FEDERATED LEARNING OF LARGE NOVEL VIEW SYNTHESIS MODEL FOR BETTER PERSONALIZED NEURAL FIELD

Anonymous authors
Paper under double-blind review

ABSTRACT

Large image-conditioned novel view synthesis (NVS) models benefit from training across diverse scenes, but collecting multi-view images from many owners on a central server can be impractical due to privacy and data-ownership constraints. Keeping each scene local avoids raw-data sharing, yet personalized neural fields trained from sparse client views often suffer from insufficient geometric supervision. Moreover, sharing such neural fields is itself undesirable because their weights can constitute a directly renderable representation of the private scene. We propose a federated framework that couples a server-side large NVS model with personalized client neural fields that remain entirely local. Each client uses its neural field to generate nearby pseudo-target views and locally refine only a trainable subset of the server model using both captured and rendered supervision. Only the resulting parameter update is communicated, while raw images, pseudo-targets, and neural-field parameters remain local. The aggregated global model is then returned to clients to improve their neural fields through consistency supervision, creating a mutual-reinforcement loop between global generalization and local personalization. Experiments with 5,000 scene-level clients show substantial improvements in personalized client neural fields, while the server model achieves competitive generalized NVS performance under zero-shot, cross-domain, and non-IID evaluations. Empirical MIA/PIA and limited-context leakage evaluations further indicate lower privacy exposure than directly sharing client neural fields, without claiming formal differential privacy.

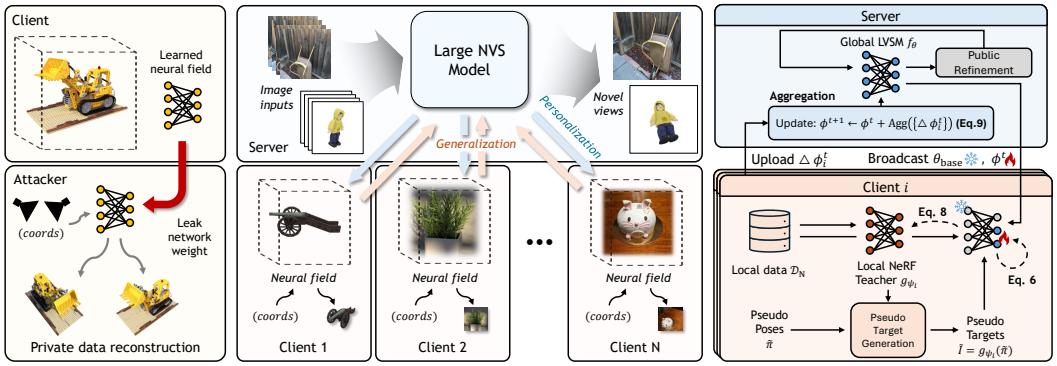
1 INTRODUCTION

Recent advances in novel view synthesis (NVS) have expanded from optimizing a separate representation for each scene (Mildenhall et al., 2020; Barron et al., 2021) to image-conditioned models trained across many scenes (Yu et al., 2021; Wang et al., 2021; Hong et al., 2024; Jin et al., 2025). A single model serves different scenes given posed input images, but training typically requires centrally collected multi-view data. These scenes may belong to different users or organizations and contain private content, making centralized collection impractical due to privacy and data-ownership constraints.

Keeping each scene and its training local avoids centralized collection, but introduces a different challenge. Each scene owner acts as a client that optimizes a personalized neural field using only its own posed observations. Sparse local views can make this optimization underconstrained and degrade reconstruction in poorly observed regions. A large image-conditioned NVS model trained across many scenes may provide useful guidance for client personalization, while private client scenes can provide scene-specific supervision for improving the server model. The server can therefore assist client reconstruction, while client models provide supervision back to the server.

Federated learning (FL) provides a natural mechanism for this interaction by aggregating locally computed updates without centrally collecting raw client data (McMahan et al., 2017; Kairouz et al., 2021). Neural fields, however, introduce an additional privacy concern (Fig. 1(a)). A trained NeRF is itself a directly renderable representation of its private scene, so exposing its parameters may allow novel views to be queried without the original images (Zhang et al., 2024a; Yun et al., 2025). We

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.



(a) Private data leak scenario (b) Generalization-personalization pipeline (c) Methodology overview

Figure 1: **Overview of our pipeline.** (a) Sharing a trained neural field exposes a directly renderable private scene, even without the original images. (b) Our framework jointly trains a generalized large NVS model and personalized per-scene neural fields through mutual reinforcement. (c) To train the large NVS model from scratch, each client fits a private teacher and renders nearby pseudo-target views for parameter-efficient local server updates. Captured views anchor each update, while pseudo targets supervise nearby camera poses beyond the client's sparse observations. The server aggregates only these updates; images and neural-field parameters remain local.

therefore use client neural fields as local teachers without communicating them. The server maintains a large image-conditioned NVS model (Jin et al., 2025), while each client fits a personalized neural field (Mildenhall et al., 2020) from its private images. At each round, clients receive the trainable server parameters and update them using captured images and nearby pseudo targets rendered by their fields. Only parameter updates are uploaded; images, pseudo targets, and field parameters remain local. The server aggregates these updates for the next broadcast, while each client uses the received server snapshot to refine its field through consistency supervision (Fig. 1(c)).

As illustrated in Fig. 1(b), this interaction forms a *virtuous cycle of mutual reinforcement*. (1) Generalization $\rightarrow$ Personalization: the server model provides additional supervision for client neural field optimization through its predictions at nearby views, which is especially useful when local observations are sparse. (2) Personalization $\rightarrow$ Generalization: the personalized neural field serves as a scene-specific teacher whose nearby pseudo targets expand the supervision available for local server updates. Improved client fields provide more accurate pseudo targets, while the improved server supplies stronger consistency supervision, coupling generalization and personalization rather than optimizing them independently.

The server requires posed input images to synthesize a client scene, unlike a directly renderable neural field. Keeping the field local avoids this direct leakage channel; secure aggregation and differential privacy offer additional protection for updates (Bonawitz et al., 2017; Dwork, 2006). When training the large NVS model from scratch, a fixed public dataset additionally serves as an optimization anchor alongside the federated client supervision.

Experiments with 5,000 scene-level clients show substantial improvements in sparse-view personalization and competitive server performance under zero-shot, cross-domain, and non-IID evaluations. We assess privacy through inference attacks and reconstruction from intercepted updates and leaked images. To the best of our knowledge, this is the first work to jointly train a large image-conditioned NVS model and distributed per-scene neural fields through an explicit federated mutual-reinforcement process without directly exchanging renderable client-scene representations.

2 Related Work

2.1 Large NVS Models and Neural Rendering

Novel view synthesis aims to generate photorealistic views of 3D scenes from posed input images. Coordinate-based neural fields, most notably NeRF (Mildenhall et al., 2020), represent scenes through continuous density and radiance functions. Subsequent work improves large-scene scalability (Tancik et al., 2022), sampling (Lee et al., 2024), and encoding (Müller et al., 2022), or uses explicit Gaussian representations for efficient rendering (Kerbl et al., 2023).

Large object collections such as Objaverse (Deitke et al., 2023b) and Objaverse-XL (Deitke et al., 2023a) support cross-scene NVS training. Generative methods also produce multi-view observations from a single image, including progressive conditioning on nearby predictions (Chen et al., 2024). These approaches typically assume centralized scene data and substantial compute; our framework instead uses private supervision computed on clients.

2.2 Federated Learning of Large Models

Federated learning of foundation models often freezes the backbone and trains low-rank adapters (Wang et al., 2024) or prompts (Guo et al., 2023; Su et al., 2024) to reduce client training and communication costs. FLoRA aggregates heterogeneous low-rank adaptations (Wang et al., 2024), while system-level work targets bandwidth-limited federated pretraining (Sani et al., 2025). Federated instruction tuning likewise aggregates local instruction-following updates without collecting private instructions centrally (Zhang et al., 2024b). For vision-language models, federated prompt learning adapts shared representations to client domains without full-model fine-tuning (Guo et al., 2023; Su et al., 2024; Qiu et al., 2024; Pan et al., 2024). These methods primarily address language or visual understanding. We instead couple large NVS training with scene-specific neural-field personalization.

2.3 Neural Radiance Fields in Federated Learning

Applying FL to neural rendering is subtle because scene representations may expose private visual content, motivating privacy-aware protocol design (Yun et al., 2025). Existing federated neural rendering methods mainly reconstruct a shared scene or environment from observations distributed across clients. Federated Neural Radiance Fields (Holden et al., 2023) and FedNeRF (Zhang & Shao, 2024) aggregate scene-specific NeRF updates toward a common radiance field, Fed3DGS (Suzuki, 2024) scales this setting with 3D Gaussians, and DecentNeRFs (Tasneem et al., 2024) separates personal and global radiance fields for crowdsourced reconstruction.

Our approach differs in what is federated and how cross-client knowledge is transferred. Rather than treating neural fields as the global model to be directly averaged or shared, we use them as private scene teachers for a separate image-conditioned NVS model. They convert local 3D information into supervision for federated optimization, addressing the structural mismatch between per-scene coordinate-based reconstruction and cross-scene image-conditioned generalization while keeping raw client images local.

3 Method

Our framework couples a server-side large NVS model with private client neural fields. Clients supervise local server updates, and the shared server guides personalization; only updates to the trainable server subset are uploaded. Algorithm 1 in Appendix A gives the round-level procedure.

3.1 Federated Mutual-Reinforcement Pipeline

Client i owns posed images

$$\mathcal{D}_i = \{(I_{i,v}, \pi_{i,v})\}_{v=1}^{V_i}, \qquad \pi_{i,v} = (K_{i,v}, T_{i,v}), \tag{1}$$

where $K_{i,v}$ and $T_{i,v}$ are the camera intrinsics and extrinsics, and $V_i = 12$ in our experiments. The server receives M posed input views and a target camera, and predicts

$$\hat{I}_{\text{tgt}} = f_\theta\big(\{(I_m, \pi_m)\}_{m=1}^{M}, \pi_{\text{tgt}}\big)\,. \tag{2}$$

We write $\theta = (\theta_{\text{base}}, \phi)$, where θ_{base} contains the image encoder and ϕ the target-pose conditioning and decoding modules. Clients run the full model locally but optimize only ϕ, keeping the encoder fixed during local updates. This parameter-efficient split makes client-side optimization practical without changing the NVS architecture. At communication round t, clients use the cached encoder and receive ϕ^t. Appendix A.2 specifies the trainable modules.

Step 1: local neural field teacher. Each selected client first fits a private neural field g_{ψ_i} from its local posed images. We use TensoRF with an InfoNeRF-style entropy regularizer,

$$\min_{\psi_i}\ \mathbb{E}_{(I,\pi)\sim\mathcal{D}_i,\ r\sim\text{Rays}(I,\pi)}\ell_{\text{pix}}(g_{\psi_i}(r), c(r)) + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}}(\psi_i). \tag{3}$$

The field serves as a local teacher and is never uploaded, avoiding direct exposure of the private scene. Appendix A.1 gives its architecture and fitting settings.

Step 2: pseudo-target generation. After fitting g_{ψ_i}, the client samples nearby camera poses by bounded SE(3) perturbations of captured poses. For an input-view pack $\mathcal{X}_i \subset \mathcal{D}_i$, let P_i^{cap} be the captured-target distribution and A_i the anchor distribution used for perturbations; Q_i^r is the resulting nearby-pose distribution. The client renders

$$\tilde{I}_i(\tilde{\pi}) = g_{\psi_i}(\tilde{\pi}), \qquad \bar{Q}_i^{\gamma} = (1-\gamma)P_i^{\text{cap}} + \gamma Q_i^r. \tag{4}$$

We select the pseudo-target probability $\gamma = 0.4$ empirically and keep it fixed across clients and rounds, except in the budget ablation. The sampled branch determines whether the target is a captured image or a teacher prediction:

$$Y_i^{\gamma}(\pi) = \begin{cases} I_i(\pi), & \pi \sim P_i^{\text{cap}}, \\ \tilde{I}_i(\pi), & \pi \sim Q_i^r. \end{cases} \tag{5}$$

Unlike independent 2D augmentations, pseudo targets share the field's geometry, visibility, and appearance. Nearby views expose parallax, silhouette motion, and small disocclusions while staying within the teacher's reliable neighborhood. Appendix B.8 analyzes their cross-view consistency.

Step 3: local refinement of the server model. Client i receives ϕ^t, samples an input-view pack $\mathcal{X}_i \subset \mathcal{D}_i$ and a target from $\bar{Q}_i^{\gamma}$, and updates ϕ by minimizing

$$\mathcal{L}_{\text{srv}}^{(i)}(\phi) = \mathbb{E}_{\mathcal{X}_i, \pi \sim \bar{Q}_i^{\gamma}} \ell(f_{\theta_{\text{base}},\phi}(\mathcal{X}_i, \pi), Y_i^{\gamma}(\pi)) \,. \tag{6}$$

The same reconstruction loss is used for real and fixed teacher targets. The client uploads only

$$\Delta\phi_i^t = \phi_i^t - \phi^t. \tag{7}$$

Raw images, pseudo-target images, and ψ_i remain local. Each client updates a local copy of the server model using its private images as context and its neural field as the source of additional targets. This lets the two models exchange scene information despite their different representations: the neural field maps coordinates to scene appearance, whereas the server predicts an image from posed image context. The reconstruction loss connects these parameterizations, allowing the client field to supervise the server without sharing its parameters. The uploaded delta contains only changes to the server's trainable subset. In the reverse direction, server predictions supervise the client field through the consistency loss. The two updates therefore use the same private scene while retaining separate parameters and learning objectives.

