SAMPLING BY TARGET-NEIGHBORHOOD SENSITIVITY FOR ACCELERATED IMPLICIT NEURAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Implicit Neural Representations (INRs) learn underlying continuous target functions from discrete observations, but full-coordinate training is typically time-intensive. To this end, we introduce Sampling by Target-Neighborhood Sensitivity (STaNS), a lightweight strategy that dynamically samples an informative coordinate subset for accelerated INR training. Our key insight is simple: *we view each observed coordinate as a target-neighborhood (TaN) representative of the underlying continuous target function*. To characterize the local informativeness of a representative, we define *TaN mismatch* as the discrepancy between the current INR prediction anchored at that coordinate and target values at locally offset coordinates. We then define *TaN sensitivity* as the worst first-order growth of this mismatch as the offset becomes infinitesimal, capturing how rapidly the anchored prediction loses consistency with the local target behavior. During INR training, STaNS dynamically updates coordinate-wise *TaN scores* derived from the closed-form *TaN sensitivity* for probabilistic sampling, prioritizing informative coordinates to accelerate training. Validated across audio, image, and shape modalities, STaNS establishes state-of-the-art INR acceleration, reducing time-to-quality by up to 23.41% over existing samplers and delivering up to $3.74\times$ speedup over full-coordinate training.

1 INTRODUCTION

Implicit Neural Representations (INRs) leverage Multilayer Perceptrons (MLPs) to parameterize diverse signals as continuous coordinate-based functions (Sitzmann et al., 2020), with discrete observations providing supervision for the underlying continuous target function (Xu et al., 2022a). This continuous coordinate-to-value representation can be applied to various signal modalities, including audio (Su et al., 2022), images (Mehta et al., 2021), and shapes (Park et al., 2019), enabling evaluation at arbitrary coordinates for applications such as signal compression (Strümpler et al., 2022), computational physics (Shen et al., 2025), and novel view synthesis (Mildenhall et al., 2021).

However, efficient INR training remains challenging. As high-resolution signals contain large numbers of observations (e.g., 250,000 for a 500×500 image), several acceleration paradigms have been studied to reduce fitting cost, each introducing trade-offs: decomposition-based methods (Liu et al., 2023; Yang, 2023) divide the domain into local representations, while increasing complexity; explicit representation methods (Chen et al., 2022; Müller et al., 2022) use trainable explicit structures, trading memory for efficiency; and meta-learning methods (Tancik et al., 2021; Chen & Wang, 2022) learn initializations or generate INR weights, but rely on representative meta-training data.

In contrast, sampling-based methods provide a lightweight alternative by sampling a small coordinate subset at each iteration for efficient training, without modifying the INR architecture or requiring external data (Zhang et al., 2025a). As coordinates contribute unequally to optimization progress (Katharopoulos & Fleuret, 2018) and their utility evolves with the INR state (Zhang et al., 2024a), subset sampling should be informative and dynamic. Existing methods select coordinates based on reconstruction error, frequency cues, or functional influence (Kheradmand et al., 2024; Zhang et al., 2025b; 2026). Although these criteria capture different aspects of coordinate utility, they overlook the discrete-to-continuous nature of INR, in which each coordinate serves as an ob-

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.

servational anchor for local target behaviors and its informativeness arises from behaviors not yet captured by the evolving INR at the current optimization stage.

To address this limitation, we propose Sampling by Target-Neighborhood Sensitivity (STaNS), a lightweight INR training acceleration strategy that views each coordinate as a target-neighborhood (TaN) representative of the underlying continuous target function. Under this view, *TaN mismatch* measures the inconsistency between the current INR prediction anchored at an observed coordinate and target values at nearby coordinates, while *TaN sensitivity* quantifies the worst first-order growth of this inconsistency under infinitesimal offsets. The resulting closed-form *TaN sensitivity* couples a target-neighborhood structure estimate computed once at initialization with the evolving INR state, enabling STaNS to dynamically update coordinate-wise *TaN scores* and probabilistically sample informative coordinate subsets throughout training. Across audio, image, and shape reconstruction tasks and evaluated by diverse metrics including PSNR, STOI, and IoU, STaNS establishes state-of-the-art INR training acceleration, reducing time-to-quality by up to 23.41% over existing sampling methods and delivering up to $3.74\times$ speedup over full-coordinate training. Ablation and sensitivity studies further validate the effectiveness of individual components. Our key contributions are:

