60	57.73	**79.00**

D.2 Temporal-transfer curves of the remaining six pairs

Fig. 13 presents the six pairs complementary to Fig. 6. Together, the two figures visualize every pair used in the main evaluation. Curves display source progress through 50%, where temporal drift meaningfully separates the attacks.

D.3 Main-table numbers

Table 4 lists transfer ASR at the 30% leakage stage for every pair and method, the pair-level numbers summarized in Sec. 5.1. RaPA is run without its image-specific transformations on all 12 pairs and reported as the mean over its three pruning ratios (App. D.4).

Table 5: **The iterative gradient attacks, RaPA, and Morpheus share the step schedule of each row.** Point-cloud perturbation budgets are provided in Table 3, and N_{M} denotes the number of Morpheus subnetworks. Exact method-specific settings follow.

Modality	ϵ	K	α	N_{M}	$[b_{\mathrm{lo}}, b_{\mathrm{hi}}]$
Image	$8/255$	20	$2/255$	32	$[0.005, 0.05]$
CREMA-D / BC-ResNet8	0.005	20	$\epsilon/20$	32	$[0.005, 0.05]$
Other audio settings	0.005	20	$\epsilon/5$	32	$[0.005, 0.05]$
Point cloud	pair-specific	30	$2\epsilon/30$	32	$[0.005, 0.05]$

D.4 Attack settings

For each pair, all attacks use the same ϵ, source and target checkpoints, and evaluation subset. Image and point-cloud pairs are evaluated on 512 sampled test inputs, with a fixed seed per modality. Audio pairs are evaluated on the complete test split, 901 examples for CREMA-D and 837 for UrbanSound8K. Transfer ASR is computed only on examples the target checkpoint classifies correctly. Writing $\mathcal{X}_c = \{(\boldsymbol{x}, y) : F_{\boldsymbol{\theta}_t}(\boldsymbol{x}) = y\}$ for this set,

$$\text{Transfer ASR} = \frac{1}{|\mathcal{X}_c|} \sum_{(\boldsymbol{x},y)\in\mathcal{X}_c} \mathbf{1}\big[F_{\boldsymbol{\theta}_t}(\boldsymbol{x}+\boldsymbol{\delta}) \neq y\big]. \tag{28}$$

Table 5 gives the (K, α) schedule shared by the iterative gradient attacks, RaPA, and Morpheus, together with the Morpheus subnetwork settings. FGSM uses one step of size ϵ. AWT uses the separate schedule reported below.

FGSM and PGD. FGSM takes a single sign-gradient step of radius ϵ. PGD applies projected sign-gradient updates for the modality-specific (K, α) in Table 5. Projection is onto the ℓ_∞ ball and the valid input domain for images and audio, and onto the ℓ_∞ coordinate-displacement ball for point clouds.

MI-FGSM and NI-FGSM. Both attacks use the same (K, α) as PGD and momentum coefficient $\mu = 1$. MI-FGSM accumulates the per-example normalized gradient before the sign update. NI-FGSM evaluates that gradient at the Nesterov look-ahead point before applying the same momentum and projection rule. This follows the standard momentum-style implementations (Dong et al., 2018; Lin et al., 2020) without modality-specific input transformations.

AWT. We port the official AWT objective, including its self-distillation, look-ahead gradient averaging, and SAM-based surrogate update (Chen et al., 2025). The original hyperparameters target ImageNet-scale models, so we calibrate them once per modality for stable optimization on the source model alone, without consulting either the target checkpoint or its predictions. Across modalities, AWT uses 10 attack steps, 20 neighbors, $\alpha = \epsilon/10$, $\beta = 3$, and $\gamma = 0.5$. The SAM learning rate/radius $(\eta_{\mathrm{SAM}}, \rho_{\mathrm{SAM}})$ are $(10^{-3}, 5 \times 10^{-3})$ for image, $(6 \times 10^{-4}, 1.5 \times 10^{-3})$ for audio, and $(2 \times 10^{-3}, 5 \times 10^{-3})$ for point clouds. These values are fixed for every pair of a modality.

RaPA. RaPA is sensitive to its pruning ratio. Selecting one ratio on the final model would leak target information and overstate transfer. We instead use the interval endpoints and geometric midpoint:

$$\mathcal{B}_{\mathrm{RaPA}} = \{0.005, 0.01581, 0.05\}. \tag{29}$$

This set is fixed across every dataset, architecture, modality, and leakage stage. We run a complete curve for each ratio and report their arithmetic mean, so no result selects the best-performing ratio on the deployed model. Each gradient estimate averages five stochastically pruned surrogates. We keep RaPA's random pruning within its official scope of normalization and linear layers, together with its momentum, and disable its image-specific input transformations, because they are undefined for audio and point clouds and one configuration must serve every modality. The attack uses the (K, α) schedule in Table 5.