Step 4: client refinement from the server model. The broadcast server model also guides client personalization. Each client retains ϕ^t as a fixed snapshot, separate from the locally updated copy ϕ_i^t used for upload. On nearby poses, it minimizes

$$\mathcal{L}_{\text{cons}}^{(i)}(\phi^t, \psi_i) = \mathbb{E}_{\pi \sim Q_i^r} \ell(g_{\psi_i}(\pi), \text{sg}[f_{\theta_{\text{base}},\phi^t}(\mathcal{X}_i, \pi)]) \,. \tag{8}$$

Here sg stops the gradient. Only ψ_i is updated; local server training uses $\mathcal{L}_{\text{srv}}^{(i)}$ in Eq. (6). These predictions supplement sparse observations; the refined field supplies better pseudo targets in later rounds. Appendix A.3 gives the combined client objective.

Step 5: aggregation and public anchor. Let $\mathcal{S}_t$ denote the set of clients participating in round t. Our implementation uses weighted update averaging:

$$\bar{\phi}^{t+1} = \phi^t + \eta_{\text{agg}} \sum_{i \in \mathcal{S}_t} \alpha_i \Delta\phi_i^t, \qquad \alpha_i = \frac{w_i}{\sum_{j \in \mathcal{S}_t} w_j}. \tag{9}$$

We use uniform weights, $w_i = 1$ and $\alpha_i = 1/|\mathcal{S}_t|$. Our contribution is the neural-field-to-NVS supervision interface, which also supports other FL optimizers. Starting from $\bar{\phi}^{t+1}$, the server refines the model on public scenes when available and obtains ϕ^{t+1} for the next broadcast. The public/client split determines

$$\rho = \frac{n_0}{n_0 + n_c^{\text{eff}}}. \tag{10}$$

Here $n_0 = |\mathcal{D}_{\text{pub}}|$ and $n_c^{\text{eff}} = N$ count public and private scenes, giving $\rho = 75/80$ in the main setting. This weight is fixed by the scene allocation, not tuned per round; it defines the risk mixture below rather than a ratio of optimizer steps. Without public data, training proceeds through client updates. Appendix A gives the implementation.

3.2 Server Generalization and Pseudo-Target Supervision

Pseudo targets supervise the server's response to camera motion. Let $H_\phi(\delta) = f_{\theta_{\text{base}},\phi}(\mathcal{X}_i, \pi \oplus \delta)$ and $T_\psi(\delta) = g_{\psi_i}(\pi \oplus \delta)$ be the server and teacher images. For zero-mean perturbations, under the smoothness assumptions in Appendix B.6,

$$\mathbb{E}_\delta \frac{1}{2}\|H_\phi(\delta) - T_\psi(\delta)\|^2 = \frac{1}{2}\|H_\phi(0) - T_\psi(0)\|^2 + \frac{1}{2}\|J_\pi H_\phi - J_\pi T_\psi\|^2_{\Sigma_\delta} + \mathcal{R}_{\text{curv}} + \mathcal{O}(\mathbb{E}\|\delta\|^3). \tag{11}$$

Here $\Sigma_\delta = \mathbb{E}[\delta\delta^\top]$ and $\mathcal{R}_{\text{curv}}$ is the residual-weighted curvature term defined in Appendix B.6. The first term matches the anchor image; the second matches its local pose derivative. Thus, nearby pseudo targets constrain how the image changes with camera motion, including parallax, silhouette displacement, and visibility-related changes. They transfer the client field's local 3D response to the server's target-pose conditioning.

Their benefit is balanced by teacher reliability. Let $P_i^\star$ be the desired pose distribution and $\ell_{i,\phi}(\pi) = \ell(f_{\theta_{\text{base}},\phi}(\mathcal{X}_i, \pi), Y_i(\pi))$. The risks $F_{c,i}^{\gamma,r}$ and $\widetilde{F}_{c,i}^{\gamma,r}$ use true and mixed real/pseudo labels under $\bar{Q}_i^{\gamma,r}$, respectively (Appendix B.1). Define

$$\Delta_{\text{pose}}^{\gamma,r} = \frac{1}{N}\sum_{i=1}^N \sup_{\phi\in\Phi}\left|\mathbb{E}_{P_i^\star}\ell_{i,\phi} - \mathbb{E}_{\bar{Q}_i^{\gamma,r}}\ell_{i,\phi}\right|, B_{\text{loss}}^{\gamma,r} = \frac{1}{N}\sum_{i=1}^N \sup_{\phi\in\Phi}\left|F_{c,i}^{\gamma,r}(\phi) - \widetilde{F}_{c,i}^{\gamma,r}(\phi)\right|. \tag{12}$$

Under the Lipschitz and anchor-sampling assumptions in Appendix B.7,

$$\Delta_{\text{pose}}^{\gamma,r} \le \frac{L_\pi}{N}\sum_i W_1(P_i^\star, \bar{Q}_i^{\gamma,r}), \qquad B_{\text{loss}}^{\gamma,r} \le \frac{\gamma L_y}{N}\sum_i(\epsilon_i^{\text{anc}} + L_e r_i). \tag{13}$$

Here W_1 is Wasserstein-1 distance, r_i is the perturbation radius, and ϵ_i^{anc} is teacher error averaged over the actual anchor law A_i. Appendix B.7 defines the Lipschitz constants. Pseudo views improve the coverage bound when their distribution is closer to the desired poses. Their label-bias bound increases with γ and r_i. The pseudo branch should therefore densify nearby coverage while retaining captured images as anchors.

The same effect appears at the gradient level. For squared image loss, let $g_i^\star$ be the ideal client gradient over $P_i^\star$, g_i^{cap} the gradient from captured target poses, and g_i^Q the gradient from nearby poses with true labels. With teacher-rendered pseudo targets,

$$\widetilde{g}_i^\gamma - g_i^\star = \underbrace{g_i^{\text{cap}} - g_i^\star}_{a_i} + \gamma\underbrace{(g_i^Q - g_i^{\text{cap}} + u_i)}_{c_i+u_i}, \qquad \mathbb{E}\|\widehat{\mu}_i - \mu_i\|^2 \le \sigma_{s,i}^2/K_Q. \tag{14}$$

Here a_i is captured-pose bias, c_i the nearby-pose correction, and u_i projected teacher error. The estimator $\widehat{\mu}_i$ averages K_Q independent teacher-target gradients, each with mean $\mu_i = g_i^Q + u_i$ and variance bounded by $\sigma_{s,i}^2$. An intermediate γ balances correction against teacher error; additional poses reduce variance. See Appendices B.10 and B.9.

We next state the server-side bound. Let F_0 be the public risk and F_c the true client risk under P_0 and P_c, respectively. Here P_c uniformly mixes the client target laws $P_i^\star$. The client risks $F_c^{\gamma,r}$ and $\widetilde{F}_c^{\gamma,r}$ average the true-label and pseudo-label risks over clients. Using the allocation-induced weight ρ in Eq. (10), we analyze the surrogate risk

$$\widetilde{\mathcal{F}}_\rho^{\gamma,r}(\phi) = \rho F_0(\phi) + (1-\rho)\widetilde{F}_c^{\gamma,r}(\phi), \tag{15}$$

and the target mixture is $\mathcal{F}_{\rho_\star} = \rho_\star F_0 + (1-\rho_\star)F_c$, where $\rho_\star$ specifies the evaluation weight. Define $\text{Disc}_\Phi(P_0, P_c) = \sup_{\phi\in\Phi}|F_0(\phi) - F_c(\phi)|$.

Theorem 3.1 (Server generalization)**.** *Fix the backbone, client teachers and input packs, pose samplers, and weights $\rho, \rho_\star \in [0,1]$. Assume losses lie in $[0, L_{\max}]$ and are L_ϕ-Lipschitz on $\Phi = \{\phi \in \mathbb{R}^{p_{\text{eff}}} : \|\phi - \phi_0\|_2 \le B_\phi\}$, with $B_\phi > 0$. Let the empirical risks use s_0 public and s_c client supervision packs, independent given these fixed quantities, with mean losses F_0 and $\widetilde{F}_c^{\gamma,r}$. For $0 < \delta < 1$, with probability at least $1-\delta$, uniformly over Φ,*

$$\mathcal{F}_{\rho_\star}(\phi) \le \widehat{\widetilde{\mathcal{F}}}_\rho^{\gamma,r}(\phi) + \mathcal{C}_{\text{PEFT}}(\delta) + (1-\rho)\big(\Delta_{\text{pose}}^{\gamma,r} + B_{\text{loss}}^{\gamma,r}\big) + |\rho - \rho_\star|\text{Disc}_\Phi(P_0, P_c), \tag{16}$$

where $\mathcal{C}_{\mathrm{PEFT}}(\delta) = \rho\Gamma(s_0, \delta) + (1-\rho)\Gamma(s_c, \delta)$ *and*

$$\Gamma(s, \delta) = \frac{2L_\phi B_\phi}{\sqrt{s}} + L_{\max}\sqrt{\frac{p_{\mathrm{eff}} \log(1 + 2\sqrt{s}) + \log(4/\delta)}{2s}}. \tag{17}$$

Each active branch has $s \geq 1$*; a zero-weight branch is omitted.*

Theorem 3.1 connects server generalization to four parts of our design. First, restricting updates to p_{eff} parameters lowers the statistical cost. Second, nearby pseudo targets improve the pose-coverage bound when they better cover the desired views. Third, teacher accuracy controls the label-bias cost $B_{\mathrm{loss}}^{\gamma,r}$. Finally, public data anchor optimization, while clients supply private-scene supervision; the discrepancy term captures mismatch between training and evaluation mixtures. The server thus uses public supervision as an anchor and client pseudo targets for scene-specific 3D coverage. Appendix B.5 gives the proof.

3.3 Client Personalization Convergence

We now formalize the two directions of mutual reinforcement for squared image loss. Write $h_\phi(\mathcal{X}_i, \pi) = f_{\theta_{\mathrm{base}},\phi}(\mathcal{X}_i, \pi)$ and $J_{\phi,i} = \partial h_\phi / \partial \phi$. For nearby pseudo poses, define

$$G_i^r(\phi) = \mathbb{E}_{\pi\sim Q_i^r} \tfrac{1}{2}\|h_\phi(\mathcal{X}_i, \pi) - Y_i(\pi)\|^2, \quad R_i^r(\psi_i) = \mathbb{E}_{\pi\sim Q_i^r} \tfrac{1}{2}\|g_{\psi_i}(\pi) - Y_i(\pi)\|^2, \tag{18}$$

$$D_i^r(\phi, \psi_i) = \mathbb{E}_{\pi\sim Q_i^r} \tfrac{1}{2}\|h_\phi(\mathcal{X}_i, \pi) - g_{\psi_i}(\pi)\|^2. \tag{19}$$

Here $Y_i(\pi)$ at pseudo poses is unavailable during training and is used only for analysis.

Proposition 3.2 (Client teachers control server update bias)**.** *Let* $g_i^{\mathrm{impl}}(\phi)$ *be the gradient of the implemented mixed real/pseudo objective in Eq. (6), and let* $g_i^{\mathrm{true}}(\phi)$ *be the gradient of the same pose-mixture objective with true labels also at pseudo poses. If* $\|\nabla_\phi h_\phi(\mathcal{X}_i, \pi)\|_{\mathrm{op}} \leq J_\phi$*, then*

$$g_i^{\mathrm{impl}}(\phi) - g_i^{\mathrm{true}}(\phi) = \gamma \mathbb{E}_{\pi\sim Q_i^r} J_{\phi,i}(\pi)^\top \big[Y_i(\pi) - g_{\psi_i}(\pi)\big], \qquad \|g_i^{\mathrm{impl}} - g_i^{\mathrm{true}}\| \leq \gamma J_\phi \sqrt{2R_i^r(\psi_i)}. \tag{20}$$

Thus, client teacher accuracy controls the bias of the server-training gradient. The proof is in Appendix B.2; the extension to general differentiable image losses is given in Appendix B.11.

Theorem 3.3 (Client personalization convergence)**.** *Fix the server and pose sampler. Assume* R_i^r *is bounded below,* L_ψ*-smooth, and* $\|\nabla_{\psi_i} g_{\psi_i}(\pi)\|_{\mathrm{op}} \leq J_\psi$*. For* K *exact-gradient consistency steps* $\psi_i^{k+1} = \psi_i^k - \beta \nabla_{\psi_i} D_i^r(\phi, \psi_i^k)$ *with* $0 < \beta \leq 1/(4L_\psi)$*,*

$$\frac{1}{K}\sum_{k=0}^{K-1} \|\nabla R_i^r(\psi_i^k)\|^2 \leq \frac{2\big(R_i^r(\psi_i^0) - R_i^{r,\star}\big)}{\beta K} + 5J_\psi^2 G_i^r(\phi), \tag{21}$$

where $R_i^{r,\star} = \inf_{\psi_i} R_i^r(\psi_i)$.