- We introduce a target-neighborhood view of INR coordinate sampling, defining TaN mismatch and TaN sensitivity to quantify each coordinate's informativeness by local target behaviors unresolved by the current INR.
- We derive the closed-form TaN sensitivity, revealing a lightweight yet informative sampling criterion that couples the target-neighborhood structure with the evolving INR state.
- We propose STaNS, an INR sampler that converts the closed-form TaN sensitivity into coordinate-wise TaN scores and selects informative coordinates for accelerated training.
- We validate STaNS across audio, image, and shape reconstruction tasks with diverse metrics and against leading baselines, demonstrating state-of-the-art INR training acceleration.

2 Related Work

2.1 Implicit Neural Representations

INRs model natural signals as continuous coordinate-based functions parameterized by neural networks (Dupont et al., 2022), accommodating diverse signal modalities including audio (Brunetto et al., 2025), shapes (Gropp et al., 2020), images (Mehta et al., 2021), scenes (Barron et al., 2021), and videos (Chen et al., 2021). Their continuous formulation enables evaluation at arbitrary coordinates, supporting various applications such as signal compression (Pistilli et al., 2022; Schwarz et al., 2023), computational physics (Smith et al., 2022; Liu et al., 2025), inverse imaging (Huang et al., 2023; Zhang et al., 2024b), image enhancement (Yang et al., 2023; Chen et al., 2024), and novel view synthesis (Verbin et al., 2024). Despite this expressivity, fitting INRs to high-resolution signals remains costly due to large observation sets, which necessitates acceleration strategies.

2.2 Acceleration for INR Training

Several paradigms have been explored for INR training acceleration. Decomposition-based methods use regular blocks (Liu et al., 2023), learned regions (Ben-Shabat et al., 2024), Voronoi cells (Rebain et al., 2021), or Laplacian-pyramid patches (Saragadam et al., 2022), while increasing representation complexity. Explicit methods store signals in sparse voxels (Liu et al., 2020), point clouds (Xu et al., 2022b), tensors (Chen et al., 2022), or metric grids (Wang et al., 2025), trading memory for efficiency. Meta-learning methods learn transferable initializations (Yang et al., 2025) or generate weights via hypernetworks (Gu & Yeung, 2025), but depend on representative meta-training distributions. Other methods accelerate INR optimization via split architectures (Liang et al., 2022), reparameterization (Shi et al., 2024), data transformation (Seo et al., 2024), or normalization (Cai et al., 2024). In contrast, sampling-based methods reduce per-iteration cost without modifying the INR architecture by sampling a small coordinate subset for efficient training based on reconstruction error (Kheradmand et al., 2024), edge or frequency cues (Gai et al., 2023; Zhang et al., 2025b), evolutionary selection (Zhang et al., 2025a), or functional influence (Zhang et al., 2024a; 2026). Yet these criteria overlook each coordinate's intrinsic role as an observational anchor for local target behaviors and how its informativeness arises from behaviors not yet captured by the evolving INR.

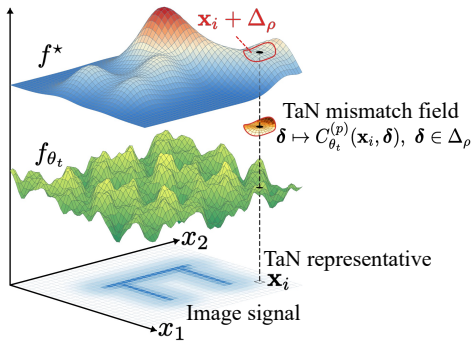


Figure 1: Illustration of the target-neighborhood view.

3 Methodology

3.1 Problem Formulation

Let $f^\star : \mathbb{R}^m \to \mathbb{R}^n$ denote the continuous function underlying a natural signal, observed through $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^D$, where $\mathbf{x}_i$ is a coordinate and $\mathbf{y}_i = f^\star(\mathbf{x}_i)$ is the observed value. An INR approximates $f^\star$ with an MLP $f_{\theta_t} : \mathbb{R}^m \to \mathbb{R}^n$ parameterized by θ_t. At iteration t, full-coordinate optimization with learning rate η_t updates

$$\theta_{t+1} = \theta_t - \eta_t \nabla_{\theta_t} \frac{1}{D} \sum_{i=1}^{D} \mathcal{L}(f_{\theta_t}(\mathbf{x}_i), \mathbf{y}_i)\,. \tag{1}$$

where $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a reconstruction loss.

