FEDFIELDS: LEARNING SHARED PRIORS FOR FAST IMPLICIT NEURAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Implicit Neural Representations (INRs) model signals as continuous coordinate-based functions, but typically require a separate network to be optimized for every signal. Amortized approaches avoid costly per-signal fitting by learning across many signals to directly predict signal-specific representations. Existing methods, however, assume centralized training on pooled data, limiting their applicability when signals are distributed across multiple clients. Federated learning provides a natural framework for learning across such decentralized data, yet existing federated neural field methods retain per-signal optimization: they either federate a single field for one scene or meta-learn an initialization that must still be optimized for each new signal. We introduce **FEDFIELDS**, a federated amortized neural field that learns a shared representation prior across decentralized signals. A shared encoder maps each signal to a latent representation, from which a lightweight per-point hypernetwork generates modulations for a shared INR decoder. Once trained, the model represents unseen signals in a single forward pass, with optional latent-only adaptation when higher reconstruction fidelity is desired. The shared model is learned collaboratively across clients through federated aggregation, while both the signals and their latent representations remain local to each client. Experiments show that **FEDFIELDS** achieves high-fidelity reconstruction and generalizes to unseen data distributions, while remaining effective across varying client configurations, data imbalance, and aggregation rules. The same formulation extends from images to videos, 3-D occupancy, and audio, demonstrating that amortized neural fields can be learned effectively across decentralized data.

1 INTRODUCTION

Implicit Neural Representations (INRs), or neural fields, represent signals as continuous functions that map coordinates to signal values, typically parameterized by multilayer perceptrons (Sitzmann et al., 2020b; Tancik et al., 2020; Essakine et al., 2024; Jayasundara & Patel, 2026). Their continuous, resolution-independent formulation provides a flexible representation for images, audio, video, 3-D geometry, and radiance fields (Xie et al., 2022). Conventional INRs, however, are learned on a per-signal basis. Representing a new image, waveform, or scene requires optimizing a separate coordinate network, often for hundreds or thousands of gradient steps. Architectures such as SIREN (Sitzmann et al., 2020b), WIRE (Saragadam et al., 2023), and FINER (Liu et al., 2024) improve the fidelity and efficiency of this optimization, but the underlying procedure remains unchanged, as each new signal requires a separate optimization process before it can be represented.

Amortized neural fields take a different approach. Instead of treating every signal as an independent optimization problem, they learn shared structure across a collection of signals and use it to infer a representation for a new signal (Dupont et al., 2022a; Chen & Wang, 2022; Kim et al., 2023). Depending on the formulation, an encoder may infer a signal-specific latent representation that conditions a shared coordinate network, while a hypernetwork may directly generate some or all of the coordinate-network parameters. The cost of learning to represent signals is therefore amortized over the training distribution, allowing an unseen signal to be represented with little or no gradient-based fitting once the shared model has been learned. This shifts neural fields from fitting an independent function to each signal toward learning reusable structure that supports efficient representation of new signals. Existing amortized neural fields, however, have largely been developed in centralized

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.

settings, where the signals used to learn the shared representation model are collected into a pooled training set (Chen & Wang, 2022; Kim et al., 2023; Chen et al., 2024; Szatkowski et al., 2023). In many applications, these signals are instead distributed across clients, devices, or institutions, where pooling the underlying data may be impractical or undesirable, particularly when the signals contain sensitive information. Federated learning (FL) (McMahan et al., 2017) provides a natural framework for this setting by enabling clients to collaboratively learn shared model parameters while keeping their raw signals local.

Existing federated neural-field methods have begun to extend beyond learning individual signal representations, but still rely on optimization to represent new signals. Early approaches distribute the optimization of a single neural field across clients (Holden et al., 2023; Suzuki, 2023), producing a representation specific to the scene being optimized. FedMeNF (Yun et al., 2025) goes further by federating a meta-learned initialization (Tancik et al., 2021), from which a coordinate network can be adapted to a new signal in fewer optimization steps. While this enables generalization across signals and reduces the cost of fitting, each new signal still requires optimization. Methods that avoid per-signal optimization by directly predicting signal-specific representations, however, have been developed primarily with centrally pooled training data. *This raises a broader question: can neural fields learn high-quality, generalizable representations directly from decentralized data while enabling new signals to be represented without per-signal optimization?*

We answer this question with **FedFields**, a federated neural-field framework that learns a shared representation model directly from decentralized signals. Rather than federating individual coordinate networks or a meta-learned initialization that must subsequently be optimized for each new signal, clients collaboratively learn a model that directly maps an input signal to its neural-field representation. An encoder first maps each signal to a latent representation, from which coordinate-specific features are sampled and transformed by a lightweight hypernetwork into modulations that condition a shared neural-field decoder. The shared model is trained through federated aggregation, while both the signals and their latent representations remain local to each client. Once learned, **FedFields** can represent an unseen signal directly, without per-signal optimization, while optionally allowing lightweight adaptation when additional fidelity is desired. Fig. 1 illustrates this quality–cost trade-off on a Kodak image (Eastman Kodak Company, 1993): **FedFields** achieves high reconstruction quality with zero test-time optimization, while optional adaptation further improves fidelity. We evaluate this capability across diverse federated settings and data distributions, and extend it beyond images to video, 3-D occupancy, and audio, demonstrating a general framework for learning neural-field representations from decentralized data.

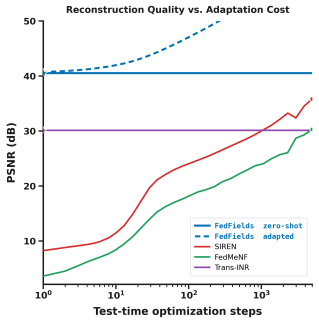


Figure 1: **High-fidelity reconstruction without per-signal optimization.**

Our contributions are:

1. **Federated amortized neural fields:** We introduce **FedFields**, a framework for learning amortized neural fields from decentralized data. It federatively learns the shared, signal-agnostic components; the encoder, hypernetwork, and modulated decoder; while signal-specific latent representations are inferred and retained locally. This enables amortized inference without centralized data pooling or per-signal optimization.

2. **Zero-shot neural-field inference:** **FedFields** reconstructs unseen signals directly, without per-signal optimization at test time. On Kodak, it achieves 39.26,dB PSNR, outperforming the federated meta-learning baseline (Yun et al., 2025), which requires 500 adaptation steps for each unseen signal.

3. **Privacy by design:** By keeping each signal's latent representation on the client and sharing only signal-agnostic weights, **FedFields** prevents private signals from being reconstructed from client updates, whereas the uploads of meta-learned federated neural fields reveal them directly.

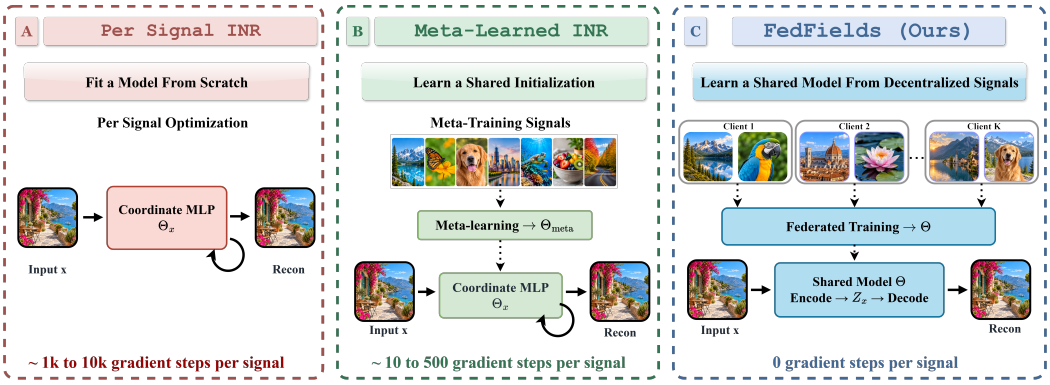


Figure 2: **Comparison of neural-field learning paradigms.** Per-signal INRs fit each signal from scratch, while meta-learning reduces this cost through a shared initialization but still requires adaptation. **FedFields** instead learns a shared model from decentralized signals, enabling direct representation without per-signal optimization.

2 Related Works

Implicit Neural Representations: INRs represent signals as continuous functions over their coordinates, providing a flexible representation that has been applied across diverse modalities, including images, audio, video, 3-D geometry, and radiance fields (Park et al., 2019; Mildenhall et al., 2021; Chen et al., 2021; 2023; Dupont et al., 2022b). A central challenge is the spectral bias of standard MLPs toward low-frequency functions, which limits their ability to reproduce fine signal detail (Rahaman et al., 2019). Fourier features (Tancik et al., 2020) address this limitation through high-frequency coordinate embeddings, while SIREN (Sitzmann et al., 2020b) uses periodic activations to represent high-frequency signals and their derivatives. More recent architectures further improve expressivity through alternative activation functions and bases (Jayasundara & Patel, 2026), including complex Gabor wavelets in WIRE (Saragadam et al., 2023) and variable-periodic activations in FINER (Liu et al., 2024). Despite these advances, most conventional INRs follow a per-signal fitting paradigm (Sitzmann et al., 2020b; Tancik et al., 2020; Saragadam et al., 2023; Liu et al., 2024; Gao et al., 2026; Jayasundara et al., 2025): given a new signal, a separate neural field is optimized specifically for that signal (See Fig. 2 A). Consequently, the cost of obtaining the representation is paid again for every new input, often requiring thousands of gradient steps per signal.

Learning across signals: The cost of per-signal optimization has motivated approaches that leverage shared knowledge across signals rather than learning each representation independently. Meta-learning (Vilalta & Drissi, 2002) provides one approach by learning parameters that can be rapidly adapted to a new signal. Learned initialization methods (Tancik et al., 2021), and MetaSDF (Sitzmann et al., 2020a), for example, learn from a distribution of signals so that a new neural field can be fitted in substantially fewer gradient steps than training from scratch. Related approaches such as Functa (Dupont et al., 2022a) and COIN++ (Dupont et al., 2022b) share a coordinate network across signals and optimize only compact signal-specific modulations. These approaches substantially reduce the cost of fitting, but representing an unseen signal still requires signal-specific optimization (See Fig. 2 B). A complementary direction uses direct amortized inference, where a learned mapping predicts the representation of a new signal directly from its observations. Hypernetwork-based INRs, for example, generate the weights and biases of a signal-specific coordinate network (Chen & Wang, 2022; Szatkowski et al., 2023; Wu et al., 2023; Gordon et al., 2024; Thatipelli & Siahkoohi, 2026). These approaches, however, have largely assumed centralized training, where signals are pooled to learn the shared inference model. Such pooling may be impractical when data are distributed across users or institutions, particularly for sensitive signals that are preferable to keep local.

Neural fields under federation. When signals cannot be centrally pooled, federated learning enables learning from distributed data while keeping the underlying signals local. Early federated neural-field methods focus on collaborative scene reconstruction, distributing observations of a scene or large-scale environment across clients and jointly optimizing the corresponding NeRF representation (Holden et al., 2023; Suzuki, 2023). The resulting representation is tied to the re-

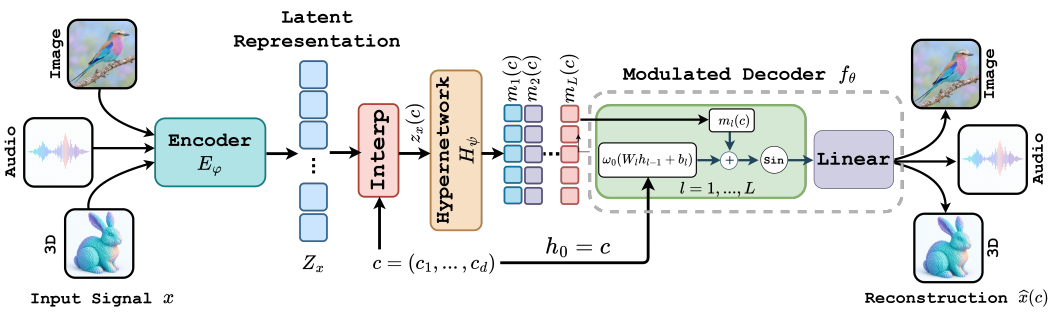


Figure 3: **Amortized neural field.** A modality-specific encoder E_φ maps a signal x to a latent representation Z_x. For a query coordinate c, the interpolated feature $z_x(c)$ is mapped by the hypernetwork H_ψ to layer-wise shift modulations $\{m_l(c)\}_{l=1}^{L}$, which condition the shared modulated decoder f_θ to predict $\hat{x}(c)$. A new signal is thus represented by a single encoder pass, without per-signal optimization.

constructed environment rather than amortized across signals for inference on unseen data. FedMeNF (Yun et al., 2025) extends this setting to multiple signals by federatively meta-learning a global initialization that accelerates adaptation to unseen signals. However, each new signal still requires signal-specific optimization, and the updates transmitted to the server are derived from networks fitted directly to a client's signals, and can therefore reveal their content. Moreover, direct amortized inference remains largely unexplored under federation: instead of federating a field or an initialization for subsequent optimization, the inference model itself can be learned across decentralized signals to directly represent unseen signals. This avoids both centralized pooling and per-signal optimization (Fig. 2 C), while keeping signal-specific representations local and communicating only shared, signal-agnostic parameters. These considerations motivate the federated formulation of amortized neural fields developed next.