The residual is proportional to server error $G_i^r(\phi)$: a better server provides a more accurate personalization descent signal (proof in Appendix B.3). Proposition 3.2 gives the reverse direction through teacher-induced server bias. Appendix B.4 relates these two directions, and Appendix B.12 analyzes client sampling and local-update drift.

4 Experiments

4.1 Implementation Detail

We use posed multi-view scenes from Objaverse (Deitke et al., 2023b), with each scene treated as a distinct client. Our main setting contains 5,000 clients, each with 12 posed RGB images and known camera parameters, together with a centrally stored public split of 75K scenes. Three A100s simulate clients with up to 13 workers each: 39 clients run concurrently, handling 117 sampled clients in three batches. Four A100s perform public refinement and one handles server orchestration and aggregation. Due to our available GPU resources, we use a reduced LVSM configuration (Jin et al., 2025) for the server model rather than the full architecture. Only the trainable subset ϕ is updated by clients during federated rounds. We report standard image reconstruction metrics used in NVS: PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). Appendix A gives implementation details and communication accounting.

Table 1: **Server generalization.** Our federated server outperforms centralized reduced LVSM at the same architecture and parameter count; full LVSM provides a larger-model reference. Our server is trained on Objaverse, evaluated zero-shot on GSO, and fine-tuned through the federated pipeline on IBRNet-LLFF for non-IID evaluation.

	Objaverse dataset			GSO (zero-shot)			IBRnet LLFF (non-IID)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
PixelNeRF (Yu et al., 2021)	22.383	0.784	0.365	20.059	0.568	0.418	16.059	0.469	0.665
IBRnet (Wang et al., 2021)	39.491	0.991	0.023	30.538	0.950	0.051	27.383	0.932	0.098
LVSM (Jin et al., 2025) (full)	**41.931**	0.992	0.013	**31.969**	**0.965**	**0.042**	29.170	0.946	0.082
LVSM (Jin et al., 2025) (reduced)	38.391	0.984	0.030	28.395	0.910	0.072	26.961	0.872	0.102
Our server (reduced)	41.852	**0.993**	**0.012**	31.341	0.958	0.056	**29.433**	**0.954**	**0.072**

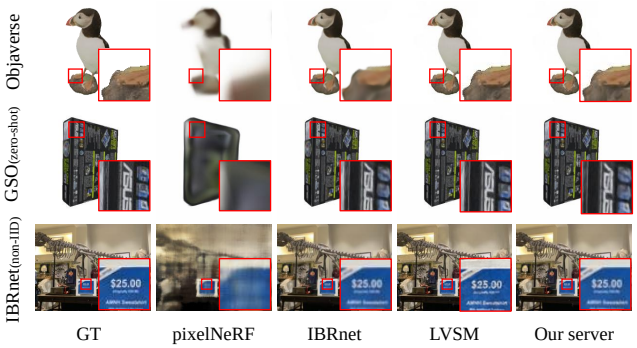


Figure 2: **Qualitative comparison for the server model.** Our federated pipeline reconstructs novel views on Objaverse, zero-shot GSO, and non-IID IBRNet-LLFF. Enlarged regions show the recovered appearance details.

Table 2: **Client personalization.** Our client neural fields outperform recent neural rendering methods on held-out Objaverse and IBRNet-LLFF views. Server guidance improves all metrics.

	Objaverse dataset			IBRnet LLFF		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF (Mildenhall et al., 2020)	25.647	0.873	0.120	20.441	0.515	0.380
InfoNeRF (Kim et al., 2022)	27.312	0.914	0.091	22.669	0.746	0.202
TensoRF (Chen et al., 2022)	40.910	0.987	0.019	23.635	0.758	0.195
FrugalNeRF (Lin et al., 2025)	41.329	0.990	0.017	24.653	0.829	0.138
Ours	**44.474**	**0.998**	**0.009**	**26.560**	**0.858**	**0.136**

4.2 Experimental results

Datasets. We train the server and client fields on Objaverse (Deitke et al., 2023b), then evaluate the server zero-shot on 1,000 GSO scenes (Downs et al., 2022). Both datasets use four randomly selected input views. To evaluate the server under non-IID client data, we use 116 real forward-facing IBRNet-LLFF scenes (Wang et al., 2021). Starting from the Objaverse-trained server, we introduce these scenes as a new client cohort and continue the same mutual-reinforcement procedure. These clients shift both the scene content and camera distribution from synthetic object-centric data to real forward-facing captures. We split the cohort 8:2 into training and test scenes, divide training scenes into public and client sets using the Objaverse allocation, and use four corner views as input.

Quantitative results. We compare the server with PixelNeRF (Yu et al., 2021), IBRNet (Wang et al., 2021), and the full original LVSM (Jin et al., 2025) on held-out evaluation scenes. Table 1 includes centralized reduced LVSM.Our server consistently outperforms the centralized reduced LVSM across Objaverse, zero-shot GSO, and non-IID IBRNet-LLFF, showing that federated client supervision improves server generalization beyond the effect of model capacity alone. Although our server does not outperform the larger full LVSM across all settings, it achieves competitive performance overall and performs better on all IBRNet-LLFF metrics. This result indicates that our learning framework is better optimized for training under non-IID client data. Figure 2 shows the corresponding reconstructions.

For client personalization, we compare with NeRF (Mildenhall et al., 2020), InfoNeRF (Kim et al., 2022), TensoRF (Chen et al., 2022), and FrugalNeRF (Lin et al., 2025) on each client's held-out views. Our client field achieves the best reported results on both Objaverse and LLFF (Table 2, Fig. 3).

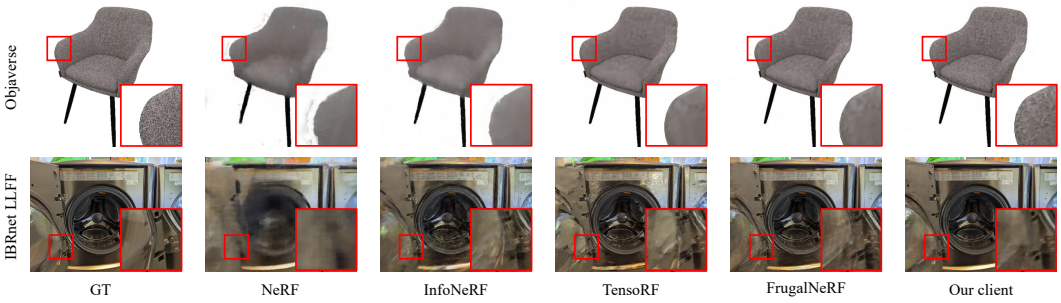


Figure 3: **Qualitative comparison for client models.** Our personalized neural fields reconstruct finer details from sparse observations on Objaverse and IBRNet-LLFF. Enlarged regions compare reconstruction quality against the ground truth.

Table 3: **MIA/PIA accuracy under controlled exposed signals on ShapeNet Cars (Chang et al., 2015).** Uploaded subset updates reveal less scene membership and vehicle-type information to the evaluated attackers than leaked client neural fields. Values are means over 15 runs.

Metric	(1) Ours $\Delta\phi$	(2) Larger subset	(3) Full update	(4) Scratch	(5) Public-warm	(6) Early FL	(7) Leaked NeRF
MIA Acc. ↓	56.8	63.1	70.4	68.9	60.5	58.9	93.2
PIA Acc. ↓	31.4	39.2	47.8	45.1	34.0	33.2	91.6

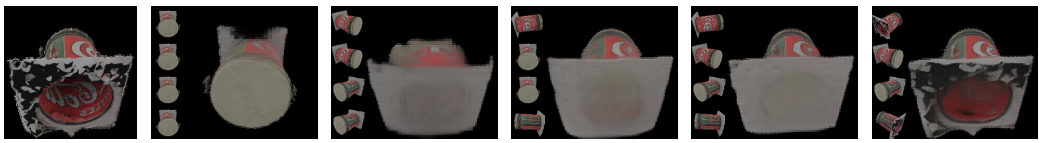


Figure 4: **Server model leakage evaluation.** Each setting shows leaked input views on the left and the attacker's target-view prediction on the right. Even with the intercepted client update, missing target-region coverage produces distorted or repeated visible content. Four surrounding views provide the full-context control at the same input count.

The global server supplies supervision beyond the sparse captured views, improving the personalized reconstruction while the scene representation remains local.

4.3 Privacy Leakage Evaluation

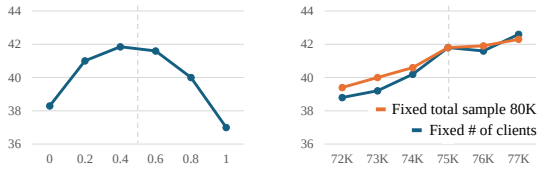
We evaluate update-level exposure under an honest-but-curious server on ShapeNet Cars (Chang et al., 2015). Membership inference (MIA) (Shokri et al., 2017) predicts whether a candidate car scene produced an update; property inference (PIA) (Ganju et al., 2018) predicts the vehicle type, following the property task in FedMeNF (Yun et al., 2025). Appendices C.1 and C.2 describe the attacks, and Table 5 reports repeated-run statistics.

Table 3 compares seven exposure conditions. The main condition exposes our uploaded subset update (1), whereas the positive control leaks the client NeRF itself (7). The uploaded update yields substantially lower MIA and PIA accuracy than the leaked neural field. Increasing the trainable subset or exposing a full-model update (2–3) strengthens both attacks. Relative to the scratch server (4), public warm-up (5) lowers attack accuracy, and the comparison between Early FL (6) and Ours (1) shows that leakage further decreases as federated training matures. Thus, restricted updates and a server trained across scenes reduce exposure of client identity and vehicle type.

Limited-context reconstruction from the aggregated server. The server state broadcast after each round absorbs the participating clients' updates, so we test whether a freshly aggregated late-round model reproduces a client's scene beyond what its leaked images show. The attacker is the honest-but-curious server holding $\bar{\phi}^{t+1}$ from Eq. (9) immediately after aggregation in a late round, together with a limited subset of a target client's posed images, which it uses as NVS context. Figure 4 compares one to four leaked views with a four-view control that surrounds the object. When the leaked views leave the target region uncovered, the model distorts or repeats visible ppearance

Table 4: **Ablation study on Objaverse.** Removing the training losses or pseudo-target supervision reduces both server generalization and client personalization. The no-public-data condition evaluates the federated pathway without a public anchor.

	Server model			Client model		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
w/o $\mathcal{L}_{ent}$	37.238	0.984	0.024	41.021	0.994	0.023
w/o $\mathcal{L}_{srv}$	38.082	0.981	0.021	41.345	0.984	0.012
w/o $\mathcal{L}_{cons}$	39.380	0.989	0.017	40.982	0.983	0.024
w/o public data	32.182	0.962	0.041	41.390	0.989	0.013
w/o pseudo target	38.296	0.979	0.028	42.292	0.990	0.012
Ours	**41.852**	**0.993**	**0.012**	**44.474**	**0.998**	**0.009**



(a) PSNR / γ for pseudo target (b) PSNR / # of $\mathcal{D}_{\text{pub}}$

Figure 5: **Effect of γ and $|\mathcal{D}_{\text{pub}}|$ on server PSNR.** (a) Excessive pseudo-target supervision degrades reconstruction; $\gamma = 0.4$ gives the best result. (b) Public-data scaling with 5K clients fixed or 80K public and client scenes in total.

rather than recovering the missing content, and the surrounding-view control outperforms four views concentrated on one side. Recovery therefore depends on the visual coverage of the leaked images, not on the client's contribution having just been aggregated into the model; the aggregated update does not act as a renderable representation of the client scene. Appendix C.3 gives the camera configurations and quantitative results.

4.4 Ablations and Analysis

Table 4 evaluates the losses, pseudo targets, and public supervision in our pipeline. Removing $\mathcal{L}_{\text{srv}}$ disables local server-model training and reduces both models' reconstruction quality. The entropy regularizer improves local fields and, through their pseudo targets, server training. Removing $\mathcal{L}_{\text{cons}}$ directly weakens client refinement and also affects the server: subsequent pseudo targets come from less accurate client teachers. Removing pseudo targets likewise reduces server generalization and the quality of the model returned to clients. The no-public-data ablation evaluates the same client-to-server training path without a public anchor.

Figure 5(a) varies the fixed pseudo-target budget γ. Too little pseudo supervision leaves nearby camera poses weakly constrained; too much increases reliance on teacher predictions at the expense of captured images. The intermediate optimum matches the coverage–bias and gradient trade-offs analyzed in Appendix D.2. Figure 5(b) compares a fixed cohort of 5K clients with a fixed total of 80K public and client scenes. Below 75K public scenes, the fixed-total setting has more clients and more training scenes overall; this intended difference shows how additional distributed scenes compensate for limited public data. At 75K, the two allocations coincide. Above this point, the fixed-5K setting retains more client scenes and has a modest advantage, while the curves remain close. Please refer to Appendix D.1 for the client-side crossover and its link to server guidance.