Morpheus. Morpheus uses $N = 32$ stochastic subnetworks per iteration, momentum coefficient $\mu = 1$, and the same modality-specific (K, α) schedule as the iterative baselines. Each stochastic subnetwork draws its weight-drop rate from the log-uniform interval $[0.005, 0.05]$ and a fresh

retention vector. The same interval is used unchanged across every dataset, architecture, and modality, and Sec. 5.4 shows that transfer is insensitive to its width and location. All non-normalization connection-bearing weights are eligible, while biases, normalization parameters, and running statistics remain unchanged. Algorithm 1 gives the complete update.

D.5 FL SCENARIO SETTINGS

All FL scenario sweeps use RAF-DB/ResNet18 and a fixed 512-image evaluation subset. Data heterogeneity is displayed as non-IIDness $1 - \alpha \in \{0, 0.1, 0.3, 0.5, 0.7, 0.9\}$: the first point is the IID anchor, and the remaining points use Dirichlet concentration $\alpha \in \{0.9, 0.7, 0.5, 0.3, 0.1\}$. The other sweeps compare FedAvg with FedSGD, vary participating clients over $\{10, 20, 50, 70, 100\}$, and vary the client population over $\{10, 20, 50, 100\}$. The participation sweep fixes 100 total clients. The population sweep fixes 10 participants per round. RaPA is averaged over its three fixed pruning ratios. Because the trajectories converge at different rates, each configuration has its own source and deployment rounds. Source progress ranges from 15% to 30% in the numerical sweeps and is 50% for FedSGD. The shared IID FedAvg, 10-participant, and 100-client anchor uses the RAF-DB/ResNet18 round-31 to round-104 checkpoint pair. All remaining settings follow the default configuration.

D.6 ABLATION SETTINGS

Table 1 and Fig. 11 use RAF-DB/ResNet18, source round 31, target round 104, 512 random test inputs, and the image settings in Table 5. The target correctly classifies 372 of these inputs, which form the common conditional-ASR denominator. All stochastic subnetworks mask Conv2d and Linear weights but exclude normalization and bias terms. The no-subnetwork-sampling control uses one unperturbed model ($N = 1, b = 0$), and the other component controls use $N = 32$. Fixed drop rates are $\{0.005, 0.01581, 0.05\}$, where 0.01581 is the geometric midpoint. These are the same three values as RaPA's pruning ratios. At $N = 4$ and $K = 20$, Morpheus uses 80 gradient evaluations per AE, fewer than the 100 that RaPA uses at each pruning ratio, and already reaches 79.30% transfer ASR against RaPA's 70.16%. The component table reports their arithmetic mean rather than selecting one fixed rate. The linear- and log-uniform controls share support $[0.005, 0.05]$ and draw fresh per-subnetwork rates at each attack iteration. Every stochastic forward pass draws a fresh retention mask. The subnetwork-count sweep uses the full log-uniform method for $N \in \{4, 8, 16, 32, 64\}$. Repeated runs of the default configuration on these inputs transfer 306 or 308 of the 372 target-clean AEs (82.26% or 82.80%), so differences of one or two inputs are not interpreted. Table 1 and Fig. 11 take the Full and $N = 32$ entries from the main-protocol run of Table 4 and the other entries from an ablation batch run with the same settings. Fig. 12 uses three attack seeds, and each seed also draws its own 512 inputs, of which 372, 349, and 356 are classified correctly by the target. Its single-rate settings are log-uniform ranges $[0.0049, 0.0051]$, $[0.015, 0.017]$, and $[0.049, 0.051]$.

D.7 DEFENSE SETTINGS

The defense evaluation uses 512 AEs generated from the leaked round-31 model. The inference-time defenses use the original round-104 target, while FedSlice uses its own trained target. Clean accuracy is measured on all 3,068 test images. Transfer ASR is conditioned on inputs that each defended target classifies correctly before attack. We choose the inference-time-defense parameters using clean accuracy alone, targeting a clean-accuracy drop of about 5 points relative to the undefended round-104 target. Transfer ASR is never consulted during selection. All three inference-time defenses use noise standard deviation $\sigma = 0.06$. Gaussian logit smoothing averages logits over 16 samples, input randomization pools 16 independent single-query trials without averaging their outputs, and RS uses 1,000 hard votes with significance level 0.001. Their full-test clean accuracies are 67.37%, 66.82%, and 66.43%, respectively, corresponding to drops of 4.69, 5.24, and 5.64 points from the 72.07% undefended target. FedSlice is not part of this inference-time calibration: it changes the FL protocol and uses independent branch initialization, a 5% public slice, three server epochs, and the best checkpoint within 100 rounds.