3 Method

3.1 Problem Setup

We consider K clients, each holding a local dataset D_k of signals, with no signal shared across clients. A signal $x \in D_k$ is defined over a coordinate domain $\Omega = [-1, 1]^d$ as $x : \Omega \to \mathbb{R}^o$, where a coordinate $c \in \Omega$ maps to its corresponding signal value $x(c)$. For example, $(d, o) = (2, 3)$ for RGB images, $(1, 1)$ for audio, and $(3, 1)$ for occupancy fields. Our goal is to learn, from these decentralized datasets, a shared model that can directly infer a neural-field representation for a previously unseen signal. We parameterize the shared model by $\Theta = \{\varphi, \psi, \theta\}$, comprising an encoder E_φ, a hypernetwork H_ψ, and a modulated coordinate decoder f_θ. As illustrated in Fig. 3, given a signal x, the encoder first produces a signal-specific latent representation $Z_x = E_\varphi(x)$. For each query coordinate $c \in \Omega$, we interpolate Z_x at c to obtain the local feature $z_x(c) = \text{interp}(Z_x, c)$. The hypernetwork then maps this feature to a set of layer-wise decoder modulations $\mathbf{m}(c) = \{m_\ell(c)\}_{\ell=1}^{L} = H_\psi(z_x(c))$, which condition the shared decoder as it maps the coordinate c to the predicted signal value $\hat{x}(c) = f_\theta(c; \mathbf{m}(c))$. During federated training, each client locally optimizes Θ using its reconstruction objective $\mathcal{L}_k$, and the

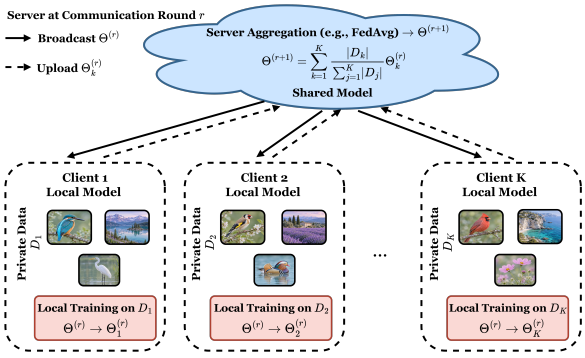


Figure 4: **Federated training.** Clients update the broadcast parameters $\Theta^{(r)}$ locally; the server aggregates $\Theta_k^{(r)}$ into $\Theta^{(r+1)}$. Signals and latents remain local.

server aggregates the resulting parameters using dataset-size weights $w_k = |D_k| / \sum_{j=1}^{K} |D_j|$ (See Fig. 4) . Only Θ is communicated across clients; the signal-specific representation Z_x remains local.

3.2 Amortized Neural Field

Our architecture separates information that is shared across signals from the representation specific to an individual signal. As illustrated in Fig. 3, an input signal is first encoded into a latent representation. For each query coordinate, a local feature sampled from this grid is converted into layer-wise modulations that condition a shared continuous decoder.

Encoder and Latent Representation: Given a signal x, the encoder produces a latent representation $Z_x = E_\varphi(x)$. For a query coordinate c, the corresponding local feature is obtained by d-linear interpolation, $z_x(c) = \mathrm{interp}(Z_x, c)$. For images, E_φ combines a lightweight convolutional branch that captures fine-scale spatial structure with features from a pretrained DINOv2 ViT-B/14 backbone (Oquab et al., 2023), providing complementary higher-level context. The DINOv2 backbone remains frozen, with rank-8 LoRA (Hu et al., 2021) applied to the attention projections of its last eight transformer blocks. The resulting DINOv2 features are resized to the latent spatial resolution and projected to a lower-dimensional feature space using a 1×1 convolution before being concatenated with the convolutional features to form Z_x. Accordingly, for images, the trainable encoder parameters are $\varphi = \{\varphi_{\mathrm{conv}}, \varphi_{\mathrm{LoRA}}, \varphi_{\mathrm{proj}}\}$, corresponding to the convolutional branch, LoRA factors, and feature projection, respectively. The pretrained DINOv2 weights remain fixed and are therefore excluded from φ. Further architectural and LoRA details, as well as modality-specific encoders for other signal domains, are provided in the supplementary material.

Hypernetwork: The latent representation provides a spatial representation of the input, while the neural field must produce signal values continuously at arbitrary coordinates. We therefore condition the decoder locally. For each query c, a lightweight hypernetwork maps the interpolated feature $z_x(c)$ to a collection of layer-wise modulation vectors, $m(c) = H_\psi(z_x(c)) = \{m_l(c)\}_{l=1}^{L}$, where $m_l(c) \in \mathbb{R}^n$ conditions the l-th decoder layer and n is the number of hidden units per decoder layer. Unlike hypernetworks that generate the weights of an entire signal-specific coordinate network (Chen & Wang, 2022), H_ψ predicts only these modulation vectors, while the decoder parameters remain shared across all signals. The output layer of H_ψ is zero-initialized, such that $m(c) = 0$ at initialization and signal-dependent conditioning is learned as a residual modification of the shared decoder.

Decorder: For images, occupancy fields, and audio, we use a shared SIREN decoder (Sitzmann et al., 2020b) with shift modulation (Mehta et al., 2021; Dupont et al., 2022a). Given the layer-wise modulations $\{m_l(c)\}_{l=1}^{L}$ produced by the hypernetwork, the decoder starts from $h_0 = c$ and computes $h_l = \sin(\omega_0(W_l h_{l-1} + b_l) + m_l(c))$ for $l = 1, \ldots, L$, followed by the linear output $\hat{x}(c) = W_{\mathrm{out}} h_L + b_{\mathrm{out}}$. Thus, the shared weights $\{W_l, b_l\}$ define the common coordinate decoder, while $m_l(c)$ provides signal- and coordinate-specific conditioning at each layer. We use $L = 6$ layers of width $n = 256$ with SIREN initialization (Sitzmann et al., 2020b); the modality-specific frequency ω_0 and output dimension are given in the supplementary material. For video, where per-pixel coordinate decoding is computationally expensive, we instead use an HNeRV-style convolutional frame decoder (Chen et al., 2023) conditioned on a per-frame spatial embedding.

3.3 Federated Training and Inference

Federated training proceeds for R communication rounds across K clients, where client k holds a private dataset D_k. At round r, the server broadcasts the current shared parameters $\Theta^{(r)}$ to the participating clients (See Fig. 4). Each client initializes a local copy from $\Theta^{(r)}$ and performs τ optimization steps to minimize $\mathcal{L}_k(\Theta) = \mathbb{E}_{x\sim D_k,\, c\sim\pi_x}\big[\rho(\hat{x}(c), x(c))\big]$, where π_x is a modality-specific sampling distribution over Ω. The per-point loss ρ depends on the signal domain. For images and audio, we use the mean squared error $\rho(\hat{x}(c), x(c)) = \frac{1}{o}\|\hat{x}(c) - x(c)\|_2^2$, where o is the output dimension. For 3-D occupancy, where $x(c) \in \{0, 1\}$ and $\hat{x}(c)$ is an occupancy logit, we use binary cross-entropy, $\rho(\hat{x}(c), x(c)) = -x(c)\log\sigma(\hat{x}(c)) - (1-x(c))\log(1-\sigma(\hat{x}(c)))$, with σ is the sigmoid function. After local training, client k returns the updated parameters $\Theta_k^{(r)}$, and the server

aggregates them into $\Theta^{(r+1)}$ (See Fig. 4). Unless otherwise stated, we use FedAvg (McMahan et al., 2017), $\Theta^{(r+1)} = \sum_{k=1}^{K} w_k \Theta_k^{(r)}$ with $w_k = |D_k| / \sum_{j=1}^{K} |D_j|$; alternative aggregation rules are evaluated in the supplementary material. At inference, a new signal is encoded in a single forward pass, $Z_x = E_\varphi(x)$, and decoded at any query coordinate. Optionally, Z_x can be refined by a few gradient steps on the reconstruction loss while all shared parameters Θ stay frozen; only the per-signal latent changes.

3.4 Extension to Other Modalities

The framework extends beyond images by adapting the architecture to the structure of each signal domain. Audio and occupancy fields retain the same latent-conditioning formulation and modulated coordinate decoder, requiring only modality-specific encoders and interpolation schemes. For **audio**, a 1-D convolutional encoder maps the waveform to a temporal latent representation, which is queried by linear interpolation. For **occupancy fields**, a 3-D convolutional encoder produces a volumetric latent representation queried by trilinear interpolation; the decoder predicts occupancy logits and is trained with binary cross-entropy, with additional point samples drawn near object boundaries. Both modalities use the same decoder configuration as images (L=6, n=256, ω_0=30), but without a pretrained backbone, so all encoder parameters are included in Θ. **Video** often requires a different architecture due to its spatiotemporal structure Chen et al. (2021). We adopt an HNeRV-style convolutional frame codec (Chen et al., 2021; 2023), where an encoder produces a per-frame spatial embedding and a pixel-shuffle decoder reconstructs the corresponding frame. Thus, unlike the other modalities, video does not use coordinate inputs, a modulation hypernetwork, or a SIREN decoder; instead, the complete frame-codec parameters are trained federatively. Detailed architectures and modality-specific configurations are provided in the supplementary material.

4 Experiments

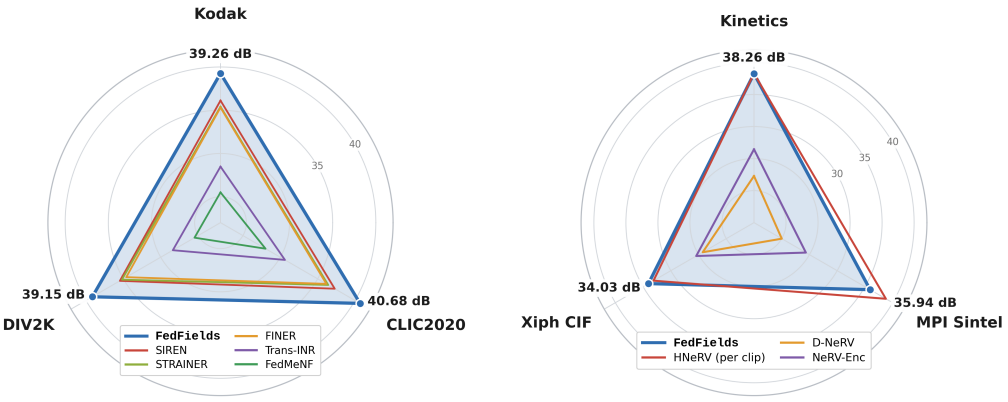
4.1 Data and Experimental Setup

We federate training across $K = 10$ clients for 300 communication rounds, with 100 local steps per client per round, with the training data partitioned uniformly at random across clients. For images, we train on 238,545 images pooled from LSDIR (Li et al., 2023), COCO-2017 (Lin et al., 2014), CelebA-HQ (Karras et al., 2017), DTD (Cimpoi et al., 2014), Flickr2K (Timofte et al., 2017), and the DIV2K training set (Agustsson & Timofte, 2017), and evaluate on Kodak (Eastman Kodak Company, 1993), the CLIC2020 validation set (Toderici et al., 2020), and DIV2K-val (Agustsson & Timofte, 2017). For video, we train on clips from DAVIS (Pont-Tuset et al., 2017) and UCF-101 (Soomro et al., 2012), and evaluate on Kinetics (Kay et al., 2017), MPI Sintel (Butler et al., 2012), and the Xiph CIF test sequences (Montgomery & Lars, 1994). For audio, we train on LibriSpeech dev-clean (Panayotov et al., 2015) and evaluate on test-clean, whose speakers are disjoint from the training set. For 3-D occupancy, we train on the official training split of ModelNet40 (Wu et al., 2015), and evaluate on its test split as well as on Google Scanned Objects (Downs et al., 2022) and Pix3D (Sun et al., 2018). All experiments are conducted on an NVIDIA RTX A5000 GPUs with 24 GB of memory. Dataset splits, preprocessing, resolutions, and modality-specific sampling and training settings are provided in the supplementary material.