5 Conclusion

We train a large image-conditioned NVS model from distributed private scenes while improving client personalization. Local fields provide scene-specific pseudo targets, and the global server guides sparse-view refinement; raw images and renderable client parameters remain local. Our analysis connects these feedback directions to pose coverage, teacher bias, and server prediction error, and examines parameter complexity, scene sampling, and local-update drift. Experiments demonstrate improved personalization and competitive server generalization.

Limitation is that privacy leakage may still arise from communicated gradients or local updates under stronger attack scenarios. Analyzing and mitigating such update-level leakage would further improve the practicality of this pipeline. Future directions include incorporating stronger privacy mechanisms (e.g., secure aggregation and differential privacy applied to communicated updates), extending the teacher paradigm beyond radiance fields to other compact scene representations, and learning from clients under highly heterogeneous data distributions.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks. (1) We used an LLM-based coding agent to assist in implementing parts of the proposed method, such as individual functions and components. The coding agent did not implement the proposed method in its entirety; the overall design of the method and the remaining implementation were carried out by the authors. (2) We used LLM-based tools to draft parts of the manuscript and to improve the clarity, fluency, and grammatical correctness of the English text. (3) We used LLM-based tools to search for and identify related work. We did not use generative AI tools for any other task in the required-disclosure category, including generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims or writing proofs, proposing or refining hypotheses, designing the research methodology or experiments, cleaning or reformatting datasets, qualitative or thematic data analysis, or interpreting results. We have reviewed all AI-assisted work. All AI-generated code was reviewed and tested for correctness by the authors before being integrated into our codebase. All AI-assisted text was carefully reviewed and revised by the authors, and every reference identified with the help of AI tools was verified against the original publication before being cited. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Int. Conf. Comput. Vis.*, 2021.

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)*, 2017.

Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *Eur. Conf. Comput. Vis.*, 2022.

Yabo Chen, Jiemin Fang, Yuyang Huang, Taoran Yi, Xiaopeng Zhang, Lingxi Xie, Xinggang Wang, Wenrui Dai, Hongkai Xiong, and Qi Tian. Cascade-zero123: One image to highly consistent 3d with self-prompted nearby views. In *Eur. Conf. Comput. Vis.*, 2024.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023a.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 13142–13153, 2023b.

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *IEEE Int. Conf. on Robot. and Autom.*, pp. 2553–2560. IEEE, 2022.

Cynthia Dwork. Differential privacy. In *International Colloquium on Automata, Languages and Programming (ICALP)*, 2006.

Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 619–633, 2018.

Tao Guo, Song Guo, and Junxiao Wang. pfedprompt: Learning personalized prompt for vision-language models in federated learning. In *Proceedings of the ACM Web Conference (WWW)*, pp. 1364–1374, 2023. doi: 10.1145/3543507.3583518.

Lachlan Holden, Feras Dayoub, David Harvey, and Tat-Jun Chin. Federated neural radiance fields. *arXiv preprint arXiv:2305.01163*, 2023.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. In *Int. Conf. Learn. Represent.*, 2024.

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. In *Int. Conf. Learn. Represent.*, 2025.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Hubert Eichner, Salim El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaid Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Sanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, Rasmus Pagh, Hang Qi, Daniel Ramage, Ramesh Raskar, Mariana Raykova, Dawn Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 14(1–2):1–210, June 2021. ISSN 1935-8237. doi: 10.1561/2200000083. URL https://doi.org/10.1561/2200000083.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 2023.

Mijeong Kim, Seonguk Seo, and Bohyung Han. Infonerf: Ray entropy minimization for few-shot neural volume rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

Junoh Lee, Hyunjun Jung, Jin-Hwi Park, Inhwan Bae, and Hae-Gon Jeon. Geometry-aware projective mapping for unbounded neural radiance fields. In *Int. Conf. Learn. Represent.*, 2024.

Chin-Yang Lin, Chung-Ho Wu, Chang-Han Yeh, Shih-Han Yen, Cheng Sun, and Yu-Lun Liu. Frugalnerf: Fast convergence for extreme few-shot novel view synthesis without learned priors. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 11227–11238, 2025.

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)*, pp. 1273–1282, 2017.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis.*, 2020.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 41(4):1–15, 2022.

Bikang Pan, Wei Huang, and Ye Shi. Federated learning from vision-language foundation models: Theoretical analysis and method. In *Neural Inform. Process. Syst.*, 2024.

Chen Qiu, Xingyu Li, Chaithanya Kumar Mummadi, Madan Ganesh, Zhenzhen Li, Lu Peng, and Wan-Yi Lin. Federated text-driven prompt generation for vision-language models. In *Int. Conf. Learn. Represent.*, 2024.

Lorenzo Sani, Alexandru-Andrei Iacob, Zeyu Cao, Royson Lee, Bill Marino, Yan Gao, Dongqi Cai, Zexi Li, Wanru Zhao, Xinchi Qiu, and Nicholas D. Lane. Photon: Federated llm pre-training. *Proceedings of Machine Learning and Systems*, 7, 2025.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Shangchao Su, Mingzhao Yang, Bin Li, and Xiangyang Xue. Federated adaptive prompt tuning for multi-domain collaborative learning. In *Assoc. Advancem. Artific. Intell.*, pp. 15117–15125, 2024.

Teppei Suzuki. Fed3dgs: Scalable 3d gaussian splatting with federated learning. *arXiv preprint arXiv:2403.11460*, 2024.

Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul P Srinivasan, Jonathan T Barron, and Henrik Kretzschmar. Block-nerf: Scalable large scene neural view synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 8248–8258, 2022.

Zaid Tasneem, Akshat Dave, Abhishek Singh, Kushagra Tiwary, Praneeth Vepakomma, Ashok Veeraraghavan, and Ramesh Raskar. Decentnerfs: Decentralized neural radiance fields from crowdsourced images. In *Eur. Conf. Comput. Vis.*, pp. 144–161, 2024.

Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4):600–612, 2004.

Ziyao Wang, Zheyu Shen, Yexiao He, Guoheng Sun, Hongyi Wang, Lingjuan Lyu, and Ang Li. Flora: Federated fine-tuning large language models with heterogeneous low-rank adaptations. In *Neural Inform. Process. Syst.*, 2024.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 4578–4587, 2021.

Junhyeog Yun, Minui Hong, and Gunhee Kim. Fedmenf: Privacy-preserving federated meta-learning for neural fields. In *Int. Conf. Comput. Vis.*, 2025.

Bokang Zhang, Yanglin Zhang, Zhikun Zhang, Jinglan Yang, Lingying Huang, and Junfeng Wu. s^2nerf: Privacy-preserving training framework for nerf. In *Proc. ACM Conf. on Computer and Communications Security (CCS)*, 2024a.

Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Tong Yu, Guoyin Wang, and Yiran Chen. Towards building the federatedgpt: Federated instruction tuning. In *ICASSP 2024-2024 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 6915–6919. IEEE, 2024b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

Yintian Zhang and Ziyu Shao. Federated neural radiance field for distributed intelligence. *arXiv preprint arXiv:2406.10474*, 2024.

Algorithm 1 Federated mutual reinforcement for large NVS and client neural fields

Input: server model $f_{\theta_{\text{base}},\phi^0}$, client datasets $\{\mathcal{D}_i\}_{i=1}^N$, centrally stored server public set $\mathcal{D}_{\text{pub}}$.
Select a trainable subset ϕ by the server scope rule in Appendix A.2; freeze all other parameters.
for round $t = 0, \ldots, T-1$ **do**
  Server samples $\mathcal{S}_t$ and sends ϕ^t and its trainable key list; clients cache θ_{base}.
  for client $i \in \mathcal{S}_t$ in parallel **do**
    Fit or warm-start the private neural field g_{ψ_i} on local posed images $\mathcal{D}_i$.
    Sample nearby target poses and render pseudo targets $\tilde{I}_i(\tilde{\pi}) = g_{\psi_i}(\tilde{\pi})$ locally.
    Retain the broadcast snapshot ϕ^t; update a separate copy to ϕ_i^t for E steps using real and pseudo targets.
    Refine g_{ψ_i} using the fixed broadcast model $f_{\theta_{\text{base}},\phi^t}$ in Eq. (8).
    Upload only $\Delta\phi_i^t = \phi_i^t - \phi^t$.
  end for
  Form $\bar{\phi}^{t+1}$ using Eq. (9); public refinement gives ϕ^{t+1}, or set $\phi^{t+1} = \bar{\phi}^{t+1}$ when $\mathcal{D}_{\text{pub}}$ is empty.
end for

A Implementation Details and FL Efficiency

We implement the full pipeline in PyTorch. To accommodate the full federated training pipeline within our available GPU resources, we use a reduced LVSM configuration with token dimension $d = 384$, head dimension $d_{\text{head}} = 64$, 6 encoder layers, 6 decoder layers, and 1024 latent tokens. We use gradient checkpointing and fp16 mixed precision for memory efficiency. During federated PEFT, only the trainable subset ϕ is communicated and updated by clients.

A.1 Client neural field architecture

The client model is a TensoRF-style factorized radiance field. We use an axis-aligned bounding box $[-1.8, 1.8]^3$ for normalized Objaverse scenes, a 128^3 grid, 8 density components, and 24 appearance components. Density and appearance are represented by three plane factors and three line factors each. The appearance factors are projected by a linear basis matrix to a 64-dimensional appearance feature. A positional encoder with 4 view-direction frequencies is concatenated to this feature and passed through a two-layer MLP of width 64 followed by a sigmoid RGB head. Rendering uses 128 samples per ray, near/far bounds $0.1/6.0$, and standard alpha compositing.

Each client optimizes the grid factors with learning rate 2×10^{-2} and the neural modules with learning rate 10^{-3}. We train each local neural field for 5000 steps with 1024 rays per step. We use the entropy regularizer with weight 0.001 and sample 8 unseen-pose candidates for the entropy term. The client neural field checkpoint, pseudo poses, and pseudo images are never uploaded.

A.2 Server trainable subset selection

The default trainable scope is `decoder_only`, selected by module name.

We first set `requires_grad=False` for all server parameters. We then unfreeze parameters whose names start with

$$\{\texttt{transformer_decoder}, \texttt{transformer_input_layernorm_decoder}, \texttt{target_pose_tokenizer}, \texttt{image_token_decoder}\} \tag{22}$$

and the latent parameter `n_light_field_latent`. The same key list is used by the server and by all clients. Clients receive this subset, update only these tensors, and compute deltas by subtracting the received base state. The `all` scope is used only in privacy ablations and updates every parameter.

This rule makes the communicated parameters correspond to target-pose conditioning and image decoding, while the image encoder and most shared representation layers remain fixed during client communication. Thus the local update changes how the server model renders from a private view pack, but it does not turn the uploaded tensor into a standalone renderable scene representation.

A.3 Step 4: client refinement from the server model

The client retains the broadcast snapshot ϕ^t while a separate copy is updated to ϕ_i^t for upload. For personalization, $f_{\theta_{\text{base}},\phi^t}$ predicts nearby views from the private context $\mathcal{X}_i$. These predictions are detached and supervise g_{ψ_i} alongside captured-image and entropy losses:

$$\mathcal{L}_{\text{client}}(\psi_i) = \mathcal{L}_{\text{pix}}(\psi_i) + \lambda_{\text{ent}}\mathcal{L}_{\text{ent}}(\psi_i) + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}}^{(i)}(\phi^t, \psi_i). \tag{23}$$

Captured images anchor the observed appearance; nearby consistency adds server guidance beyond those views. Theorem 3.3 analyzes the squared-loss consistency component with the server fixed; Eq. (23) specifies the complete implemented client objective. The next received broadcast incorporates other clients' updates, completing the global-to-local feedback path.

A.4 FL efficiency accounting

Our main Objaverse configuration uses $T = 3000$ rounds, $m = 117$ sampled clients per round, $E = 100$ local server-model update steps, and $E_0 = 3000$ public-refinement steps per round. The communicated subset contains $p_{\text{eff}} = 11{,}237{,}376$ parameters. For dense subset exchange with a cached encoder, let b be the bytes per transmitted parameter. The logical payload is

$$C_{\text{client,round}} = 2bp_{\text{eff}}, \qquad C_{\text{total}} = 2bp_{\text{eff}} \sum_{t=0}^{T-1} |\mathcal{S}_t|. \tag{24}$$

At 16-bit precision, one upload or download is 21.4 MiB, or 42.9 MiB bidirectionally per participating client per round. The stated m, T give 15.78 TB across all client participations, averaging 3.16 GB per client under uniform sampling. These are independent-client payload estimates, excluding the initial encoder transfer and transport overhead; shared-memory simulation and broadcast reuse change physical traffic. FP32 transmission doubles the payload. Local optimization steps do not add communication until the final update is sent. Three client-worker GPUs and five server/orchestration GPUs are allocated for roughly 14 days, corresponding to about 1,008 client GPU-hours and 1,680 server GPU-hours.