To reduce fitting cost, sampling-based methods select an index subset $\mathcal{B}_t \subset \{1, \ldots, D\}$ at iteration t for training (Zhang et al., 2026). Under the discrete-to-continuous INR formulation, each observed coordinate anchors local behavior of $f^\star$, making its evolving informativeness critical for sampling. This motivates modeling each observed coordinate as a target-neighborhood representative.

3.2 Target-Neighborhood Mismatch and Sensitivity

We characterize a TaN representative by comparing its anchored INR prediction with neighboring target values.

Definition 1 (TaN mismatch). *Given an observed coordinate $\mathbf{x}_i$ and radius $\rho > 0$, let $\Delta_\rho := \{\boldsymbol{\delta} \in \mathbb{R}^m : \|\boldsymbol{\delta}\|_2 \leq \rho\}$ denote the offsets defining its target neighborhood $\mathbf{x}_i + \Delta_\rho$. For $p \geq 1$ and $\boldsymbol{\delta} \in \Delta_\rho$, the p-order TaN mismatch is*

$$C^{(p)}_{\theta_t}(\mathbf{x}_i, \boldsymbol{\delta}) = \|f_{\theta_t}(\mathbf{x}_i) - f^\star(\mathbf{x}_i + \boldsymbol{\delta})\|_2^p\,. \tag{2}$$

As illustrated in Fig. 1, each TaN representative $\mathbf{x}_i$ induces the TaN mismatch field $\boldsymbol{\delta} \mapsto C^{(p)}_{\theta_t}(\mathbf{x}_i, \boldsymbol{\delta})$ over $\boldsymbol{\delta} \in \Delta_\rho$. This is summarized by TaN sensitivity as follows.

Definition 2 (TaN sensitivity). *The p-order TaN sensitivity of $\mathbf{x}_i$ is the worst-case first-order growth of its TaN mismatch field in the infinitesimal limit:*

$$S^{(p)}_{\theta_t}(\mathbf{x}_i) = \lim_{\rho \to 0^+} \frac{\sup_{\boldsymbol{\delta} \in \Delta_\rho} \left[C^{(p)}_{\theta_t}(\mathbf{x}_i, \boldsymbol{\delta}) - C^{(p)}_{\theta_t}(\mathbf{x}_i, \mathbf{0}) \right]}{\rho}. \tag{3}$$

The following proposition establishes the closed-form expression for TaN sensitivity.

Proposition 1 (Closed-form TaN sensitivity). *Let $\mathbf{r}_i^t = f_{\theta_t}(\mathbf{x}_i) - \mathbf{y}_i = f_{\theta_t}(\mathbf{x}_i) - f^\star(\mathbf{x}_i)$ denote the current residual at $\mathbf{x}_i$. If $f^\star$ is differentiable in the neighborhood of $\mathbf{x}_i$ and $\mathbf{r}_i^t \neq \mathbf{0}$, then for any $p \geq 1$,*

$$S^{(p)}_{\theta_t}(\mathbf{x}_i) = p \|\mathbf{r}_i^t\|_2^{p-2} \left\| J_{f^\star}(\mathbf{x}_i)^\top \mathbf{r}_i^t \right\|_2. \tag{4}$$

Proof. By Definition 2, TaN sensitivity is the worst-case first-order growth of $C^{(p)}_{\theta_t}(\mathbf{x}_i, \boldsymbol{\delta})$ as $\boldsymbol{\delta} \to \mathbf{0}$. It therefore suffices to compute the first-order variation of the TaN mismatch with respect to $\boldsymbol{\delta}$. Since $f^\star$ is differentiable around $\mathbf{x}_i$,

$$f^\star(\mathbf{x}_i + \boldsymbol{\delta}) = f^\star(\mathbf{x}_i) + J_{f^\star}(\mathbf{x}_i)\boldsymbol{\delta} + \mathbf{o}(\|\boldsymbol{\delta}\|_2).$$

Substituting this into TaN mismatch gives

$$C^{(p)}_{\theta_t}(\mathbf{x}_i, \boldsymbol{\delta}) = \left\|\mathbf{r}_i^t - J_{f^\star}(\mathbf{x}_i)\boldsymbol{\delta} + \mathbf{o}(\|\boldsymbol{\delta}\|_2)\right\|_2^p.$$