4.2 Results

4.2.1 Image Representation

We first evaluate whether a federatively learned amortized neural field can represent unseen images without per-signal optimization. Fig. 6 (a) summarizes reconstruction quality on Kodak, CLIC2020, and DIV2K-val, where **FedFields** achieves 39.26, 40.68, and 39.15 dB, respectively, from a single forward pass. We compare against methods spanning different representation and inference regimes. FedMeNF (Yun et al., 2025) provides the closest federated comparison: while it also learns across decentralized clients, it meta-learns an initialization that must be optimized for each unseen image. Even after 500 test-time steps, FedMeNF remains 13.76, 12.65, and 13.67 dB below the zero-shot performance of **FedFields**. Trans-INR (Chen & Wang, 2022) instead shares our zero-shot inference setting, directly predicting a neural field for an unseen image, but is trained



(a) Image reconstruction (b) Video reconstruction

Figure 6: **Reconstruction quality on image and video benchmarks.** PSNR (dB) on (a) Kodak, CLIC2020, and DIV2K-val and (b) Kinetics, MPI Sintel, and Xiph CIF; vertex values give the PSNR of **FEDFIELDS**.

centrally; **FEDFIELDS** improves upon it by 10.78, 10.06, and 10.77 dB despite being learned from decentralized data. STRAINER (Vyas et al., 2025) transfers shared parameters to unseen signals but still requires per-image optimization, with **FEDFIELDS** exceeding its reconstruction quality by 3.85, 4.27, and 3.96 dB without any test-time fitting.

We further compare against conventional per-signal INRs, which optimize a separate coordinate network for every test image. Despite this additional test-time optimization, SIREN (Sitzmann et al., 2020b) falls 3.14, 3.40, and 3.69 dB below **FEDFIELDS** on the three datasets, while FINER (Liu et al., 2024) falls 3.95, 4.52, and 4.54 dB below. Thus, the federatively learned representation not only enables direct zero-shot inference on unseen images, but also surpasses methods that fit a separate representation to each test signal. Fig. 5 shows representative qualitative results on Kodak. Additional qualitative results, exact results for all methods, and further experimental details are provided in the supplementary material.



Figure 5: **Reconstruction of a held-out Kodak image.** In a single forward pass, **FEDFIELDS** recovers fine detail such as the lettering on the raft (inset), exceeding per-image SIREN and STRAINER fits and far surpassing FedMeNF and Trans-INR.

4.2.2 VIDEO REPRESENTATION

We next examine whether the federated amortized formulation extends to video. Fig. 6 (b) summarizes reconstruction quality on Kinetics, MPI Sintel, and Xiph CIF, where **FEDFIELDS** achieves 38.26, 35.94, and 34.03 dB, respectively. As with images, we compare against methods that obtain representations for unseen videos under different inference regimes. NeRV-Enc (Chen et al., 2024) and D-NeRV (He et al., 2023) learn across collections of videos and infer representations for unseen clips without optimizing a new network from scratch. **FEDFIELDS**



Figure 7: **Video reconstruction on Xiph CIF.**

outperforms NeRV-Enc by 11.81, 11.59, and 8.61 dB on Kinetics, MPI Sintel, and Xiph CIF, respectively, and D-NeRV by 15.98, 15.93, and 9.78 dB. Notably, D-NeRV additionally receives two reconstructed keyframes for each 8-frame sub-clip, yet its performance drops to 20.01 dB on MPI Sintel, while **FEDFIELDS** maintains above 34 dB across all three benchmarks.

We further compare against HNeRV (Chen et al., 2023), which fits a separate network to each test video rather than directly inferring its representation. Despite HNeRV's per-video optimization, **FEDFIELDS** achieves comparable PSNR on Kinetics (38.26 vs. 38.30 dB) and surpasses it by 0.91 dB on Xiph CIF. MPI Sintel is the only benchmark with a notable gap, where HNeRV reaches 38.77 dB compared with 35.94 dB for **FEDFIELDS**. Nevertheless, these results show that a video representation learned from decentralized training data can transfer directly to unseen videos while remaining competitive with methods that optimize a separate representation for each test clip. Fig. 7 shows a representative reconstruction on a Xiph CIF sequence, where **FEDFIELDS** closely follows the ground truth even under fast motion. The evaluation protocol, exact results for all methods, qualitative results, and further experimental details are provided in the supplementary material.

4.2.3 OCCUPANCY FIELDS

We next examine whether the federated amortized formulation transfers to 3-D occupancy fields. As shown in Table 1, **FEDFIELDS** achieves 0.9950, 0.9955, and 0.9945 IoU on ModelNet40, GSO, and Pix3D, respectively. ConvONet (Peng et al., 2020) and GenSDF (Chou et al., 2022) provide comparisons with learned shape priors that transfer across shapes. As can be seen, **FEDFIELDS** exceeds ConvONet by 0.28, 0.15, and 0.27 IoU and GenSDF by 0.20, 0.31, and 0.18 IoU on ModelNet40, GSO, and Pix3D, respectively.

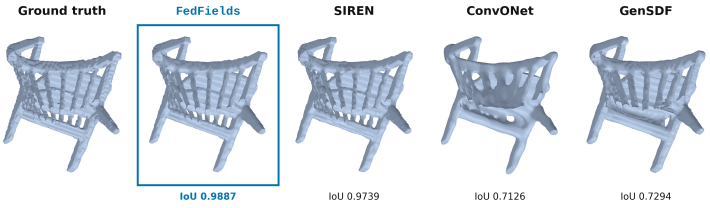


Figure 8: **Occupancy reconstruction of an unseen Pix3D chair** (IoU below each result). **FEDFIELDS** preserves the thin, separate slats of the backrest and legs, which the baselines blur or merge.

The difference is also evident qualitatively in Fig. 8: on the unseen Pix3D chair, the baseline reconstructions lose or merge thin structures in the backrest, whereas **FEDFIELDS** better preserves the individual slats and legs.

Table 1: **Occupancy reconstruction** (IoU ↑) on test shapes

Method	ModelNet40	GSO	Pix3D
ConvONet	0.7120	0.8409	0.7247
GenSDF	0.7969	0.6861	0.8102
SIREN	0.9946	0.9923	0.9934
FEDFIELDS	**0.9950**	**0.9955**	**0.9945**

We further compare against SIREN (Sitzmann et al., 2020b), which optimizes a separate coordinate network for each test shape and provides the strongest per-signal baseline, reaching 0.992–0.995 IoU across the three datasets. Without fitting a separate network to each shape, **FEDFIELDS** slightly exceeds SIREN on all three benchmarks, with the largest difference on the unseen GSO dataset (0.9955 vs. 0.9923 IoU). On the Pix3D example in Fig. 8, this difference is reflected in the recovery of thin repeated structures (0.989 vs. 0.974 IoU). Overall, the results show that the representation learned from decentralized ModelNet40 data transfers directly to unseen shapes, including shapes from datasets not observed during training. The evaluation protocol, additional qualitative results, and further experimental details are provided in the supplementary material.

4.3 PRIVACY OF CLIENT UPDATES

Federated learning keeps the signals locally with the clients, but the transmitted updates are computed from them and may reveal their content. The risk is acute for neural fields: in meta-learned approaches such as FedMeNF (Yun et al., 2025), each client fits a coordinate network to its own signals, so the transmitted weights directly encode them. In **FEDFIELDS**, signal content resides in the latent representation Z_x, which is computed and kept with the client; only the shared encoder, hypernetwork, and decoder are transmitted. We test whether

this separation prevents reconstruction of private signals, comparing against federated Reptile (Nichol et al., 2018) as a representative meta-learned federated neural field. The attacker knows the architecture and the broadcast weights and targets six images unseen during training. We measure the PSNR between the attacker's reconstruction and the private image; as a reference, guessing without any information from the update scores 8.2 dB with random noise and 11.7 dB with a flat grey image (See Supplementary).

Table 2: **Reconstructing private images from client updates** (mean PSNR in dB; lower is more private).

Attack	Reptile	FEDFIELDS
Gradient inversion	21.01	**5.43**
Weight readout	25.51	**11.75**
before the update	11.56	11.73

Gradient inversion. We apply the standard gradient-matching attack (Zhu et al., 2019; Geiping et al., 2020), which optimizes a dummy image until its gradient matches an intercepted one, to a gradient computed from a single image, the setting most favorable to the attacker. Reptile's gradients yield recognizable images (21.01 dB), whereas ours yield only 5.43 dB, below even the random-noise guess (See Table 2).

Weight readout. Since gradient inversion may fail simply when its optimization fails, we also attack the uploaded weights directly, after the client's full local training. A Reptile upload is itself a coordinate network and can simply be rendered, reproducing the private image at 25.51 dB. A FEDFIELDS upload cannot render anything without the client's latent representation, so the attacker instead passes images it has, such as public photos, a grey image, or noise, through the updated model. The best of these scores 11.75 dB, the same as on the weights before the update (11.73 dB, See Table 2): the update adds no recoverable content. Fig. 9 provides a qualitative example, with additional details and results reported in the supplementary material.

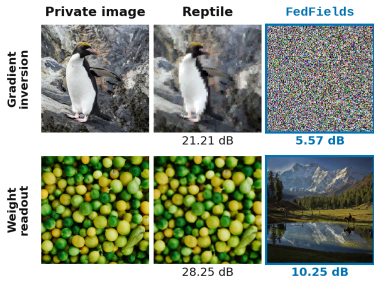


Figure 9: **What an attacker recovers from client updates.**

4.4 ADDITIONAL EXPERIMENTS

The supplementary material provides detailed experimental setups and comprehensive quantitative and qualitative results for all evaluated modalities, including images, video, 3D occupancy, and audio. It further reports transfer beyond the training domain to medical and hyperspectral image reconstruction, architectural ablations isolating each component of the encoder and latent representation, federated optimization ablations comparing FedAvg, FedNova, FedExp, and FedACG under equal budget, and full privacy attack protocols with additional reconstructions.

5 CONCLUSION

We presented FEDFIELDS, a federated framework that learns a single amortized neural field across clients without sharing their data. Its central design choice is to separate what is shared from what is private: an encoder, hypernetwork, and modulated decoder are trained collaboratively, while each signal is represented by a latent representation that is computed and kept on its client. This separation allows the shared model to represent new signals without fitting a dedicated network to each of them, and keeps their content out of every transmitted update. Across modalities, the federated model proves a strong and general representation of unseen signals. On images, a single forward pass outperforms per-image SIREN, FINER, and STRAINER fits by more than 3 dB and the federated meta-learning approach FedMeNF by over 12 dB. The same formulation surpasses HNeRV on natural video and reaches 0.995 IoU on occupancy fields for shape datasets never seen during training. At the same time, neither gradient inversion nor a direct readout of the uploaded weights recovers private signals from our updates, whereas the uploads of a meta-learned federated neural field reveal them. Together, these results show that amortized neural fields offer a practical path to learning continuous signal representations from data that cannot leave its source.

6 AI USE STATEMENT

In this work, we used generative AI tools for grammar correction and writing enhancement, as well as to assist in generating some methodological figures. All AI-assisted text and figures were carefully reviewed and verified by the authors for accuracy and consistency with the proposed methodology. Generative AI was not used for research ideation, methodology development, data generation or analysis, experimental design, or the formulation of scientific claims and conclusions. We take full responsibility for the final content of this work, including text and figures produced with the assistance of generative AI.

REFERENCES

Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 1122–1131. IEEE, 2017.

Ujjwal Baid, Satyam Ghodasara, Suyash Mohan, Michel Bilello, Evan Calabrese, Errol Colak, Keyvan Farahani, Jayashree Kalpathy-Cramer, Felipe C Kitamura, Sarthak Pati, et al. The rsna-asnr-miccai brats 2021 benchmark on brain tumor segmentation and radiogenomic classification. *arXiv preprint arXiv:2107.02314*, 2021.

Spyridon Bakas, Hamed Akbari, Aristeidis Sotiras, Michel Bilello, Martin Rozycki, Justin S Kirby, John B Freymann, Keyvan Farahani, and Christos Davatzikos. Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features. *Scientific data*, 4 (1):170117, 2017.

Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pp. 611–625. Springer, 2012.

Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. *Advances in Neural Information Processing Systems*, 34:21557–21568, 2021.