A.5 Per-client computation

Three A100s host up to 39 concurrent client workers, processing each round's 117 clients in three batches. Peak memory is 2.86 GB per client, including 231 MB for the neural field; mean client learning time is 152 s. A shared GPU simulates multiple owners, while each owner's local workload includes neural-field fitting and server-model execution.

B Additional Derivations for Mutual Reinforcement, Generalization, and Convergence

This appendix contains the full derivations behind the statements used in Sec. 3.2 and Sec. 3.3. We use squared image loss in Sections B.2–B.3 for clarity; Section B.11 gives the corresponding Taylor expansion for general differentiable image losses.

B.1 Definitions used in the federated analysis

For client i, condition on $\mathcal{X}_i$ and write $\hat{Y}_{i,\phi}(\pi) = h_\phi(\mathcal{X}_i, \pi) = f_{\theta_{\text{base}},\phi}(\mathcal{X}_i, \pi)$ and $\tau_i(\pi) = g_{\psi_i}(\pi)$. $Y_i(\pi)$ is the true image, used analytically at unobserved poses. $P_i^\star$ is the desired target-pose law and P_i^{cap} the captured-target law. For the actual anchor law A_i, let $K_i^r(\cdot \mid \pi_a)$ be the bounded perturbation kernel and set

$$\Pi_i(d\pi_a, d\pi) = A_i(d\pi_a)K_i^r(d\pi \mid \pi_a), \qquad Q_i^r(d\pi) = \int \Pi_i(d\pi_a, d\pi). \tag{25}$$

Its support satisfies $d_\pi(\pi_a, \pi) \le r_i$, where $r = (r_1, \ldots, r_N)$ records the client radii. This notation accommodates distinct captured-target and anchor samplers. All pose laws are conditional on the fixed input pack.

Write $\bar{Q}_i^{\gamma,r} = (1-\gamma)P_i^{\mathrm{cap}} + \gamma Q_i^r$; $\bar{Q}_i^{\gamma}$ abbreviates it for fixed r. The sampled real/pseudo branch determines the label even when pose supports overlap. With $\ell_{i,\phi}(\pi) = \ell(\hat{Y}_{i,\phi}(\pi), Y_i(\pi))$, define

$$\begin{aligned} F_{c,i}^{\star}(\phi) &= \mathbb{E}_{P_i^{\star}} \ell_{i,\phi}(\pi), \\ F_{c,i}^{\gamma,r}(\phi) &= (1-\gamma)\mathbb{E}_{P_i^{\mathrm{cap}}} \ell_{i,\phi}(\pi) + \gamma \mathbb{E}_{Q_i^r} \ell_{i,\phi}(\pi), \\ \widetilde{F}_{c,i}^{\gamma,r}(\phi) &= (1-\gamma)\mathbb{E}_{P_i^{\mathrm{cap}}} \ell_{i,\phi}(\pi) + \gamma \mathbb{E}_{Q_i^r} \ell(\hat{Y}_{i,\phi}(\pi), \tau_i(\pi)). \end{aligned} \tag{26}$$

$F_c = N^{-1}\sum_i F_{c,i}^{\star}$, $F_c^{\gamma,r}$, and $\widetilde{F}_c^{\gamma,r}$ are the respective client averages. The gaps used throughout are

$$\begin{aligned} \Delta_{\mathrm{pose}}^{\gamma,r} &= \frac{1}{N}\sum_i \sup_{\phi\in\Phi} |F_{c,i}^{\star}(\phi) - F_{c,i}^{\gamma,r}(\phi)|, \\ B_{\mathrm{loss}}^{\gamma,r} &= \frac{1}{N}\sum_i \sup_{\phi\in\Phi} |F_{c,i}^{\gamma,r}(\phi) - \widetilde{F}_{c,i}^{\gamma,r}(\phi)|. \end{aligned} \tag{27}$$

Here Φ is the parameter ball in Theorem 3.1. Vector norms are Euclidean. The identities extend to averaging over a fixed context-sampling law; gradient calculations assume interchange of differentiation and expectation and the stated uniform derivative bounds.

B.2 Proof of Proposition 3.2: client teacher quality controls server update bias

Proof. Fix the client teacher and input views. Write $\hat{Y}_\phi(\pi) = h_\phi(\mathcal{X}_i, \pi)$, $\tau_i(\pi) = g_{\psi_i}(\pi)$, and $J_{\phi,i} = \partial \hat{Y}_\phi / \partial \phi$. Assume differentiation and expectation commute. The implemented gradient in Eq. (6) is

$$g_i^{\mathrm{impl}}(\phi) = (1-\gamma)\mathbb{E}_{\pi\sim P_i^{\mathrm{cap}}} J_{\phi,i}(\pi)^\top \big(\hat{Y}_\phi(\pi) - Y_i(\pi)\big) \tag{28}$$

$$+ \gamma \mathbb{E}_{\pi\sim Q_i^r} J_{\phi,i}(\pi)^\top \big(\hat{Y}_\phi(\pi) - \tau_i(\pi)\big). \tag{29}$$

If true target images were available also at the pseudo poses, the corresponding gradient would be

$$g_i^{\mathrm{true}}(\phi) = (1-\gamma)\mathbb{E}_{\pi\sim P_i^{\mathrm{cap}}} J_{\phi,i}(\pi)^\top \big(\hat{Y}_\phi(\pi) - Y_i(\pi)\big) \tag{30}$$

$$+ \gamma \mathbb{E}_{\pi\sim Q_i^r} J_{\phi,i}(\pi)^\top \big(\hat{Y}_\phi(\pi) - Y_i(\pi)\big). \tag{31}$$

The captured-view terms cancel exactly, yielding

$$g_i^{\mathrm{impl}}(\phi) - g_i^{\mathrm{true}}(\phi) = \gamma \mathbb{E}_{\pi\sim Q_i^r} J_{\phi,i}(\pi)^\top \big(Y_i(\pi) - \tau_i(\pi)\big). \tag{32}$$

If $\|J_{\phi,i}(\pi)\|_{\mathrm{op}} \le J_\phi$, then

$$\|g_i^{\mathrm{impl}} - g_i^{\mathrm{true}}\| \le \gamma J_\phi \mathbb{E}_{\pi\sim Q_i^r} \|Y_i(\pi) - \tau_i(\pi)\| \tag{33}$$

$$\le \gamma J_\phi \sqrt{\mathbb{E}_{\pi\sim Q_i^r} \|Y_i(\pi) - \tau_i(\pi)\|^2} \tag{34}$$

$$= \gamma J_\phi \sqrt{2R_i^r(\psi_i)}. \tag{35}$$

□

Interpretation. Only the pseudo branch contributes teacher-label bias because captured views use their true images. The factor γ explicitly quantifies how strongly teacher quality affects the uploaded server update.

B.3 Proof of Theorem 3.3: LVSM consistency is a client-personalization descent signal

Proof. Fix the server parameters, $\mathcal{X}_i$, and Q_i^r during the K client steps. Write $\tau_\psi(\pi) = g_\psi(\pi)$ and $h_\phi(\pi) = f_{\theta_{\mathrm{base}},\phi}(\mathcal{X}_i, \pi)$. The squared-loss risks R_i^r, D_i^r, G_i^r are defined in Eqs. (18)–(19). Assume

R_i^r is bounded below and L_ψ-smooth on the iterate segments, and $\|J_{\psi,i}(\pi;\psi)\|_{\mathrm{op}} \le J_\psi$ there, where $J_{\psi,i} = \partial\tau_\psi/\partial\psi$. With the server prediction fixed, the two gradients are

$$\nabla_\psi R_i^r(\psi) = \mathbb{E}_{Q_i^r} J_{\psi,i}(\pi;\psi)^\top \big(\tau_\psi(\pi) - Y_i(\pi)\big),$$
$$\nabla_\psi D_i^r(\phi,\psi) = \mathbb{E}_{Q_i^r} J_{\psi,i}(\pi;\psi)^\top \big(\tau_\psi(\pi) - h_\phi(\pi)\big).$$

Subtracting cancels the client prediction and isolates the server error:

$$\begin{aligned} \nabla_\psi D_i^r(\phi,\psi) &= \nabla_\psi R_i^r(\psi) - q_i(\phi,\psi), \\ q_i(\phi,\psi) &= \mathbb{E}_{Q_i^r} J_{\psi,i}(\pi;\psi)^\top (h_\phi(\pi) - Y_i(\pi)). \end{aligned} \tag{36}$$

The Jacobian bound and Cauchy–Schwarz inequality yield

$$\|q_i(\phi,\psi)\| \le J_\psi \mathbb{E}_{Q_i^r}\|h_\phi - Y_i\| \le J_\psi \sqrt{\mathbb{E}_{Q_i^r}\|h_\phi - Y_i\|^2} = J_\psi \sqrt{2G_i^r(\phi)}. \tag{37}$$

Set $a_k = \nabla R_i^r(\psi_i^k)$ and $q_k = q_i(\phi,\psi_i^k)$. For the update $\psi_i^{k+1} = \psi_i^k - \beta(a_k - q_k)$, smoothness gives

$$\begin{aligned} R_i^r(\psi_i^{k+1}) &\le R_i^r(\psi_i^k) - \beta\|a_k\|^2 + \beta\langle a_k, q_k\rangle + \frac{L_\psi\beta^2}{2}\|a_k - q_k\|^2 \\ &\le R_i^r(\psi_i^k) - \left(\frac{3\beta}{4} - L_\psi\beta^2\right)\|a_k\|^2 + (\beta + L_\psi\beta^2)\|q_k\|^2, \end{aligned} \tag{38}$$

using $\langle a_k, q_k\rangle \le \frac{1}{4}\|a_k\|^2 + \|q_k\|^2$ and $\|a_k - q_k\|^2 \le 2\|a_k\|^2 + 2\|q_k\|^2$. Since $0 < \beta \le 1/(4L_\psi)$,

$$R_i^r(\psi_i^{k+1}) \le R_i^r(\psi_i^k) - \frac{\beta}{2}\|a_k\|^2 + \frac{5\beta}{2}J_\psi^2 G_i^r(\phi). \tag{39}$$

Summing over k and using $R_i^r(\psi_i^K) \ge R_i^{r,\star}$ gives

$$\frac{\beta}{2}\sum_{k=0}^{K-1}\|a_k\|^2 \le R_i^r(\psi_i^0) - R_i^{r,\star} + \frac{5\beta K}{2}J_\psi^2 G_i^r(\phi). \tag{40}$$

Dividing by $\beta K/2$ proves

$$\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla R_i^r(\psi_i^k)\|^2 \le \frac{2(R_i^r(\psi_i^0) - R_i^{r,\star})}{\beta K} + 5J_\psi^2 G_i^r(\phi). \tag{41}$$

□

Interpretation. The server error $G_i^r(\phi)$ controls both the gradient discrepancy and the residual in the client bound. Reducing it provides a more accurate descent signal for personalization, establishing the generalization-to-personalization direction.

B.4 Interpretation of the mutual-reinforcement coupling

Proposition 3.2 and Theorem 3.3 give the two feedback directions:

$$\|g_i^{\mathrm{impl}} - g_i^{\mathrm{true}}\| \le \gamma J_\phi \sqrt{2R_i^r(\psi_i)},$$
$$\frac{1}{K}\sum_{k=0}^{K-1}\|\nabla R_i^r(\psi_i^k)\|^2 \le \frac{2(R_i^r(\psi_i^0) - R_i^{r,\star})}{\beta K} + 5J_\psi^2 G_i^r(\phi). \tag{42}$$

Client teacher error controls the server-update bias; server prediction error controls the client-refinement residual. These dependencies explain why the two models reinforce each other: client improvements strengthen the supervision sent to the server, and server improvements strengthen the consistency signal returned to the clients.

B.5 Proof of Theorem 3.1: server generalization bound

Sampling model. Condition on the backbone, client teachers, input packs, pose samplers, and parameter ball Φ. P_c selects one of the N clients uniformly and then samples a target from $P_i^\star$. Draw s_0 public and s_c client supervision packs independently conditional on these objects. Client packs follow the real/pseudo law in Appendix B.1. Let their losses be in $[0, L_{\max}]$, uniformly L_ϕ-Lipschitz in ϕ, with means F_0 and $\widetilde{F}_c^{\gamma,r}$. Their sample means form

$$\widehat{\widetilde{\mathcal{F}}}_\rho^{\gamma,r} = \rho\widehat{F}_0 + (1-\rho)\widehat{\widetilde{F}}_c^{\gamma,r}. \tag{43}$$

The weights $\rho, \rho_\star$ are fixed before sampling. The counts s_0, s_c measure fresh supervision-pack draws from these fixed laws; n_0, n_c^{eff} in Eq. (10) count the scene allocation.