Hao Chen, Matthew Gwilliam, Ser-Nam Lim, and Abhinav Shrivastava. Hnerv: A hybrid neural representation for videos. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10270–10279. IEEE, 2023.

Hao Chen, Saining Xie, Ser-Nam Lim, and Abhinav Shrivastava. Fast encoding and decoding for implicit video representation. In *European conference on computer vision*, pp. 402–418. Springer, 2024.

Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pp. 170–187. Springer, 2022.

Gene Chou, Ilya Chugunov, and Felix Heide. Gensdf: Two-stage learning of generalizable signed distance functions. *Advances in Neural Information Processing Systems*, 35:24905–24919, 2022.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3606–3613, 2014.

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 *2022 international conference on robotics and automation (ICRA)*, pp. 2553–2560. Ieee, 2022.

Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you can treat it like one. *arXiv preprint arXiv:2201.12204*, 2022a.

Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goliński, Yee Whye Teh, and Arnaud Doucet. Coin++: Neural compression across modalities. *arXiv preprint arXiv:2201.12904*, 2022b.

Eastman Kodak Company. Kodak lossless true color image suite. `http://r0k.us/graphics/kodak/`, 1993.

Amer Essakine, Yanqi Cheng, Chun-Wun Cheng, Lipei Zhang, Zhongying Deng, Lei Zhu, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero. Where do we stand with implicit neural representations? a technical and performance survey. *arXiv preprint arXiv:2411.03688*, 2024.

Kyle Gao, Yina Gao, Hongjie He, Dening Lu, Linlin Xu, and Jonathan Li. Neural radiance fields in 3d vision: A comprehensive review. *Computational Visual Media*, 2026.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems*, 33:16937–16947, 2020.

Cameron Gordon, Lachlan E. MacDonald, Hemanth Saratchandran, and Simon Lucey. D'oh: Decoder-only random hypernetworks for implicit neural representations. In *Asian Conference on Computer Vision*, pp. 128–147. Springer, 2024.

Bo He, Xitong Yang, Hanyu Wang, Zuxuan Wu, Hao Chen, Shuaiyi Huang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Towards scalable neural representation for diverse videos. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6132–6142. IEEE, 2023.

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Dhananjaya Jayasundara and Vishal M Patel. Implicit neural representations: A signal processing perspective. *arXiv preprint arXiv:2604.15047*, 2026.

Dhananjaya Jayasundara, Heng Zhao, Demetrio Labate, and Vishal M Patel. Mire: Matched implicit neural representations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8279–8288. IEEE, 2025.

Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. Fedexp: Speeding up federated averaging via extrapolation. *arXiv preprint arXiv:2301.09604*, 2023.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Chiheon Kim, Doyup Lee, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Generalizable implicit neural representations via instance pattern composers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11808–11817. IEEE, 2023.

Geeho Kim, Jinkyu Kim, and Bohyung Han. Communication-efficient federated learning with accelerated client gradient. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12385–12394. IEEE, 2024.

Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, et al. Lsdir: A large scale dataset for image restoration. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1775–1787. IEEE, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Zhen Liu, Hao Zhu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. Finer: Flexible spectral-bias tuning in implicit neural representation by variableperiodic activation functions. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2713–2722. IEEE, 2024.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. Pmlr, 2017.

Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 14194–14203. IEEE, 2021.

Bjoern H Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, Nicole Porz, Johannes Slotboom, Roland Wiest, et al. The multimodal brain tumor image segmentation benchmark (brats). *IEEE transactions on medical imaging*, 34 (10):1993–2024, 2014.

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. *Communications of the ACM*, 65(1):99–106, 2021.

Christopher Montgomery and H Lars. Xiph. org video test media (derf's collection). *Online, https://media. xiph. org/video/derf*, 6:1, 1994.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5206–5210. IEEE, 2015.

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 165–174. IEEE, 2019.

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European conference on computer vision*, pp. 523–540. Springer, 2020.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International conference on machine learning*, pp. 5301–5310. PMLR, 2019.

Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *2023 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 18507–18516. IEEE, 2023.

Vincent Sitzmann, Eric Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems*, 33:10136–10147, 2020a.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in neural information processing systems*, 33:7462–7473, 2020b.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 2974–2983. IEEE, 2018.

Teppei Suzuki. Federated learning for large-scale scene modeling with neural radiance fields. *arXiv preprint arXiv:2309.06030*, 2023.

Filip Szatkowski, Karol J Piczak, Przemysław Spurek, Jacek Tabor, and Tomasz Trzciński. Hypernetworks build implicit neural representations of sounds. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 661–676. Springer, 2023.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P Srinivasan, Jonathan T Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2846–2855, 2021.

Anirudh Thatipelli and Ali Siahkoohi. Hypernetwork-based approach for grid-independent functional data clustering. *arXiv preprint arXiv:2602.22823*, 2026.

Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, Kyoung Mu Lee, et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW)*, pp. 1110–1121. IEEE, 2017.

George Toderici, Wenzhe Shi, Radu Timofte, Lucas Theis, Johannes Balle, Eirikur Agustsson, Nick Johnston, and Fabian Mentzer. Workshop and challenge on learned image compression (clic2020). In *CVPR*, 2020.

Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

Kushal Vyas, Ahmed Imtiaz Humayun, Aniket Dashpute, Richard G Baraniuk, Ashok Veeraraghavan, and Guha Balakrishnan. Learning transferable features for implicit neural representations. 2025.

Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

Qi Wu, David Bauer, Yuyang Chen, and Kwan-Liu Ma. Hyperinr: A fast and predictive hypernetwork for implicit neural representations via knowledge distillation. *arXiv preprint arXiv:2304.04188*, 2023.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912–1920, 2015.

Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer graphics forum*, volume 41, pp. 641–676. Wiley Online Library, 2022.

Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE transactions on image processing*, 19(9):2241–2253, 2010.

Junhyeog Yun, Minui Hong, and Gunhee Kim. Fedmenf: Privacy-preserving federated meta-learning for neural fields. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2161–2171. IEEE, 2025.

Shuyi Zhang, Ke Liu, Jingjun Gu, Xiaoxu Cai, Zhihua Wang, Jiajun Bu, and Haishuai Wang. Attention beats linear for fast implicit neural representation generation. In *European Conference on Computer Vision*, pp. 1–18. Springer, 2024.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in neural information processing systems*, 32, 2019.

A APPENDIX

A.1 IMAGE EXPERIMENTS

Training data FEDFIELDS is trained on a corpus of 238,545 natural images drawn from six public datasets (See Table 3). Each image is decoded once, resized so that its shorter side is at most 512 pixels, and cached as an 8-bit array. Images with a shorter side below 256 pixels are discarded, as they cannot provide a full 256×256 training crop. The resulting corpus is shuffled using a fixed seed and partitioned IID across $K{=}10$ clients, with each client receiving 23,854–23,855 images.

Table 3: Federated training corpus for images

Dataset	Content	Images
COCO 2017 (train) (Lin et al., 2014)	everyday scenes and objects	117,465
LSDIR (Li et al., 2023)	high-resolution photographs	84,991
CelebA-HQ (Karras et al., 2017)	faces	27,000
DTD (Cimpoi et al., 2014)	textures	5,639
Flickr2K (Timofte et al., 2017)	high-resolution photographs	2,650
DIV2K (train) (Agustsson & Timofte, 2017)	high-resolution photographs	800
Total		**238,545**

Architecture The image model has four components (See Table 4). *Convolutional branch*: three $3{\times}3$ convolutions with ReLU (64, 128, and 256 channels; strides 2, 1, 1) and a $1{\times}1$ projection to 128 channels, giving a $128{\times}128$ map for a $256{\times}256$ input. *DINOv2 branch*: the image is resized to $224{\times}224$, normalized with ImageNet statistics, and passed through DINOv2 ViT-B/14 (Oquab et al., 2023); the $16{\times}16$ grid of 768-dimensional patch tokens (class and register tokens discarded) is bilinearly upsampled to $128{\times}128$ and projected to 64 channels by a $1{\times}1$ convolution (DINOv2 projection). The 85.7M backbone weights are frozen; LoRA adapters (Hu et al., 2021) with rank $r{=}8$ and $\alpha{=}16$ are inserted into the query–key–value projection of the last 8 of its 12 blocks, with B initialized to zero. *Latent representation*: the two branches are concatenated into a latent representation of $128{\times}128$ cells with 192 channels; a query coordinate $c \in [-1, 1]^2$ reads its feature by bilinear interpolation. *Hypernetwork and Decoder*: a two-layer MLP ($192 \to 256 \to 1536$, ReLU, zero-initialized output layer) predicts one shift per neuron of a SIREN decoder (Sitzmann et al., 2020b) with 6 layers of width 256 and $\omega_0{=}30$, $h_l = \sin(\omega_0(W_l h_{l-1} + b_l) + m_l(c))$, followed by a linear map to RGB.

Table 4: Image model components. Only the trainable parameters (1.42M) are exchanged during federated training; the frozen backbone is never transmitted.

Component	Configuration	Parameters	Transmitted
DINOv2 ViT-B/14 backbone	frozen, 224^2 input, $16{\times}16$ tokens	85.72 M	no
LoRA adapters	rank 8, $\alpha{=}16$, qkv of last 8 blocks	0.197 M	yes
Convolutional encoder	$64/128/256$ channels, strides $2, 1, 1$, proj. 128	0.404 M	yes
DINOv2 projection	$1{\times}1$ conv, $768{\to}64$	0.049 M	yes
Hypernetwork	$192{\to}256{\to}1536$	0.444 M	yes
Modulated SIREN	$6{\times}256$, $\omega_0{=}30$	0.330 M	yes

Federated training We train FEDFIELDS using FedAvg (McMahan et al., 2017) for 300 communication rounds, with all 10 clients participating in every round. At each round, each client performs 100 local Adam steps with a learning rate of $3{\times}10^{-4}$ using mean squared error and bfloat16 mixed precision. The server then aggregates the resulting client parameters by weighted averaging according to local dataset size.

Evaluation data We evaluate on Kodak (Eastman Kodak Company, 1993), the CLIC2020 validation set (Toderici et al., 2020), and the DIV2K validation set (Agustsson & Timofte, 2017), using 24, 20, and 24 images, respectively. We use the full Kodak dataset at its native $768{\times}512$ resolution. For CLIC2020, we select 20 images from the 41-image validation set using a fixed random seed,

while for DIV2K, we select every fourth image from the 100-image validation set. CLIC2020 and DIV2K images are center-cropped to 768×512 to ensure a consistent evaluation resolution across all three datasets.

Evaluation protocol All images are reconstructed at their native resolution. *Zero-shot*: the encoder produces the latent representation in a single forward pass, which is then decoded without any optimization. *Adapted*: starting from the zero-shot latent representation, only the latent representation is optimized for 500 Adam steps (learning rate 10^{-2}, multiplied by 0.3 after 50% and 80% of the steps), with all network weights frozen. PSNR is computed as $-10\log_{10}(\mathrm{MSE})$ on $[0,1]$ RGB over the whole image and averaged over images.

Baselines *Per-image fitting*: SIREN (Sitzmann et al., 2020b), and FINER (Liu et al., 2024) are fitted to each image for 5000 steps. STRAINER (Vyas et al., 2025) learns 5 shared layers of a 6-layer, 256-wide SIREN from 10 CelebA-HQ images resized to the Kodak resolution, following the authors' protocol; the encoder is trained once and used to initialize the fit of each test image, which is then optimized for 2,100 steps (learning rate 10^{-4}). *Amortized*: Trans-INR (Chen & Wang, 2022) and ANR (Zhang et al., 2024) are trained by us on CelebA with the training configurations (30 epochs at 178×178 and 200,000 steps at 128×128, respectively). Because their encoders accept only these fixed input sizes, test images are processed as overlapping tiles with linear-ramp blending. *Federated*: FedMeNF (Yun et al., 2025) is trained with the authors' code and published settings (a 6×256 SIREN meta-learned with 8 inner steps) on our training corpus, distributed across 10 clients; each test image is adapted from the meta-learned initialization for 500 steps.

Table 5: Image reconstruction (PSNR in dB ↑)

Type	Method	Kodak (24)	CLIC2020 (20)	DIV2K (24)
Per-image	SIREN (5,000 steps)	36.12	37.28	35.46
	FINER (5,000 steps)	35.31	36.16	34.61
	STRAINER (2,100 steps)	35.41	36.41	35.19
Amortized	Trans-INR	28.48	30.62	28.38
	ANR	39.25	40.15	38.68
Federated	FedMeNF (500 steps)	25.50	28.03	25.48
Ours	FEDFIELDS zero-shot	**39.26**	**40.68**	**39.15**
	FEDFIELDS adapted	**52.98**	**53.31**	**53.00**

Results Table 5 reports all results, and Fig. 10 shows the qualitative results. From a single forward pass, FEDFIELDS outperforms every per-image baseline by at least 3.1 dB, although these fit a separate network to each image for thousands of steps. Among amortized methods, it matches ANR on Kodak (39.26 vs. 39.25 dB) and exceeds it on CLIC2020 and DIV2K, while being trained under federated constraints. Optimizing only the latent representations for 500 steps raises the quality of FEDFIELDS to about 53 dB on all three benchmarks.

A.2 VIDEO EXPERIMENTS

Data All videos are processed consistently: frames are center-cropped to a square, resized to 128×128 using bicubic interpolation, and grouped into 16-frame clips. For training, we use all 90 sequences from DAVIS 2017 (Pont-Tuset et al., 2017) and the first 20 videos from each of the 101 UCF-101 classes (Soomro et al., 2012), extracting clips with a temporal stride of 8 frames. This produces 14,707 clips, of which 8 are held out for validation and the remaining 14,699 are shuffled with a fixed seed and partitioned IID across $K=10$ clients. We evaluate on videos not used for training, taking one 16-frame clip from the center of each video. Kinetics comprises 30 validation videos from Kinetics-400 (Kay et al., 2017), sampled from distinct action classes with a fixed seed. MPI Sintel comprises all 23 training sequences from the clean pass (Butler et al., 2012), and Xiph CIF comprises all 37 sequences from the Xiph.org CIF test collection (Montgomery & Lars, 1994).

Architecture Video uses an HNeRV-style convolutional frame codec (Chen et al., 2023) instead of a coordinate decoder. The encoder is a ConvNeXt network with three stages of 64 channels and

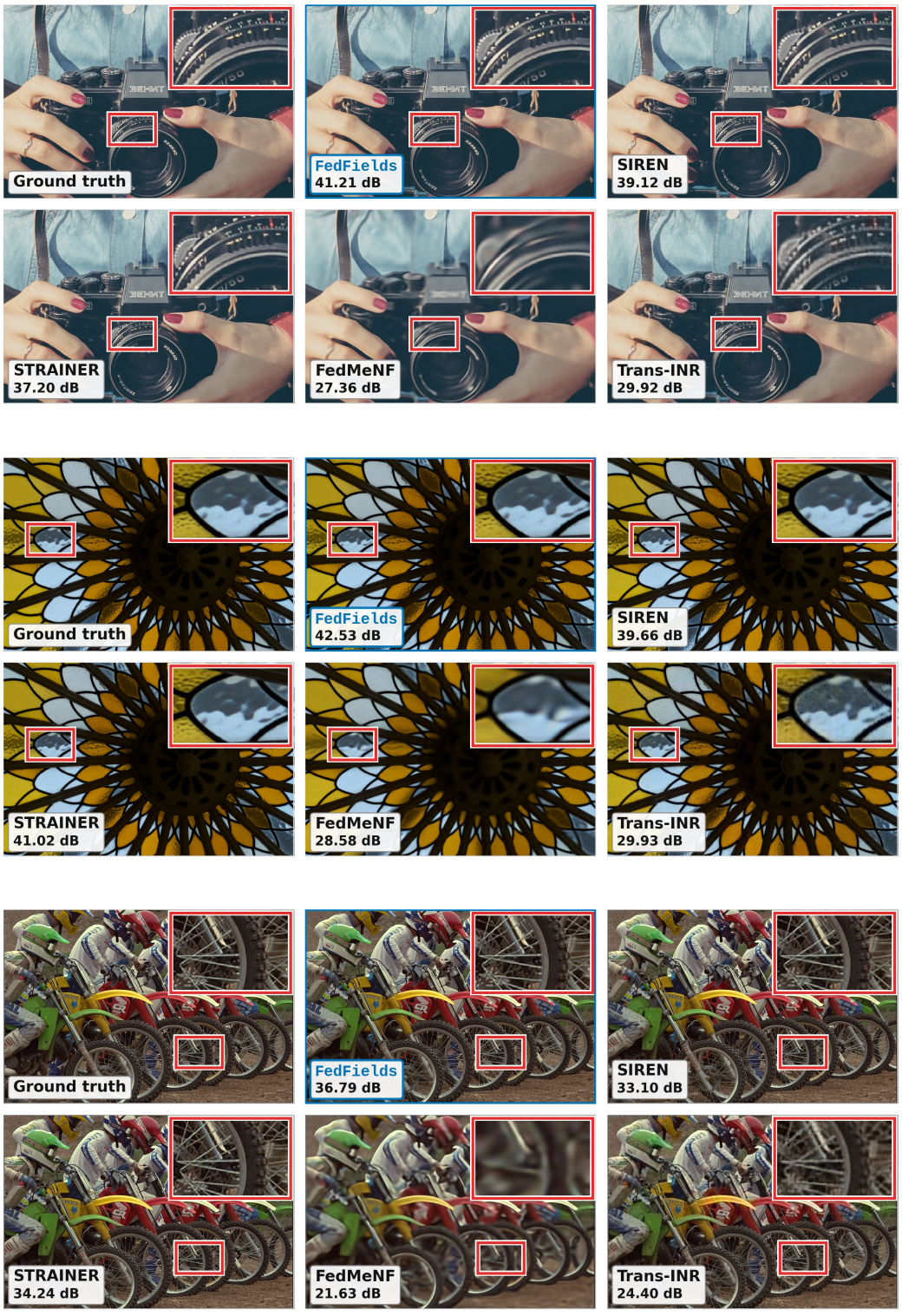


Figure 10: **Qualitative reconstruction results on unseen images.** Representative examples from CLIC2020 (top), DIV2K (middle), and Kodak (bottom). **FEDFIELDS** reconstructs the images directly using the federatively learned amortized neural field, without per-image optimization at test time.

Table 6: Video reconstruction quality on unseen clips, measured by PSNR (dB, ↑). Best results are shown in bold.

Method	Kinetics (30)	MPI Sintel (23)	Xiph CIF (37)
NeRV-Enc	26.45	24.35	25.42
D-NeRV	22.28	20.01	24.25
HNeRV (5k steps)	**38.30**	**38.77**	33.12
FEDFIELDS	38.26	35.94	**34.03**

strides 4, 2, and 1, which maps each 128×128 frame to a spatial latent representation of $64\times16\times16$ values. The decoder expands this representation with a 1×1 convolution to 204 channels, followed by two upsampling blocks (convolution, pixel shuffle by factors 4 and 2, GELU) with 170 and 142 channels, a 5×5 convolution with 118 channels at full resolution, and a 3×3 convolution to RGB. The model has 2.00M parameters (encoder 0.13M, decoder 1.86M), all of which are federated; the per-frame latent representations never leave the client.

Federated training We train for 300 communication rounds, with all 10 clients participating in every round. At each round, each client performs 200 local Adam steps with a learning rate of 10^{-3}, retaining its optimizer state across rounds. Each local step samples 8 frames from a randomly selected clip and minimizes their reconstruction mean squared error using bfloat16 mixed precision. The server then aggregates the resulting client models using FedAvg, weighted by the number of clips held by each client.

Baselines We compare against HNeRV (Chen et al., 2023), D-NeRV (He et al., 2023), and NeRV-Enc (Chen et al., 2024). HNeRV is fitted independently to each test clip from random initialization for 5000 Adam steps using the authors' training schedule. D-NeRV and NeRV-Enc are trained centrally on the same training corpus as FEDFIELDS using the authors' implementations and settings, and are evaluated on unseen test clips. D-NeRV processes each clip as two 8-frame sub-clips and additionally receives reconstructed keyframes as input, whereas NeRV-Enc directly infers the representation of an unseen clip.

Results Table 6 reports video reconstruction quality on unseen clips, and Fig. 11 shows the qualitative results. FEDFIELDS substantially outperforms NeRV-Enc and D-NeRV across all three datasets. Compared with HNeRV, which optimizes a separate representation for each test video for 5,000 steps, FEDFIELDS achieves nearly identical performance on Kinetics (38.26 vs. 38.30 dB), while outperforming it on Xiph CIF (34.03 vs. 33.12 dB). HNeRV performs better on MPI Sintel (38.77 vs. 35.94 dB).

A.3 Occupancy Experiments

Data and preprocessing Each mesh is centered at its bounding-box center, scaled so that its longest side measures 0.95, and voxelized at a resolution of 64^3 within $[-1,1]^3$. Closed meshes are filled; meshes that cannot be filled are kept as surface voxels. We train on the official training split of ModelNet40 (Wu et al., 2015) (9,843 shapes from 40 categories), distributed across $K=10$ clients in an IID split. We evaluate on 130 shapes from the ModelNet40 test split, 150 shapes from Google Scanned Objects (Downs et al., 2022), and 146 shapes from Pix3D (Sun et al., 2018).

Architecture The encoder maps each 64^3 occupancy grid to a 32^3 latent volume with 128 channels using three 3-D convolutional layers with 32, 64, and 128 channels, followed by a $1\times1\times1$ projection. For a query coordinate $c \in [-1,1]^3$, the corresponding latent feature is obtained by trilinear interpolation. The hypernetwork ($128 \to 256 \to 1536$) produces layer-wise shift modulations for a six-layer SIREN decoder of width 256 with $\omega_0=30$, following the same modulation scheme used for images. The decoder outputs a single occupancy logit for each query coordinate. The complete model contains 1.05M parameters, all of which are federated.

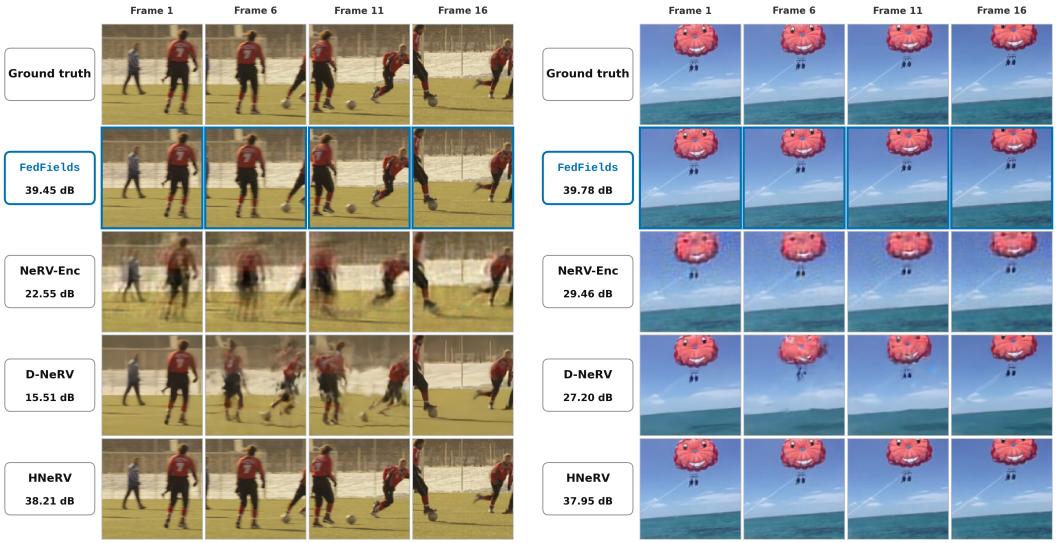


Figure 11: **Qualitative video reconstruction results.** Representative examples from Xiph (left) and Kinetics (right).

Table 7: **Occupancy reconstruction performance.** IoU (↑) on ModelNet40, GSO, and Pix3D. Test shapes duplicated in the training set are excluded.

Method	**ModelNet40** (130)	**GSO** (150)	**Pix3D** (146)
ConvONet	0.7120	0.8410	0.7250
GenSDF	0.7970	0.6860	0.8100
SIREN (5000 steps)	0.9946	0.9923	0.9934
FEDFIELDS	**0.9950**	**0.9955**	**0.9945**

Federated training All 10 clients participate in every communication round. Each client performs 200 local Adam steps with a learning rate of 3×10^{-4}, retaining its optimizer state across rounds. Each step uses 16 shapes with 4096 query points per shape, sampled to emphasize the object surface: half are drawn from the boundary region, one quarter from occupied voxels, and one quarter uniformly from the volume. Each point is jittered within its voxel and assigned the corresponding occupancy label. Training minimizes binary cross-entropy on the predicted occupancy logits, and the server aggregates client models using FedAvg weighted by local dataset size.