Proof. Consider either branch, with s independent packs, population risk F and sample mean $\widehat{F}$. Take an ε-net of Φ with $\varepsilon = B_\phi/\sqrt{s}$. A maximal ε-separated set is such a net: its disjoint radius-$\varepsilon/2$ balls lie within a radius-$B_\phi + \varepsilon/2$ ball. Comparing their volumes gives

$$|\mathcal{N}_\varepsilon| \le (1 + 2B_\phi/\varepsilon)^{p_{\text{eff}}} = (1 + 2\sqrt{s})^{p_{\text{eff}}}. \tag{44}$$

For a fixed net point v, Hoeffding's inequality yields

$$\Pr\{|F(v) - \widehat{F}(v)| > a\} \le 2\exp(-2sa^2/L_{\max}^2). \tag{45}$$

A union bound over the net, with failure probability $\delta/2$ for this branch, gives

$$\max_{v\in\mathcal{N}_\varepsilon} |F(v) - \widehat{F}(v)| \le L_{\max}\sqrt{\frac{p_{\text{eff}}\log(1+2\sqrt{s}) + \log(4/\delta)}{2s}}. \tag{46}$$

For every $\phi \in \Phi$, choose v within ε. Both risks are L_ϕ-Lipschitz, so

$$|F(\phi) - \widehat{F}(\phi)| \le |F(v) - \widehat{F}(v)| + 2L_\phi\varepsilon \le \Gamma(s,\delta). \tag{47}$$

A second union bound covers both branches. With probability at least $1-\delta$, uniformly over Φ,

$$\left|\widetilde{\mathcal{F}}_\rho^{\gamma,r}(\phi) - \widehat{\widetilde{\mathcal{F}}}_\rho^{\gamma,r}(\phi)\right| \le \rho\Gamma(s_0,\delta) + (1-\rho)\Gamma(s_c,\delta) = \mathcal{C}_{\text{PEFT}}(\delta). \tag{48}$$

A zero-weight branch contributes no term.

Insert the true-label mixture $\mathcal{F}_\rho^{\gamma,r} = \rho F_0 + (1-\rho)F_c^{\gamma,r}$:

$$\begin{aligned}\mathcal{F}_{\rho_\star} - \widetilde{\mathcal{F}}_\rho^{\gamma,r} = (\mathcal{F}_{\rho_\star} - \mathcal{F}_\rho) + (\mathcal{F}_\rho - \mathcal{F}_\rho^{\gamma,r}) \\ + (\mathcal{F}_\rho^{\gamma,r} - \widetilde{\mathcal{F}}_\rho^{\gamma,r}).\end{aligned} \tag{49}$$

All risks are evaluated at the same ϕ. The mixture discrepancy satisfies

$$|\mathcal{F}_{\rho_\star} - \mathcal{F}_\rho| = |\rho_\star - \rho|\,|F_0 - F_c| \le |\rho_\star - \rho|\text{Disc}_\Phi(P_0, P_c). \tag{50}$$

The definitions in Appendix B.1 give

$$\mathcal{F}_\rho - \mathcal{F}_\rho^{\gamma,r} = (1-\rho)(F_c - F_c^{\gamma,r}) \le (1-\rho)\Delta_{\text{pose}}^{\gamma,r}, \tag{51}$$

$$\mathcal{F}_\rho^{\gamma,r} - \widetilde{\mathcal{F}}_\rho^{\gamma,r} = (1-\rho)(F_c^{\gamma,r} - \widetilde{F}_c^{\gamma,r}) \le (1-\rho)B_{\text{loss}}^{\gamma,r}. \tag{52}$$

Adding these three bounds to Eq. (48) proves

$$\begin{aligned}\mathcal{F}_{\rho_\star}(\phi) \le \widehat{\widetilde{\mathcal{F}}}_\rho^{\gamma,r}(\phi) + \mathcal{C}_{\text{PEFT}}(\delta) \\ + (1-\rho)\big(\Delta_{\text{pose}}^{\gamma,r} + B_{\text{loss}}^{\gamma,r}\big) + |\rho - \rho_\star|\text{Disc}_\Phi(P_0, P_c)\end{aligned} \tag{53}$$

for every $\phi \in \Phi$ on the same event. □

Interpretation. The estimation term separates parameter dimension and sample count from pose coverage, teacher-label error, and mixture mismatch. Restricted updates control statistical complexity, nearby targets supply camera coverage, and accurate teachers limit label bias. The discrepancy term quantifies the change from the training risk mixture to the evaluation mixture. Together, these terms explain the roles of the trainable subset, local 3D supervision, and public/client allocation. In the implementation, public refinement supplies an optimization anchor and client fields supply private-scene targets; the bound evaluates their weighted risks under the stated sampling model.

B.6 Proof of Eq. (11): nearby pseudo targets supervise local pose response

Proof. Let $A(\delta) = H_\phi(\delta) - T_\psi(\delta)$ in a local SE(3) coordinate chart, with $\mathbb{E}\delta = 0$ and $\Sigma_\delta = \mathbb{E}[\delta\delta^\top]$. Assume A has a bounded, Lipschitz Hessian on the perturbation neighborhood and $\|\delta\| \le r_0 \le 1$. For output coordinate j, write $H_j = \nabla^2_\delta A_j(0)$ and $b_j(\delta) = \delta^\top H_j \delta$. Taylor expansion gives

$$A(\delta) = A(0) + J_A\delta + \tfrac{1}{2}b(\delta) + R_3(\delta), \qquad \|R_3(\delta)\| \le C\|\delta\|^3, \tag{54}$$

where $J_A = J_\pi H_\phi - J_\pi T_\psi$ at the anchor pose. Expanding the squared residual through second order,

$$\begin{aligned}\tfrac{1}{2}\|A(\delta)\|^2 = \tfrac{1}{2}\|A(0)\|^2 + \langle A(0), J_A\delta\rangle + \tfrac{1}{2}\|J_A\delta\|^2 \\ + \tfrac{1}{2}\sum_j A_j(0)\delta^\top H_j\delta + \mathcal{O}(\|\delta\|^3).\end{aligned} \tag{55}$$

The linear term has zero expectation, while $\mathbb{E}[\delta^\top H_j\delta] = \operatorname{tr}(H_j\Sigma_\delta)$. Define

$$\mathcal{R}_{\mathrm{curv}} = \tfrac{1}{2}\sum_j A_j(0)\operatorname{tr}(H_j\Sigma_\delta), \qquad \|J\|^2_{\Sigma_\delta} = \operatorname{tr}(J\Sigma_\delta J^\top). \tag{56}$$

Taking expectations yields

$$\begin{aligned}\mathbb{E}_\delta\tfrac{1}{2}\|H_\phi(\delta) - T_\psi(\delta)\|^2 = \tfrac{1}{2}\|H_\phi(0) - T_\psi(0)\|^2 + \tfrac{1}{2}\|J_\pi H_\phi - J_\pi T_\psi\|^2_{\Sigma_\delta} \\ + \mathcal{R}_{\mathrm{curv}} + \mathcal{O}(\mathbb{E}\|\delta\|^3).\end{aligned} \tag{57}$$

If $\|b(\delta)\| \le M_A\|\delta\|^2$, then $|\mathcal{R}_{\mathrm{curv}}| \le \frac{1}{2}M_A\|A(0)\|\mathbb{E}\|\delta\|^2$; in particular, the curvature contribution vanishes when $A(0) = 0$. □

Interpretation. The first term matches image appearance at the anchor pose. The second matches its first-order response to camera motion: parallax, silhouette displacement, and local visibility-related image changes. The additional curvature term is weighted by the anchor residual. Thus, as anchor predictions agree, nearby supervision directly matches the local pose response of the client teacher and server model.

B.7 Proof of the coverage–bias trade-off for pseudo-target supervision

Proof. The risks in Eq. (26) separate the desired poses, the training-pose mixture with true labels, and the same mixture with teacher labels. Inserting the intermediate true-label risk separates pose coverage from target reliability:

$$F_c - \widetilde{F}_c^{\gamma,r} = (F_c - F_c^{\gamma,r}) + (F_c^{\gamma,r} - \widetilde{F}_c^{\gamma,r}). \tag{58}$$

By the definitions of $\Delta^{\gamma,r}_{\mathrm{pose}}$ and $B^{\gamma,r}_{\mathrm{loss}}$, for every $\phi \in \Phi$,

$$F_c(\phi) \le \widetilde{F}_c^{\gamma,r}(\phi) + \Delta^{\gamma,r}_{\mathrm{pose}} + B^{\gamma,r}_{\mathrm{loss}}. \tag{59}$$

Pose coverage. Let d_π be the pose metric. Assume $\ell_{i,\phi}(\pi) = \ell(\hat{Y}_{i,\phi}(\pi), Y_i(\pi))$ is uniformly L_π-Lipschitz in pose and the pose distributions have finite first moments. Kantorovich–Rubinstein duality gives

$$\left|\mathbb{E}_{P_i^\star}\ell_{i,\phi} - \mathbb{E}_{\bar{Q}_i^{\gamma,r}}\ell_{i,\phi}\right| \le L_\pi W_1(P_i^\star, \bar{Q}_i^{\gamma,r}). \tag{60}$$

Taking the supremum over ϕ and the client average,

$$\Delta^{\gamma,r}_{\mathrm{pose}} \le \frac{L_\pi}{N}\sum_i W_1(P_i^\star, \bar{Q}_i^{\gamma,r}). \tag{61}$$

Mixing couplings with the same first marginal $P_i^\star$ shows that

$$W_1(P_i^\star, \bar{Q}_i^{\gamma,r}) \le (1-\gamma)W_1(P_i^\star, P_i^{\mathrm{cap}}) + \gamma W_1(P_i^\star, Q_i^r). \tag{62}$$

When Q_i^r is closer to $P_i^\star$ than P_i^{cap} in Wasserstein distance, this upper bound is smaller than the captured-only one. The coverage term therefore measures which additional camera poses the teacher makes available, separately from the quality of its rendered images.

Teacher-label bias. The captured branch uses true images and cancels exactly:

$$F_{c,i}^{\gamma,r}(\phi) - \widetilde{F}_{c,i}^{\gamma,r}(\phi) = \gamma \mathbb{E}_{Q_i^r}\big[\ell(\hat{Y}_{i,\phi}(\pi), Y_i(\pi)) - \ell(\hat{Y}_{i,\phi}(\pi), \tau_i(\pi))\big]. \tag{63}$$

Assume ℓ is uniformly L_y-Lipschitz in its target under image distance d. Then

$$|F_{c,i}^{\gamma,r}(\phi) - \widetilde{F}_{c,i}^{\gamma,r}(\phi)| \le \gamma L_y \mathbb{E}_{Q_i^r} d(\tau_i(\pi), Y_i(\pi)). \tag{64}$$

Define $e_i(\pi) = d(\tau_i(\pi), Y_i(\pi))$ and $\epsilon_i^{\mathrm{anc}} = \mathbb{E}_{\pi_a \sim A_i} e_i(\pi_a)$. Assume e_i is L_e-Lipschitz on the sampled neighborhoods. The actual sampler in Eq. (25) couples A_i and Q_i^r with $d_\pi(\pi_a, \pi) \le r_i$. Then

$$\mathbb{E}_{Q_i^r} e_i(\pi) = \mathbb{E}_{\Pi_i} e_i(\pi) \le \mathbb{E}_{\Pi_i}[e_i(\pi_a) + L_e r_i] = \epsilon_i^{\mathrm{anc}} + L_e r_i. \tag{65}$$

Substituting into Eq. (64), taking the supremum over ϕ, and averaging over clients gives

$$B_{\mathrm{loss}}^{\gamma,r} \le \frac{\gamma L_y}{N} \sum_i (\epsilon_i^{\mathrm{anc}} + L_e r_i). \tag{66}$$

This proves Eq. (13). When the anchor and captured-target laws coincide, $A_i = P_i^{\mathrm{cap}}$, the anchor error equals the captured-target average $\epsilon_i^{\mathrm{cap}} = \mathbb{E}_{P_i^{\mathrm{cap}}} e_i$. □

Interpretation. The pose term measures coverage gained from nearby cameras; the bias term measures the cost of replacing unavailable images with teacher targets. Increasing γ gives more weight to nearby coverage and teacher error. Increasing r_i broadens that coverage but raises the teacher-error bound. The two terms explain why pseudo targets complement captured images rather than replace them.

B.8 Proof of the nearby-view consistency bound

Proof. Consider two nearby cameras π_a, π_b and a pixel u whose surface point is visible in both views. Let $w_{a\to b}(u)$ be the warp induced by depth and the cameras. Write

$$Y_i(\pi_a, u) = Y_i(\pi_b, w_{a\to b}(u)) + r_{\mathrm{vis}}(u), \tag{67}$$

where r_{vis} accounts for non-Lambertian appearance, discretization, and visibility residuals. Adding and subtracting the two true images,

$$\begin{aligned}
&\|\tau_i(\pi_a, u) - \tau_i(\pi_b, w_{a\to b}(u))\| \\
&\quad \le \|\tau_i(\pi_a, u) - Y_i(\pi_a, u)\| + \|Y_i(\pi_a, u) - Y_i(\pi_b, w_{a\to b}(u))\| \\
&\qquad + \|Y_i(\pi_b, w_{a\to b}(u)) - \tau_i(\pi_b, w_{a\to b}(u))\| \\
&\quad = \|\tau_i(\pi_a, u) - Y_i(\pi_a, u)\| + \|r_{\mathrm{vis}}(u)\| \\
&\qquad + \|\tau_i(\pi_b, w_{a\to b}(u)) - Y_i(\pi_b, w_{a\to b}(u))\|.
\end{aligned} \tag{68}$$

□

Interpretation. Cross-view inconsistency is bounded by teacher rendering errors and the visibility residual. Rendering both views from the same density–radiance field exposes the scene's local view changes, rather than independent image perturbations.