Evaluation We report the volumetric IoU between the predicted occupancy (logit > 0) and the ground-truth grid, evaluated at the centers of all 64^3 voxels.

Baselines We compare against SIREN (Sitzmann et al., 2020b), ConvONet (Peng et al., 2020), and GenSDF (Chou et al., 2022), evaluating all methods on the same test shapes using IoU. SIREN is fitted independently to each test shape for 5000 Adam steps using the same point sampling strategy and reconstruction objective as **FEDFIELDS**. For ConvONet, we evaluate the authors' pretrained voxel-based models and report the best-performing variant. For GenSDF, we use the authors' released stage-two checkpoint and construct its input point cloud from the surface voxels of each test shape.

Results Table 7 presents the quantitative results. After adaptation, **FEDFIELDS** achieves nearly 0.995 IoU across all three datasets, demonstrating consistently high reconstruction accuracy. Qualitative comparisons are provided in Fig. 12.

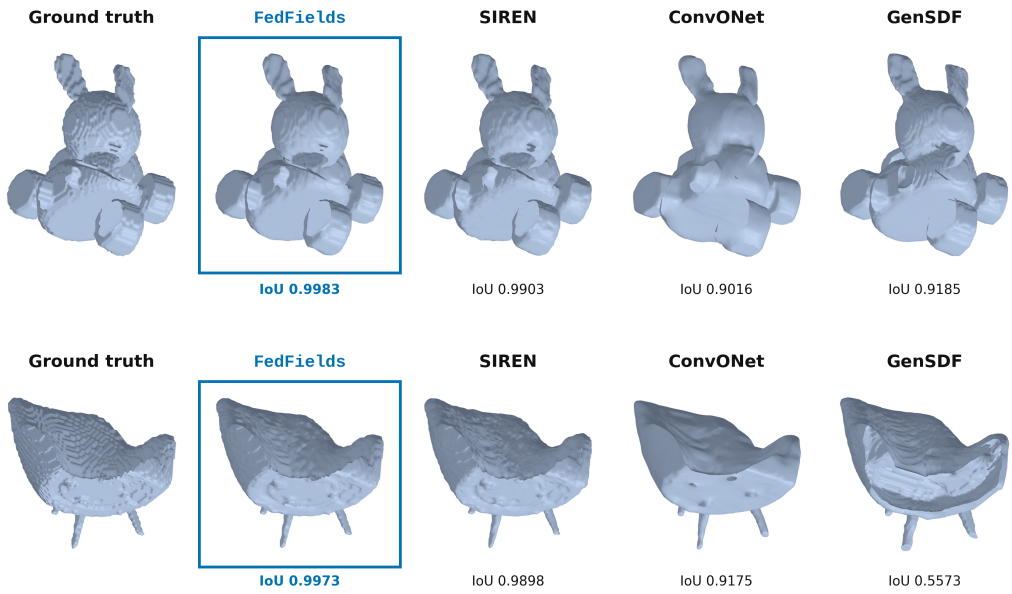


Figure 12: **Qualitative occupancy reconstruction results.** Representative examples from GSO (top) and Pix3D (bottom).

A.4 Audio Experiments

Data and preprocessing Each utterance is converted to mono, peak-normalized to $[-1, 1]$, and represented as a 16384-sample clip at 16kHz (1.02s). We train on LibriSpeech dev-clean (Panayotov et al., 2015), comprising 2693 utterances from 40 speakers, distributed IID across $K{=}10$ clients. During training, each clip is randomly cropped from its corresponding utterance. For evaluation, we use the first 16,384 samples from 10 held-out dev-clean utterances from speakers seen during training, and from 40 LibriSpeech test-clean utterances, one per speaker, representing speakers unseen during training.

Architecture The encoder consists of three 1-D convolutions (64, 128, and 256 channels; kernel size 9; stride 4; ReLU) followed by a $1{\times}1$ convolution to 128 channels, producing a temporal latent representation of 256 cells, each covering 64 samples. A query time $t \in [-1, 1]$ reads its feature by linear interpolation. The hypernetwork ($128 \to 256 \to 1536$, ReLU, zero-initialized output layer) predicts one shift per neuron of a SIREN decoder with $L{=}6$ layers of width 256 and $\omega_0{=}30$, using the same modulation as for images, $h_l = \sin(\omega_0(W_l h_{l-1} + b_l) + m_l(t))$, and a linear output giving the waveform amplitude.

Federated training We train for 120 rounds with all 10 clients participating in each round. Each client performs 200 local Adam steps with learning rate $3{\times}10^{-4}$, each on 32 clips and 4,096 randomly sampled time points shared across the clips, minimizing the mean squared error. A new optimizer is created in every round. The server aggregates client models by FedAvg, weighting each client by its number of utterances.

Evaluation For zero-shot reconstruction, the encoder produces the latent representation of a clip in one forward pass, and the decoder is queried at all 16,384 time points. For adapted reconstruction, only the latent representation is optimized for 500 Adam steps (learning rate 10^{-2}, multiplied by 0.3 after 250 and 400 steps; 4,096 sampled points per step), with all shared parameters frozen. We report PSNR computed as $10 \log_{10}(4/\mathrm{MSE})$ for signals in $[-1, 1]$, scale-invariant SNR (SI-SNR), and log-spectral distance (LSD) using a 1024-point STFT with hop size 256 and a Hann window. Because speech contains long low-amplitude segments, a silent (all-zero) prediction already achieves 24.4 and 23.8 dB PSNR on the seen and unseen sets; SI-SNR, which is insensitive to this effect, is therefore the more informative metric.

Table 8: Audio reconstruction on LibriSpeech (mean over clips).

Speakers	Setting	PSNR (dB) ↑	SI-SNR (dB) ↑	LSD ↓
Seen (10 clips)	Silent prediction	24.39	–	–
	Zero-shot	30.98	5.26	1.829
	Adapted	39.05	14.44	1.617
Unseen (40 clips)	Silent prediction	23.80	–	–
	Zero-shot	28.52	2.51	1.819
	Adapted	34.94	10.52	1.609

Results Table 8 summarizes the results. The federated model generalizes to speakers never seen during training, reaching 10.5 dB SI-SNR after adaptation, with a moderate gap to seen speakers.

A.5 How does FedFields work on RGB Hyperspectral image reconstruction?

FedFields is trained only on ordinary color photographs. To test how far this training carries, we apply the same image model, without any retraining, to hyperspectral images, which record not three colors but a full spectrum of light at every pixel.

Data We use the CAVE multispectral database (Yasuma et al., 2010), which captures indoor scenes (objects, faces, flowers, toys) at 31 narrow wavelength bands from 400 to 700 nm in 10 nm steps, at 512×512 resolution. We evaluate two tasks:

- **RGB**: the color rendering of 8 scenes that the database provides alongside each spectral scan.
- **Hyperspectral**: the full 31-band scans of 4 scenes (balloons, beads, face, and flowers).

The hyperspectral task is far from the training data. Each band is a single-wavelength grey-level image, and the 31 bands of a scene are highly correlated with each other, unlike the three color channels of a photograph. Since all compared models take three-channel images, we group the bands into triplets (400–420 nm, 430–450 nm, and so on; the last triplet covers the final three bands) and reconstruct each triplet as an image, so that no model needs to be changed. The full spectrum of a scene is then reassembled from the reconstructed triplets.

Measures Besides PSNR on the RGB images and on the band triplets, we report two measures on the reassembled 31-band scans. The *full-spectrum PSNR* measures the error over all 31 bands, with each band counted once. The *spectral angle* measures, for every pixel, how much the shape of the reconstructed spectrum differs from the true one, in degrees (lower is better); it ignores overall brightness and captures whether the relative amounts of light at different wavelengths are preserved, which is what matters for most uses of hyperspectral data. All methods follow the same protocols as in Section A.1.

Results Table 9 shows that **FedFields** transfers well to this unseen kind of data. In a single forward pass, **FedFields** outperforms the other single-pass methods on every measure, by 1.9 dB (RGB) and 1.7 dB (hyperspectral bands) over the strongest of them, ANR. It also outperforms SIREN and FINER, which fit a separate network to each image for 5,000 steps, and is comparable to STRAINER, which fits each image for 2,100 steps starting from a learned initialization: **FedFields** zero-shot is 0.7 dB better on RGB and within 0.6 dB on the hyperspectral bands. After a short adaptation of the latent representation (500 steps, with the network unchanged), **FedFields** reaches 50.5 dB on RGB and 59.2 dB on the full spectrum, 11 dB above the best competing method, and reduces the spectral angle from 5.5° (best competing method) to 0.5°, meaning that the spectrum of each pixel is reproduced almost exactly. This holds for every one of the four scenes.

Fig. 13 shows one reconstructed band (550 nm). **FedFields** recovers the fine structure of the flower petals and the grey calibration patches, while FedMeNF produces a blurred image and FINER adds visible grain. Fig. 14 shows the reconstructed spectrum at three pixels: the spectra of **FedFields** follow the true curves across the full wavelength range, whereas Trans-INR and SIREN visibly deviate at some wavelengths.

Table 9: Reconstruction of out-of-distribution CAVE images. RGB: 8 color renderings. Hyperspectral: 4 scenes with 31 bands each. PSNR in dB; spectral angle in degrees (lower is better). **FEDFIELDS** was trained on color photographs only and is not retrained for this data.

	RGB	Hyperspectral		
Method	PSNR	Band PSNR	Full-spectrum PSNR	Spectral angle
Single forward pass				
Trans-INR	32.79	39.66	39.42	7.88
ANR	40.44	45.92	44.21	7.75
FEDFIELDS (zero-shot)	**42.30**	**47.64**	**47.18**	**6.24**
Per-image optimization				
FedMeNF (500 steps)	31.42	34.27	34.13	8.44
FINER (5,000 steps)	35.93	36.88	34.76	16.72
SIREN (5,000 steps)	41.25	45.85	44.66	7.26
STRAINER (2,100 steps)	41.60	48.23	47.51	5.51
FEDFIELDS (adapted, 500 steps)	**50.52**	**58.97**	**59.21**	**0.47**

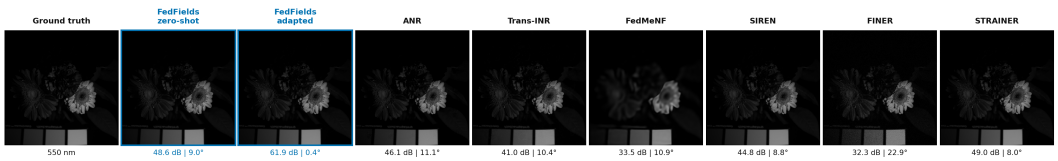


Figure 13: One band (550 nm) of the *flowers* scene. Numbers give the full-spectrum PSNR and the spectral angle for the whole scene.

Why does it work? A likely reason is that **FEDFIELDS** describes each small region of an image by local features, such as edges, textures, and smooth gradients, and rebuilds the region from this description, rather than relying on what a whole color photograph typically looks like. Such local patterns are shared by photographs and by single-wavelength images alike, so the model carries over to hyperspectral data without retraining. Where the new data does differ from the training data, a short adaptation of the latent representation, which never changes the shared network, is enough to close the remaining gap.

A.6 How does **FEDFIELDS** work on MRI reconstruction?

We next apply **FEDFIELDS**, trained only on ordinary photographs, to brain MRI scans, without any retraining.

Data We use the BraTS 2023 glioma dataset (Menze et al., 2014; Baid et al., 2021; Bakas et al., 2017), which contains brain MRI scans of patients with brain tumours, acquired at many hospitals. We take the first 10 patients and, for each, the FLAIR scan slice that shows the largest tumour area according to the provided tumour annotations, at 240×240 resolution. The slices are therefore selected from the annotations alone, independently of any method. Each grey-level slice is copied into the three colour channels expected by all compared models.

Protocol We report **FEDFIELDS** zero-shot and after 500 adaptation steps, in which only the latent representation of the image is refined while the network stays fixed. We compare it with the single-pass methods ANR and Trans-INR, and with methods that fit a separate network to each image (SIREN, STRAINER, and FedMeNF), both after the same 500 steps and at their full budgets. All methods are timed on an NVIDIA RTX A5000 GPU.

Results Table 10 shows that **FEDFIELDS** transfers well to MRI. In a single forward pass, which takes 0.02 seconds per image, it gives the highest average PSNR among the single-pass methods. With 500 adaptation steps, **FEDFIELDS** reaches 63.35 dB in 5.6 seconds per image, 17 to 31 dB higher than the per-image methods after the same number of steps and in a similar time. It also

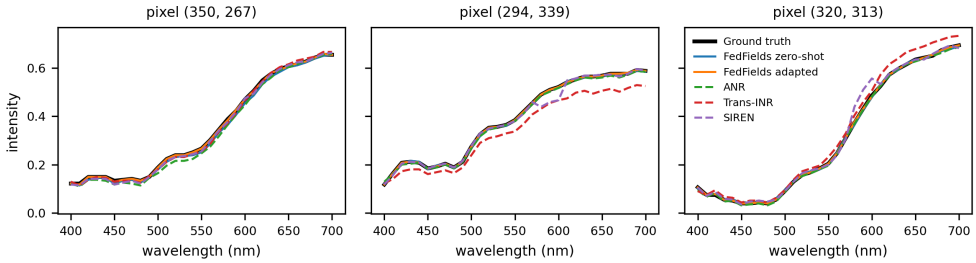


Figure 14: Reconstructed 31-band spectra at three pixels of the *flowers* scene (black: ground truth).

Table 10: Reconstruction of 10 brain MRI slices from BraTS 2023, averaged over the slices. Time is per image on one GPU.

	Method	PSNR (dB)	Time
Single pass	Trans-INR	33.30	0.04 s
	ANR	43.13	0.10 s
	FEDFIELDS (zero-shot)	**43.75**	0.02 s
500 steps	FedMeNF	32.62	8.1 s
	SIREN	34.81	5.3 s
	STRAINER	46.24	7.0 s
	FEDFIELDS	**63.35**	5.6 s
Full budget	SIREN (5,000 steps)	52.34	53 s
	STRAINER (2,100 steps)	54.62	29 s

outperforms SIREN after 5,000 steps and STRAINER after 2,100 steps on all 10 slices, while being 5 to 10 times faster. **FEDFIELDS** therefore offers a much better trade-off between quality and time.

Fig. 15 shows one slice and a close-up of the tumour. Already in a single pass, **FEDFIELDS** reproduces the brain folds and the tumour boundary, and after 500 steps, taking 5.5 seconds, it also recovers the fine texture inside the tumour, comparable to per-image methods that take 30 to 50 seconds.

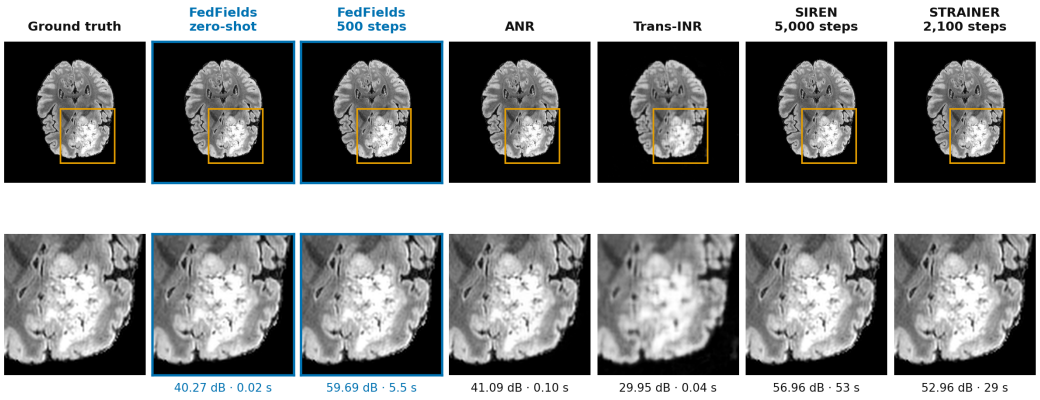


Figure 15: Reconstruction of a brain MRI slice with a tumour (BraTS 2023, the first patient of our selection). Top: full slice, with the tumour region marked in orange. Bottom: close-up of that region. Below each close-up, the PSNR on this slice and the time needed to produce it. **FEDFIELDS** was trained only on photographs and is applied without retraining, either in a single pass (zero-shot) or with 500 adaptation steps of its latent representation.

A.7 PRIVACY

This section expands the privacy evaluation of the main paper. We ask what an attacker can learn about a client's private images from what the client transmits, using three attacks of increasing realism, and report all individual results together with the reconstructions they produce.

A.7.1 SETTING

What is transmitted For images, a **FEDFIELDS** client transmits only its trainable parameters (LoRA adapters, convolutional encoder, hypernetwork, and decoder); the frozen DINOv2 weights never change and need not be sent. The latent representation of each image, which carries its specific content, is computed on the client and never leaves it. As a baseline, we use federated Reptile (Nichol et al., 2018) as a representative meta-learned federated neural field: each client adapts a shared 6-layer SIREN to its own images and transmits the adapted network. Reptile updates a network in two steps: it first fits a copy of the network to the image with a few optimizer steps, and moves the shared network only part of the way toward this fitted copy, by a fraction called the Reptile step size. In our simulation, the client performs 50 such iterations, each with 8 Adam steps on the private image and a Reptile step size of 0.1, the value used when training our federated Reptile baseline.

Attacker The attacker knows the model architecture and the shared weights sent to the client at the start of each round, and observes the update returned by the client. The attacker has no access to the client's private images. Whenever an attack involves additional choices or information, we favor the attacker by providing access beyond what would typically be available in practice, making unsuccessful attacks a stronger indication of limited information leakage.

Private images and scoring All attacks target six images from the DIV2K validation set, none of which is used for training. We measure the PSNR between the attacker's reconstruction and the private image. PSNR is not zero for an attacker who learns nothing, since even an unrelated image shares some pixel values with the private one. To show what a failed attack scores, we compute the PSNR of four guesses that need no access to the client's update (See Table 11): random noise, one of the other private images, a flat grey image, and a flat image in the private image's average colour. The last guess is the best one could do knowing only the average colour, so it marks the upper end of this range. Reconstructions scoring about 8–13 dB therefore reveal, at most, the rough colour of the image, whereas recognizable reconstructions typically score above 20 dB.

Table 11: Reference PSNR of guesses made without any attack.

Guess	PSNR (dB)
Random noise	8.17
A different private image	8.47
Flat grey image	11.65
Flat image in the private image's average colour	12.63

A.7.2 GRADIENT INVERSION

Gradient inversion (Zhu et al., 2019; Geiping et al., 2020) reconstructs an image from an intercepted gradient. The attacker starts from a random image, computes the gradient that this image would produce, and repeatedly adjusts the image so that its gradient points in the same direction as the intercepted one, with a mild smoothness prior. We use $3{,}000$ Adam steps and two random starts, and keep the start whose gradient matches best, a choice the attacker can make without knowing the private image. We attack a gradient computed from a single image, the easiest setting for the attacker, and tell the attacker the exact pixel positions on which it was computed.

Table 12 and Fig. 16 show that Reptile's gradients reveal every private image (21.01 dB on average). The attack fails completely against **FEDFIELDS**: its reconstructions are colourful noise, scoring below even a random-noise guess (5.43 dB versus 8.2 dB).

Table 12: Gradient inversion: PSNR (dB) of the reconstruction of each private image.

	#1	#2	#3	#4	#5	#6	Mean
Reptile	21.21	20.78	21.52	20.03	19.42	23.09	21.01
FEDFIELDS	**5.57**	**5.05**	**5.62**	**5.45**	**5.48**	**5.44**	**5.43**

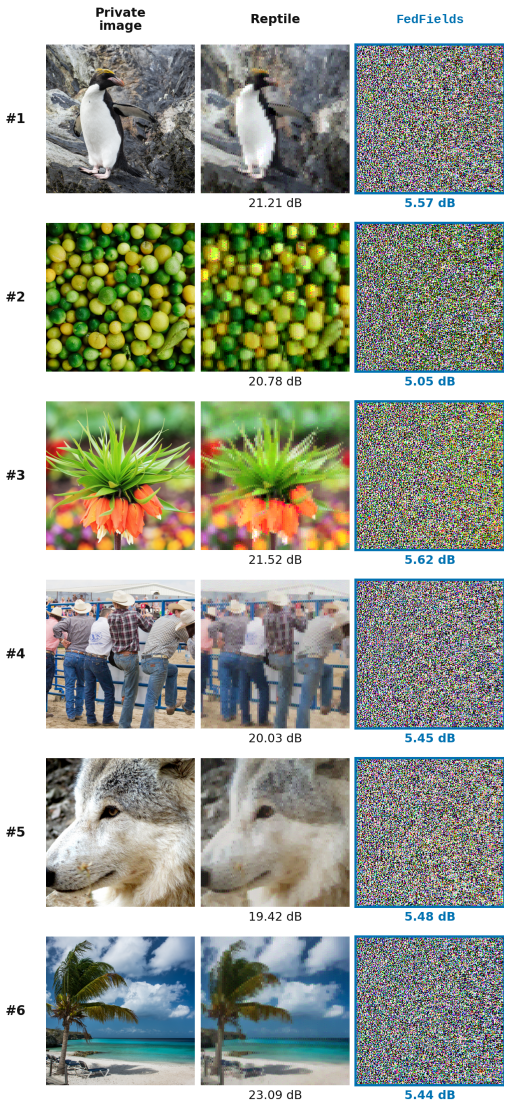


Figure 16: Gradient inversion on all six private images. Reptile's gradients reveal recognizable images; the reconstructions obtained from **FEDFIELDS**'s gradients contain no information about the images.

A.7.3 WEIGHT READOUT

Gradient inversion requires solving a difficult optimization problem, so a failed attack could in principle be caused by the optimization rather than by privacy. We therefore test a simpler attack that requires no optimization. The attacker directly uses the weights uploaded by the client after local training and attempts to recover the client's private image from them.

For Reptile, this attack is straightforward. The uploaded weights define a coordinate network that maps pixel coordinates to colours. The attacker therefore evaluates this network at every pixel coor-

dinate to directly render an image. For **FEDFIELDS**, the uploaded weights alone cannot produce the private image because reconstruction also requires the image-specific latent representation, which remains on the client. We therefore give the attacker every input available to it: 8 public training images, a flat grey image, and random noise. Each input is passed through the uploaded model, and the attacker keeps the output with the highest similarity to the private image. This gives the attacker an additional advantage, since a real attacker would not have the private image available to select the best output.

To determine whether the client's update reveals any new information, we also perform exactly the same readout using the weights sent to the client *before* local training. These weights serve as a control because they have not yet been updated using the client's private image. If the client update leaks private content, the readout from the weights *after* local training should therefore reconstruct the private image more accurately than the readout from the weights *before* local training.

We repeat this attack for each of the six private images, with each image held by a client in turn. We consider two client settings: a client containing only the private image, which is the setting most favourable to the attacker, and a client containing the private image together with 15 public images, matching our training setup.

Table 13 and Fig. 17 show that every Reptile upload directly reproduces its client's private image, reaching 25.5,dB on average. Importantly, the reconstructed image corresponds specifically to that client's private image: when the same rendering is compared with the other five private images, the PSNR drops to 8.5,dB, comparable to two unrelated images (See Table 11). In contrast, the best readout from **FEDFIELDS** simply reproduces the attacker's supplied input, such as a public photograph or the grey image. Its PSNR remains within 0.15,dB of the corresponding readout obtained *before* the client update for every private image and in both client settings. Thus, the client update does not add recoverable content from the private image.

Table 13: Weight readout: PSNR (dB) measured against each private image. For **FEDFIELDS**, the value in parentheses reports the same readout using the weights before the client update.

	#1	#2	#3	#4	#5	#6	Mean
Reptile upload	24.34	28.25	26.20	23.85	24.73	25.67	25.51
FEDFIELDS, 1-image client	12.54 (12.59)	10.25 (10.10)	12.38 (12.33)	11.78 (11.80)	11.63 (11.62)	11.93 (11.92)	11.75 (11.73)
FEDFIELDS, 16-image client	12.57 (12.59)	10.09 (10.10)	12.33 (12.33)	11.80 (11.80)	11.65 (11.62)	11.93 (11.92)	11.73 (11.73)

What does an update change? The previous attacks score a whole reconstruction with a single PSNR value, which could hide a small leak. We therefore look directly at what a client's update changes in the model. For each attack input, we take the image the model produces after the update and subtract the image it produced before the update. If the update has stored the private image, this difference should look like the private image. Fig. 18 shows these difference images; since the changes can be small, each is stretched to full contrast, with mid-grey meaning no change. Table 14 measures how closely each difference resembles the private image, using a correlation where 1 means identical and 0 means unrelated. We report it separately for the image's structure (its shapes and edges) and for its average colour.

For Reptile, the difference is the private image itself. Its structure matches the private image almost perfectly (correlation 0.97), and the change is as large as the image itself.