B.9 Proof of the pose-sampling variance bound

Proof. Fix the server and client teacher. Let $g(\pi)$ be the true-label server gradient at $\pi \sim Q_i^r$, with $\mathbb{E} g(\pi) = g_i^Q$ and $\mathbb{E}\|g(\pi) - g_i^Q\|^2 \le \sigma_{Q,i}^2$. For K_Q independent poses, set $\widehat{g}_i^{Q,K_Q} = K_Q^{-1} \sum_k g(\pi_k)$. Independence and zero-mean centered gradients give

$$\begin{aligned}
\mathbb{E}\|\widehat{g}_i^{Q,K_Q} - g_i^Q\|^2 &= \frac{1}{K_Q^2} \sum_k \mathbb{E}\|g(\pi_k) - g_i^Q\|^2 \\
&\quad + \frac{1}{K_Q^2} \sum_{k \ne l} \mathbb{E}\langle g(\pi_k) - g_i^Q, g(\pi_l) - g_i^Q \rangle \\
&\le \frac{\sigma_{Q,i}^2}{K_Q}.
\end{aligned} \tag{69}$$

The cross terms vanish. The same calculation applies to the implemented teacher-target gradient $s_i(\pi) = J_{\phi,i}(\pi)^\top(h_\phi(\mathcal{X}_i,\pi) - \tau_i(\pi))$. Its mean is $\mu_i = g_i^Q + u_i$, where u_i is defined in Appendix B.10. If $\mathbb{E}\|s_i - \mu_i\|^2 \le \sigma_{s,i}^2$ and $\widehat{\mu}_i = K_Q^{-1}\sum_k s_i(\pi_k)$, then

$$\mathbb{E}\|\widehat{\mu}_i - \mu_i\|^2 \le \frac{\sigma_{s,i}^2}{K_Q}, \qquad \mathbb{E}\|\widehat{\mu}_i - g_i^Q\|^2 \le \frac{\sigma_{s,i}^2}{K_Q} + \|u_i\|^2. \tag{70}$$

The second inequality follows by expanding $\widehat{\mu}_i - g_i^Q = (\widehat{\mu}_i - \mu_i) + u_i$; the cross term again has zero expectation. □

Interpretation. More independent pseudo poses reduce sampling variance as $1/K_Q$. Teacher bias is controlled by teacher quality, while γ weights the pseudo branch. For fixed K_Q pseudo samples, the branch contributes at most $\gamma^2\sigma_{s,i}^2/K_Q$ sampling variance to the mixed gradient.

B.10 Proof of the pseudo-target gradient effect

Proof. Fix the model states and pose samplers. For squared image loss, define

$$\begin{aligned} g_i^\star &= \nabla_\phi \mathbb{E}_{P_i^\star}\tfrac{1}{2}\|h_\phi(\mathcal{X}_i,\pi) - Y_i(\pi)\|^2, \\ g_i^{\mathrm{cap}} &= \nabla_\phi \mathbb{E}_{P_i^{\mathrm{cap}}}\tfrac{1}{2}\|h_\phi(\mathcal{X}_i,\pi) - Y_i(\pi)\|^2, \\ g_i^Q &= \nabla_\phi \mathbb{E}_{Q_i^r}\tfrac{1}{2}\|h_\phi(\mathcal{X}_i,\pi) - Y_i(\pi)\|^2. \end{aligned} \tag{71}$$

At each pseudo pose,

$$\nabla_\phi \tfrac{1}{2}\|h_\phi - \tau_i\|^2 = \nabla_\phi \tfrac{1}{2}\|h_\phi - Y_i\|^2 + J_{\phi,i}^\top(Y_i - \tau_i). \tag{72}$$

Hence the implemented mixed gradient is

$$\begin{aligned} \widetilde{g}_i^\gamma &= (1-\gamma)g_i^{\mathrm{cap}} + \gamma(g_i^Q + u_i), \\ u_i &= \mathbb{E}_{Q_i^r} J_{\phi,i}(\pi)^\top(Y_i(\pi) - \tau_i(\pi)). \end{aligned} \tag{73}$$

With $a_i = g_i^{\mathrm{cap}} - g_i^\star$ and $c_i = g_i^Q - g_i^{\mathrm{cap}}$,

$$\widetilde{g}_i^\gamma - g_i^\star = g_i^{\mathrm{cap}} - g_i^\star + \gamma(g_i^Q - g_i^{\mathrm{cap}} + u_i) = a_i + \gamma(c_i + u_i). \tag{74}$$

Squaring gives

$$\|\widetilde{g}_i^\gamma - g_i^\star\|^2 = \|a_i\|^2 + 2\gamma\langle a_i, c_i + u_i\rangle + \gamma^2\|c_i + u_i\|^2. \tag{75}$$

For $\gamma > 0$, the error is smaller than its captured-only value precisely when

$$2\langle a_i, c_i + u_i\rangle + \gamma\|c_i + u_i\|^2 < 0. \tag{76}$$

□

Interpretation. a_i is the captured-pose bias, c_i is the nearby-pose correction, and u_i is the teacher error projected onto trainable server parameters. Pseudo targets improve the update when the correction offsets the original bias. Their weight must also account for teacher error, as the quadratic term penalizes an excessive correction. This explains the intermediate budget selected in the ablation.

B.11 Proof of update-relevant teacher error for general image losses

Proof. Fix $\mathcal{X}_i, \pi, \phi$ and set $\Gamma_\phi(Y) = \nabla_\phi \ell(h_\phi(\mathcal{X}_i,\pi), Y)$ and $e_i = \tau_i(\pi) - Y_i(\pi)$. Assume Γ_ϕ is continuously differentiable along the segment $Y_i + se_i$, $s \in [0,1]$, with derivative $A(Y) = \partial\Gamma_\phi(Y)/\partial Y$ that is $M_{\nabla Y}$-Lipschitz. The fundamental theorem of calculus gives

$$\Gamma_\phi(Y_i + e_i) - \Gamma_\phi(Y_i) = A(Y_i)e_i + \int_0^1 [A(Y_i + se_i) - A(Y_i)]e_i\, ds. \tag{77}$$

For the remainder $\mathcal{R}_i$,

$$\|\mathcal{R}_i\| \le \int_0^1 M_{\nabla Y} s\|e_i\|^2\, ds = \tfrac{1}{2}M_{\nabla Y}\|e_i\|^2. \tag{78}$$

Writing $A_{\phi,i,\pi} = A(Y_i(\pi))$, we obtain

$$\nabla_\phi \ell(h_\phi, \tau_i) - \nabla_\phi \ell(h_\phi, Y_i) = A_{\phi,i,\pi} e_i + \mathcal{O}(\|e_i\|^2). \tag{79}$$

With a uniform remainder bound and finite second moments, the captured branch cancels in the mixed objective, yielding

$$\begin{aligned} g_i^{\text{impl}} - g_i^{\text{true}} &= \gamma \mathbb{E}_{Q_i^r}[A_{\phi,i,\pi} e_i + \mathcal{R}_i], \\ \left\|\gamma \mathbb{E}_{Q_i^r} \mathcal{R}_i\right\| &\le \tfrac{1}{2}\gamma M_{\nabla Y} \mathbb{E}_{Q_i^r}\|e_i\|^2. \end{aligned} \tag{80}$$

For squared loss, $\Gamma_\phi(Y) = J_{\phi,i}^\top(h_\phi - Y)$, so $A_{\phi,i,\pi} = -J_{\phi,i}^\top$ and $\mathcal{R}_i = 0$. This recovers Proposition 3.2. □

Interpretation. Teacher error affects server updates through the mixed derivative $A_{\phi,i,\pi}$. For squared loss this is exactly the trainable server Jacobian $-J_{\phi,i}^\top$: the relevant teacher error is the component that changes the server update, not image-space error alone.

B.12 Proof of the scene-clustered sampling and local-update drift claims

Proof. Fix the teachers and samplers, and let $\widetilde{F}_i^\gamma = \widetilde{F}_{c,i}^{\gamma,r}$, $\widetilde{g}_i = \nabla \widetilde{F}_i^\gamma$, and $\widetilde{g} = N^{-1}\sum_i \widetilde{g}_i$. For $N > 1$ and $1 \le m \le N$, sample m clients uniformly without replacement. Then

$$\mathbb{E}_{\mathcal{S}_t}\left[\frac{1}{m}\sum_{i\in\mathcal{S}_t} \widetilde{g}_i\right] = \widetilde{g}. \tag{81}$$

Set $h_i = \widetilde{g}_i - \widetilde{g}$, so $\sum_i h_i = 0$. The single and pair inclusion probabilities are m/N and $m(m-1)/(N(N-1))$. Thus

$$\begin{aligned} \mathbb{E}\left\|\frac{1}{m}\sum_{i\in\mathcal{S}_t} h_i\right\|^2 &= \frac{1}{m^2}\left[\frac{m}{N}\sum_i \|h_i\|^2 + \frac{m(m-1)}{N(N-1)}\sum_{i\neq j}\langle h_i, h_j\rangle\right] \\ &= \frac{N-m}{m(N-1)}\frac{1}{N}\sum_i \|h_i\|^2, \end{aligned} \tag{82}$$

since $\sum_{i\neq j}\langle h_i, h_j\rangle = -\sum_i \|h_i\|^2$. Defining $H_{\text{scene}}(\phi) = N^{-1}\sum_i \|\widetilde{g}_i - \widetilde{g}\|^2$ gives

$$\mathbb{E}_{\mathcal{S}_t}\left\|\frac{1}{m}\sum_{i\in\mathcal{S}_t}\widetilde{g}_i - \widetilde{g}\right\|^2 = \frac{N-m}{m(N-1)} H_{\text{scene}}(\phi). \tag{83}$$

For local drift, consider gradient descent $\phi_{i,k+1} = \phi_{i,k} - \eta\widetilde{g}_i(\phi_{i,k})$, $\phi_{i,0} = \phi$. Assume bounded gradients and Hessians and a Lipschitz Hessian on a neighborhood containing the trajectories, with sufficiently small ηE. Write $\widetilde{H}_i = \nabla^2 \widetilde{F}_i^\gamma(\phi)$. Taylor expansion and the unrolled update give

$$\begin{aligned} \widetilde{g}_i(\phi_{i,k}) &= \widetilde{g}_i + \widetilde{H}_i(\phi_{i,k} - \phi) + \mathcal{O}(\|\phi_{i,k} - \phi\|^2), \\ \phi_{i,k} - \phi &= -\eta\sum_{s=0}^{k-1}\widetilde{g}_i(\phi_{i,s}) = -\eta k\widetilde{g}_i + \mathcal{O}(\eta^2 k^2), \\ \widetilde{g}_i(\phi_{i,k}) &= \widetilde{g}_i - \eta k\widetilde{H}_i\widetilde{g}_i + \mathcal{O}(\eta^2 k^2). \end{aligned} \tag{84}$$

Summing over E steps yields

$$\begin{aligned} \Delta_i = \phi_{i,E} - \phi &= -\eta\sum_{k=0}^{E-1}\widetilde{g}_i(\phi_{i,k}) \\ &= -\eta E\widetilde{g}_i + \eta^2\frac{E(E-1)}{2}\widetilde{H}_i\widetilde{g}_i + \mathcal{O}(\eta^3 E^3). \end{aligned} \tag{85}$$

Let $\widetilde{H} = N^{-1}\sum_i \widetilde{H}_i$. The difference between averaged local and centralized second-order terms is

$$\frac{1}{N}\sum_i \widetilde{H}_i\widetilde{g}_i - \widetilde{H}\widetilde{g} = \frac{1}{N}\sum_i (\widetilde{H}_i - \widetilde{H})(\widetilde{g}_i - \widetilde{g}). \tag{86}$$

□

Table 5: **Repeated MIA/PIA evaluations.** The exposure conditions match Table 3. Values are mean±standard deviation over five shadow/target-client splits and three attacker initializations per split, followed by 95% run-level bootstrap confidence intervals.

Exposed signal / server condition	MIA Acc. ↓	PIA Acc. ↓
Ours $\Delta\phi$	56.8 ± 1.4 [56.0,57.6]	31.4 ± 1.8 [30.4,32.4]
Larger trainable subset	63.1 ± 1.6 [62.2,64.0]	39.2 ± 2.0 [38.1,40.3]
Full-model update	70.4 ± 1.3 [69.7,71.1]	47.8 ± 1.7 [46.9,48.7]
Scratch server	68.9 ± 1.8 [67.9,69.9]	45.1 ± 2.1 [43.9,46.3]
Public-warmed server	60.5 ± 1.5 [59.7,61.3]	34.0 ± 1.9 [33.0,35.0]
Early federated round	58.9 ± 1.7 [58.0,59.8]	33.2 ± 1.8 [32.2,34.2]
Leaked client NeRF	93.2 ± 0.7 [92.8,93.6]	91.6 ± 0.9 [91.1,92.1]

Interpretation. With one local step, sampled clients form an unbiased scene-level minibatch. Multiple local steps introduce drift through curvature–gradient covariance: each client follows its own scene objective before aggregation. The variance identity separates client sampling from this local optimization effect, and the update expansion identifies the leading drift term.

C Privacy Boundary and Leakage Evaluations

Attacker access. The honest-but-curious server knows the architecture, training protocol, θ_{base}, ϕ^t, and each uploaded $\Delta\phi_i^t$. Images, pseudo targets, and client field parameters stay local. MIA and PIA measure identity and vehicle-type exposure in the updates; a leaked neural field provides a directly renderable positive control.

Repeated-run statistical analysis. Table 5 provides the statistical analysis of the main-paper results in Table 3. For each condition, we use five independently sampled shadow/target-client splits and three attacker initializations per split, giving 15 attack runs. The corresponding server checkpoint is fixed, so variation comes from attack-data sampling and attacker optimization rather than server retraining. We report the mean and standard deviation, followed by a 95% confidence interval obtained from 10,000 bootstrap resamples of the run-level accuracies.

C.1 Membership inference attack

Goal. MIA predicts whether a candidate car scene contributed to the target client's update. The membership unit is a scene because each client owns one car instance. Candidate negatives are other car instances with the same input-view budget, so membership cannot be inferred merely from the number of available images.

Shadow-client construction. We split ShapeNet Cars clients into shadow and target clients. For every shadow client, we run the same pipeline used for the target clients: train the private neural field, render nearby pseudo targets, locally update the server trainable subset, and store the resulting signal for the selected attack condition. The attacker is trained on shadow pairs and evaluated on disjoint target clients.

Update-based attack features. For a candidate tuple $z = (\mathcal{X}, \pi, Y)$, compute

$$g_z^t = \nabla_\phi \ell(f_{\theta_{\text{base}},\phi^t}(\mathcal{X}, \pi), Y). \tag{87}$$

The main feature is update-gradient alignment,

$$a_{\text{align}}(z, i, t) = -\frac{\langle \Delta\phi_i^t, g_z^t \rangle}{\|\Delta\phi_i^t\|_2 \|g_z^t\|_2}, \tag{88}$$

for nonzero update and candidate-gradient norms; the minus sign follows the descent direction. We also use the loss-drop feature

$$a_{\text{loss}}(z, i, t) = \ell(f_{\theta_{\text{base}},\phi^t}(\mathcal{X}, \pi), Y) - \ell(f_{\theta_{\text{base}},\phi^t+\Delta\phi_i^t}(\mathcal{X}, \pi), Y). \tag{89}$$

The alignment feature measures agreement between the candidate's descent direction and the exposed update; the loss-drop feature measures how much that update helps reconstruct the candidate. A classifier combines these signals to predict scene membership.

Attack conditions. *Ours* exposes the uploaded trainable-subset update from the fully trained server. *Larger subset* and *Full update* vary the updated parameter set. *Scratch*, *Public-warm*, and *Early FL* vary the server checkpoint using the same update-based task. For the *Leaked NeRF* control, the attacker renders the exposed field and predicts whether the rendered and candidate images depict the same scene, following the output-based setup of FedMeNF.

C.2 Property inference attack

Goal. PIA predicts vehicle type within ShapeNet Cars, as in FedMeNF (Yun et al., 2025). The attacker trains on shadow clients and is evaluated on disjoint target clients. The private label distinguishes vehicle types within the Cars category, rather than Cars from other ShapeNet categories. We use 11 vehicle types with class-balanced sampling.

Update-based PIA. Features combine layer-wise update statistics, reduced-dimensional update features, and public-probe loss drops. For a probe set $\mathcal{B}_a$ of tuples $z = (\mathcal{X}, \pi, Y)$ with vehicle type a, define

$$r_a(\Delta_i) = \frac{1}{|\mathcal{B}_a|} \sum_{(\mathcal{X},\pi,Y)\in\mathcal{B}_a} \left[\ell(f_{\phi^t}(\mathcal{X},\pi), Y) - \ell(f_{\phi^t+\Delta_i}(\mathcal{X},\pi), Y)\right]. \tag{90}$$

Here $f_\phi = f_{\theta_{\text{base}},\phi}$ and $\Delta_i = \Delta\phi_i^t$. The vector $(r_1, \ldots, r_K)$ records which public vehicle-type groups benefit from the update. Concatenating these probe responses with update statistics gives the multi-class attacker access to both parameter changes and their effect on public examples.

Leaked-NeRF positive control. For the non-transmitted *Leaked NeRF* condition, we render a fixed set of canonical views from g_{ψ_i} and train an image classifier to predict the vehicle type. This mirrors the output-based privacy evaluation style used by FedMeNF, and serves as a direct-exposure control: our protocol never uploads g_{ψ_i} or its rendered images.

Interpretation. Table 5 shows that the uploaded subset exposes less client identity and vehicle-type information than larger-subset or full-model updates. Public warm-up reduces attack accuracy relative to scratch, and the fully trained server exposes less client-specific information than the early-FL state. Leaking the client neural field produces the strongest attacks because it directly exposes rendered scene content. These comparisons distinguish residual update-level information from the much more direct exposure created by sharing a per-scene representation.

C.3 Limited-context reconstruction attack

Attack setup.

Attack setup. The attacker is given the architecture, θ_{base}, and the aggregated server state $\bar{\phi}^{t+1}$ of a late communication round, taken immediately after the round's client updates are averaged and before any further training. The target client participates in that round, so its update $\Delta\phi_i^t$ is among those just aggregated. The attacker additionally receives a limited subset of the target client's posed images and performs NVS with $f_{\theta_{\text{base}},\bar{\phi}^{t+1}}$ using them as context. This evaluates whether a server model that has just absorbed client updates recovers appearance outside the leaked views. The experiment uses 100 Objaverse client scenes and a target view at $180°$. The server receives four image–pose pairs in every condition; repeated copies fill the context when only one, two, or three distinct views are leaked. Each tuple in Table 6 lists the input azimuth angles in degrees. The limited-context configurations concentrate views in the front half of the object, whereas the $(-135°, -45°, 45°, 135°)$ control surrounds it and places cameras near the target region. The two four-view conditions therefore separate the effect of camera placement from input count.

Interpretation. The reconstruction metrics improve as the leaked context changes from a repeated single view to multiple distinct views. The surrounding-view control improves all three metrics further, despite using the same number of inputs as the local four-view condition. This comparison shows that coverage of the target region matters beyond input count. In the limited-context examples in Fig. 4, the attacker repeats or distorts visible appearance rather than recovering the target's missing details, even after incorporating the intercepted update.

Table 6: **Limited-context reconstruction attack on 100 Objaverse client scenes.** Tuples specify the four input azimuth angles; repeated angles denote duplicated leaked views. The attacker renders the 180° target using the intercepted client-updated model. The surrounding-view control improves all three metrics over the local four-view setting.

Input-view angles	PSNR	SSIM	LPIPS
$(0^\circ, 0^\circ, 0^\circ, 0^\circ)$ (1-view input)	13.901	0.542	0.432
$(-45^\circ, -45^\circ, 45^\circ, 45^\circ)$ (2-view input)	15.718	0.606	0.383
$(-90^\circ, 0^\circ, 0^\circ, 90^\circ)$ (3-view input)	18.614	0.684	0.270
$(-90^\circ, -45^\circ, 45^\circ, 90^\circ)$ (local 4-view input)	19.011	0.698	0.256
$(-135^\circ, -45^\circ, 45^\circ, 135^\circ)$ (control group)	22.389	0.775	0.159

The experiment measures reconstruction from the aggregated server state and leaked image context. It complements MIA/PIA, which measure whether an individual update reveals scene membership or vehicle type: even a model that has just aggregated client updates recovers a client's unseen appearance only to the extent that the leaked views cover it.

D Additional Ablation Protocols and Interpretation

D.1 Client-side response to server-training ablations

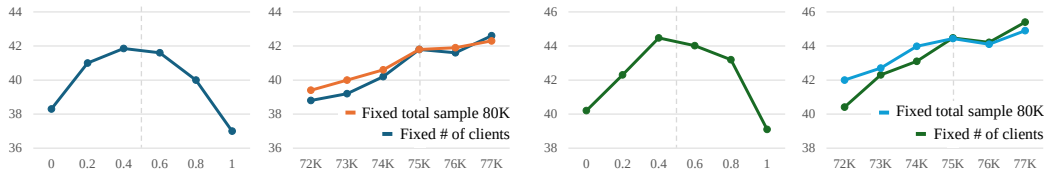


Figure 6: **Server and client ablations for pseudo-target supervision and public-data allocation.** (a–b) Server PSNR for the pseudo-target budget and public-scene count, reproduced from Fig. 5. (c–d) Client PSNR under the corresponding pseudo-target and public-data settings. The allocation study fixes either 5K clients or 80K public and client scenes in total. Both models peak at $\gamma = 0.4$; client personalization follows the server-side allocation trend.

Figure 6 extends the server ablations in Fig. 5 to client personalization. The left panels reproduce the server results, while the right panels show how the same training changes affect the personalized fields. This correspondence links the two directions of mutual reinforcement: client supervision changes the server, and the resulting global predictions guide client refinement. In Eq. (21), that dependence appears through the server error $G_i^r(\phi)$ in the client's nearby-pose neighborhood.

For the pseudo-target budget, both models peak at $\gamma = 0.4$. Too little pseudo supervision leaves captured-pose bias insufficiently corrected, while too much puts greater weight on teacher error. The effect first enters local server training and then returns to the client through the global consistency signal. Appendix D.2 connects this pattern to the coverage–bias decomposition and the quadratic gradient error.

For public-data allocation, the fixed-total setting uses $80\text{K} - |\mathcal{D}_{\text{pub}}|$ client scenes, whereas the fixed-client setting retains 5K clients. Below 75K public scenes, the fixed-total setting therefore contains more clients and yields higher client PSNR. At 75K, the allocations match and the results nearly coincide. Above this point, the fixed-5K setting retains more client scenes and shows a modest advantage; the two curves remain close. This is the same ordering described in the main-paper analysis. It shows how additional private scenes contribute to the global model and then improve personalized reconstruction, rather than benefiting only the server.

D.2 Interpretation of the pseudo-target budget γ

Figure 5(a) shows an intermediate pseudo-target optimum. At fixed ρ, the server bound separates

$$\begin{aligned}\mathcal{F}_{\rho_\star} \le \widehat{\widetilde{\mathcal{F}}}_{\rho}^{\gamma,r} + (1-\rho)\big(\Delta_{\mathrm{pose}}^{\gamma,r} + B_{\mathrm{loss}}^{\gamma,r}\big) \\ + \mathcal{C}_{\mathrm{PEFT}}(\delta) + |\rho - \rho_\star|\mathrm{Disc}_\Phi(P_0, P_c).\end{aligned} \tag{91}$$

The pose-coverage term measures the mismatch between available training cameras and the desired target poses. Nearby pseudo views improve its bound when their distribution is closer to those target poses. The teacher-label term measures the cost of replacing unavailable true images with neural-field predictions, and its bound increases with the pseudo-target weight and pose radius. Increasing γ therefore changes both where the server is supervised and how strongly it relies on the client teacher.

The gradient analysis makes this balance explicit:

$$\widetilde{g}_i^{\gamma} - g_i^{\star} = a_i + \gamma(c_i + u_i), \tag{92}$$

$$\|\widetilde{g}_i^{\gamma} - g_i^{\star}\|^2 = \|a_i\|^2 + 2\gamma\langle a_i, c_i + u_i\rangle + \gamma^2\|c_i + u_i\|^2. \tag{93}$$

Here a_i is captured-pose bias, c_i the nearby-pose correction, and u_i projected teacher error (Appendix B.10). For fixed model states, the error initially falls when $\langle a_i, c_i + u_i\rangle < 0$ and has an interior minimum when $-\|c_i + u_i\|^2 < \langle a_i, c_i + u_i\rangle < 0$. Beyond that minimum, the quadratic term increases the error. The coverage analysis identifies the competing sources of supervision, while the gradient expansion explains how their directions can produce an interior optimum. The numerical optimum depends on the client teachers, pose distributions, and server state. Our ablation selects $\gamma = 0.4$, which is then fixed across clients and rounds.

E Broader Impacts

Our framework trains large NVS models from distributed client scenes and improves personalized reconstruction without centrally collecting private multi-view images. This setting is relevant when data belong to different users or organizations, or when transferring raw scene observations is undesirable. A shared NVS model supplies guidance to clients with sparse captures while their images and renderable scene representations remain local. Potential applications include AR/VR content creation, telepresence, and product visualization.

The same reconstruction capabilities can be misused for unauthorized modeling of private spaces or objects, or for creating misleading visual content. The protocol removes direct transmission of client images and neural fields, while our attack evaluations examine information exposed by parameter updates and leaked context. Deployment should therefore include consent and ownership checks, access controls, and privacy auditing, with secure aggregation or differential privacy where appropriate.

Public-data bias and uneven client participation can also affect reconstruction quality across object types, environments, and capture conditions. Evaluating these groups separately is important when deploying a global model to a heterogeneous client population. The local fitting and NVS optimization costs should likewise be considered when choosing client hardware and participation schedules.