For **FEDFIELDS**, the difference contains none of the private image's structure: the correlation is close to 0, whether the attacker feeds a grey image or a public photograph. With a public photograph as input, the difference does show some edges, but they are the edges of that photograph, not of the private image. The update only slightly changes how the model renders the input it is given.

One weak signal does remain, and it concerns colour only. When a client holds a single image, its update shifts the model's output colours slightly toward that image's average colour. For example, the private image #2 is a green-and-yellow scene, and the difference for that client has a green tint. This shift is small, at most 0.17 on the $[-1, 1]$ intensity scale compared with about 1.2 for Reptile,

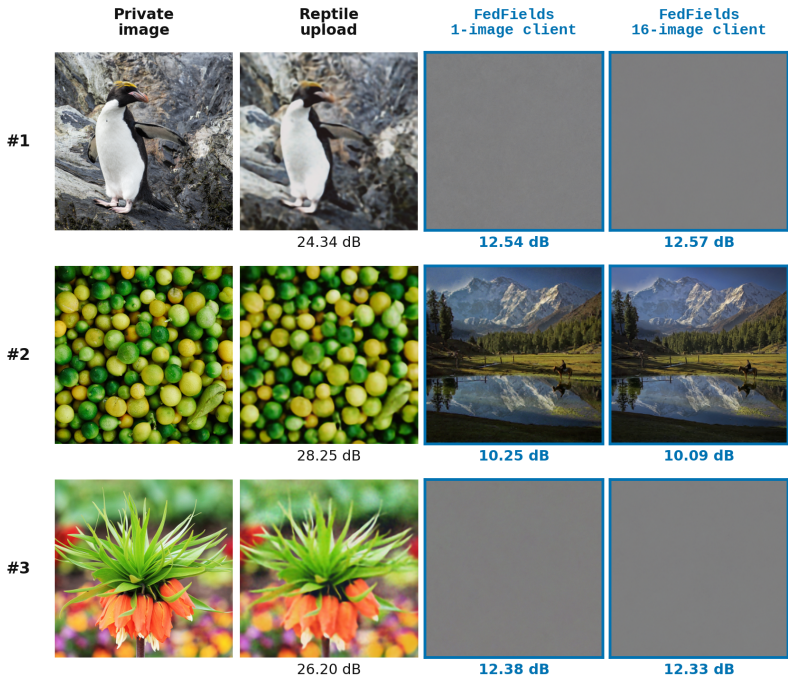


Figure 17: Weight readout: A Reptile upload, evaluated at every pixel, reproduces the private image. The best readout of a **FedFields** upload only reproduces the attacker's own input (here a flat grey image or a public photograph), regardless of the number of images the client holds.

and it reveals no shapes or objects. When the client holds 16 images, as in our training, the shift becomes even smaller and no longer follows the colour of the private image.

Table 14: Correlation between the change caused by an update and the change that would reproduce the private image, averaged over the six images (1: identical, 0: unrelated). Structure compares the images pixel by pixel after removing their average colour; average colour compares only the red, green, and blue averages, and is therefore noisy for a single image.

	Structure	Average colour
Reptile	0.97	0.92
FedFields, 1-image client, grey input	−0.03	0.80
FedFields, 1-image client, public-image input	0.07	0.50
FedFields, 16-image client, grey input	0.00	−0.01

A.7.4 Membership inference

The previous attacks try to recover image content. Membership inference asks a simpler question: was a given image among a client's training data? Even when an update reveals no content, it may still reveal participation. The attack works as follows. A client trains on N private images, called *members*. The attacker holds a candidate image and asks whether it was one of them. To decide, the attacker evaluates the candidate under two models: the broadcast model from before the client's update, and the model after it. Each evaluation reconstructs the candidate and yields a reconstruction error, and the membership score is the difference between the two, how much the client's update improved the model on that particular image. Training improves the model most on the images it actually trained on, so a large improvement suggests the candidate is a member. We simulate clients holding between 1 and 1,024 face images from CelebA, so that members and *non-members* (held-out images) come from the same distribution and cannot be told apart by appearance alone. Each client performs the usual 100 local steps; we use five clients per setting, scoring up to 32 members and 32 non-members per client. Success is measured with AUC: the probability that

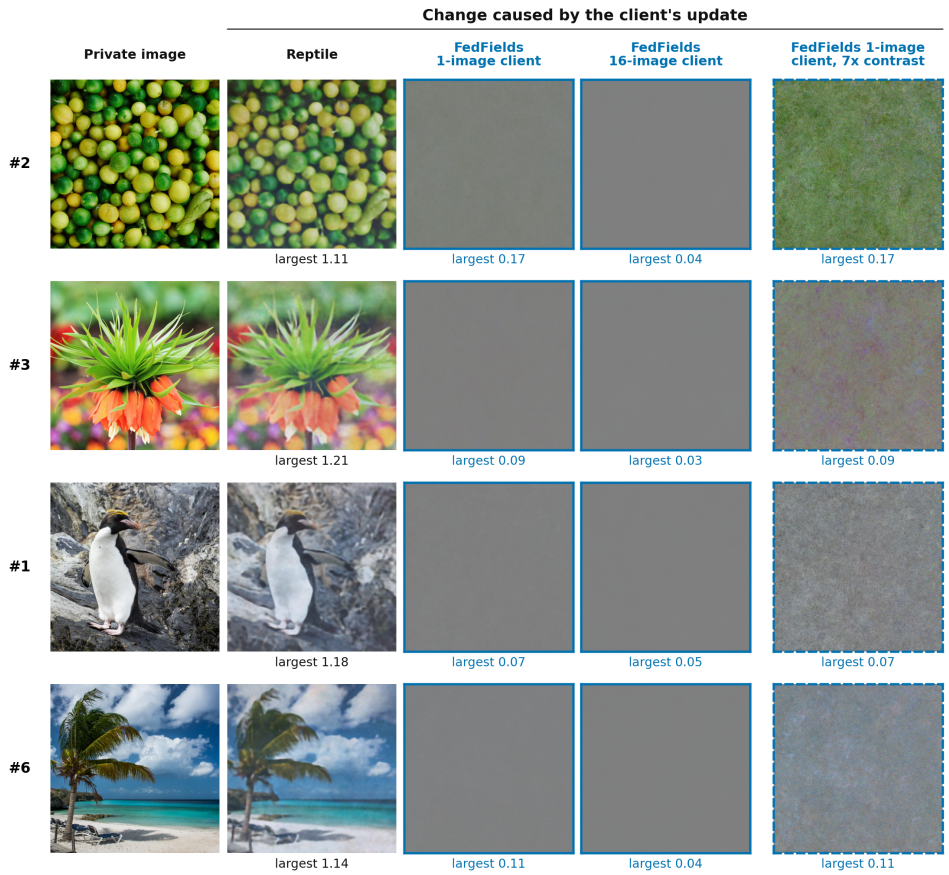


Figure 18: What a client's update reveals. A client trains on its private image (left) and uploads its updated model. To see what the update stored, we compare the image the model produces after the update with the image it produced before, and show their difference (mid-grey: no change; all difference images on the same scale). Reptile's model is evaluated directly at every pixel; **FEDFIELDS**'s model cannot produce an image without the private latent representation, so we feed it a flat grey image instead. For Reptile, the difference is the private image itself: the update has stored it. For **FEDFIELDS**, the difference is nearly flat, both for a client holding only the private image and for a client holding it among 16 images. Magnified $7\times$ (last column), the 1-image client's difference shows only a faint colour tint and no shapes or objects of the private image. Numbers give the largest change on the $[-1, 1]$ intensity scale.

a randomly chosen member receives a higher score than a randomly chosen non-member, so 0.5 means the attacker is guessing and 1.0 means perfect identification. Alongside AUC, we report the member advantage: the average improvement of member images minus that of non-members, in dB. AUC measures whether members can be ranked above non-members; the advantage measures how large the separation is, and a value near zero indicates that the membership signal itself has vanished rather than merely going undetected by this attacker.

Table 15 shows that membership can be inferred when a client holds few images: the attack is perfect for up to 4 images and still reaches 0.94 AUC for 16 images. As the client's dataset grows, the contribution of each image to the update shrinks, and the attack approaches chance level at 256 images, where the member advantage also vanishes (0.1 ± 0.4 dB). This behaviour is inherent to federated updates rather than specific to our architecture: separating content into client-side latent representations prevents the reconstruction of images, but a client that trains on very few images still leaves a measurable trace of which images they were. Standard defenses, such as secure aggregation or differential privacy, address this case.

Table 15: Membership inference against **FEDFIELDS** as a function of the number of images a client holds (5 clients per setting, up to 32 member and 32 non-member candidates each). The member advantage is how much better the updated model reconstructs the client's own images than images it did not train on (difference in PSNR, dB; mean $\pm$ standard deviation over clients).

Images per client	1	4	16	64	256	1,024
Attack AUC	1.00	1.00	0.94	0.68	0.53	0.52
Member advantage (dB)	7.7 ± 2.3	3.1 ± 1.1	1.8 ± 0.7	0.0 ± 0.6	0.1 ± 0.4	0.3 ± 0.5

A.8 ABLATION STUDIES

We study how the design of **FEDFIELDS** and the federated training setting affect reconstruction quality. All results report zero-shot PSNR on the 24 Kodak images at native resolution, obtained with a single forward pass. We consider two latent representation resolutions. Model-design and aggregation studies use the final configuration with a 128×128 latent representation (*grid 128*), whereas experiments requiring multiple federated training runs use a lighter 64×64 configuration (*grid 64*) to reduce computational cost. Within each table, all variants are trained on the same data using the same number of communication rounds and local optimization steps, ensuring a consistent comparison.

Model design *(grid 128)* Table 16 examines the contribution of individual components of **FEDFIELDS** by modifying one component at a time. The spatial structure of the latent representation is particularly important, replacing it with a single global latent vector reduces PSNR from 39.26 to 19.51 dB, indicating that spatially localized conditioning is critical for representing unseen images. Both encoder branches also contribute substantially. Removing the convolutional branch reduces PSNR to 23.02 dB, highlighting its role in preserving local image detail, while removing the DINOv2 branch yields 34.00 dB, showing the additional benefit of pretrained semantic features.

Table 16: Model design ablations using grid 128. Zero-shot reconstruction performance on Kodak, measured by PSNR (dB, ↑).

	Variant	**PSNR (dB)**
	FEDFIELDS	**39.26**
Encoder	Without DINOv2 branch (convolutional only)	34.00
	Without convolutional branch (DINOv2 only)	23.02
Adaptation	Without LoRA (frozen DINOv2)	35.38
	LoRA rank 4 instead of 8	36.40
Latent	Single global latent instead of spatial	19.51

Aggregation rule *(grid 128)* Table 17 compares FedAvg with three aggregation rules designed for federated optimization: FedNova (Wang et al., 2020), FedExp (Jhunjhunwala et al., 2023), and FedACG (Kim et al., 2024). As can be seen, our framework therefore works well with the standard FedAvg rule and requires no specialized aggregation.

Table 17: Aggregation rules (grid 128; heterogeneous local steps). Kodak zero-shot PSNR (dB).

	FedAvg	FedNova	FedExp	FedACG
PSNR	**39.26**	36.78	34.84	25.56

Number of clients *(grid 64)* Table 18 examines the effect of distributing the same training corpus across 10 to 100 clients, corresponding to approximately 23,900 to 2,400 images per client. Reconstruction quality does not degrade as the data is distributed across more clients: PSNR improves from 10 to 25 clients and remains stable thereafter. Because each client performs the same number of local optimization steps per round, increasing the number of clients also increases the total local

computation per round. These results show that **FEDFIELDS** remains effective as the training data is distributed across a larger number of clients with smaller local datasets.

Table 18: Number of clients (grid 64, 300 rounds, all clients participating in every round). Kodak zero-shot PSNR (dB).

Clients	10	25	50	100
Images per client	∼23,900	∼9,500	∼4,800	∼2,400
PSNR	29.39	30.85	**31.13**	30.98

Non-IID data *(grid 64)* We distribute the data across the 10 clients with a Dirichlet distribution, which makes client dataset sizes increasingly unequal as the concentration parameter α decreases (See Table 19). At α=1, the largest client holds about 94 times more images than the smallest. Quality changes by at most 0.13 dB, indicating that **FEDFIELDS** is robust to imbalanced client data.

Table 19: Client data imbalance (grid 64, 150 rounds). Kodak zero-shot PSNR (dB).

Partition	IID	Dirichlet α=10	Dirichlet α=1
PSNR	29.33	29.33	29.20