Let $g_i^t(\boldsymbol{\delta}) = C^{(p)}_{\theta_t}(\mathbf{x}_i, \boldsymbol{\delta})$. Since $\mathbf{r}_i^t \neq \mathbf{0}$, g_i^t is differentiable at $\boldsymbol{\delta} = \mathbf{0}$, and the chain rule yields

$$\nabla_{\boldsymbol{\delta}} g_i^t(\mathbf{0}) = -p\|\mathbf{r}_i^t\|_2^{p-2} J_{f^\star}(\mathbf{x}_i)^\top \mathbf{r}_i^t.$$

Thus,

$$g_i^t(\boldsymbol{\delta}) - g_i^t(\mathbf{0}) = \nabla_{\boldsymbol{\delta}} g_i^t(\mathbf{0})^\top \boldsymbol{\delta} + o(\|\boldsymbol{\delta}\|_2).$$

Taking the supremum over $\boldsymbol{\delta} \in \Delta_\rho$ and using Euclidean norm duality yields

$$\sup_{\boldsymbol{\delta}\in\Delta_\rho} \left[g_i^t(\boldsymbol{\delta}) - g_i^t(\mathbf{0})\right] = \rho \left\|\nabla_{\boldsymbol{\delta}} g_i^t(\mathbf{0})\right\|_2 + o(\rho).$$

Dividing by ρ and taking $\rho \to 0^+$ gives the result. □

Notably, the signal-local Jacobian $J_{f^\star}(\mathbf{x}_i)$ is a static property of the underlying continuous target function $f^\star$ and is independent of the evolving INR parameters θ_t. Given $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^D$, it can therefore be estimated once from neighboring observations at initialization and reused throughout training (see Appendix A.3.1 for estimation details). Reusing this static estimate keeps TaN sensitivity evaluation lightweight, while the residual $\mathbf{r}_i^t$ captures the evolving INR state. Their coupling evaluates target-neighborhood variation in relation to the current reconstruction error, yielding an informative criterion for coordinate sampling (see Sec. 4.6 for component ablations).

3.3 A Spectral Interpretation of TaN Sensitivity

We further interpret TaN sensitivity via the spectral learning behavior of MLPs to characterize its informativeness for INR coordinate sampling. Let $\widehat{\mathbf{r}}_i^t = \mathbf{r}_i^t / \|\mathbf{r}_i^t\|_2$ denote the normalized residual. For $\mathbf{r}_i^t \neq \mathbf{0}$, Proposition 1 yields the equivalent form

$$S^{(p)}_{\theta_t}(\mathbf{x}_i) = p\|\mathbf{r}_i^t\|_2^{p-1} \left\|J_{f^\star}(\mathbf{x}_i)^\top \widehat{\mathbf{r}}_i^t\right\|_2. \tag{5}$$

To expose the spectral role of the residual-projected signal-local Jacobian $J_{f^\star}(\mathbf{x}_i)^\top \widehat{\mathbf{r}}_i^t$, consider a target function admitting the Fourier representation $f^\star(\mathbf{x}) = \sum_{\boldsymbol{\omega}\in\mathbb{Z}^m} \mathbf{a}_{\boldsymbol{\omega}} e^{\mathrm{i}2\pi\boldsymbol{\omega}^\top \mathbf{x}}$, where $\mathbf{a}_{\boldsymbol{\omega}}$ denotes the vector-valued Fourier coefficient at frequency $\boldsymbol{\omega}$. Substituting this representation into $J_{f^\star}(\mathbf{x}_i)^\top \widehat{\mathbf{r}}_i^t$ gives

$$J_{f^\star}(\mathbf{x}_i)^\top \widehat{\mathbf{r}}_i^t = \sum_{\boldsymbol{\omega}\in\mathbb{Z}^m} \mathrm{i}2\pi\boldsymbol{\omega}\, \widehat{\mathbf{r}}_i^{t\top} \mathbf{a}_{\boldsymbol{\omega}} e^{\mathrm{i}2\pi\boldsymbol{\omega}^\top \mathbf{x}_i}. \tag{6}$$

Equation 6 shows that the signal-local Jacobian applies frequency-weighted scaling $\mathrm{i}2\pi\boldsymbol{\omega}$ to each residual-projected local Fourier mode $\widehat{\mathbf{r}}_i^{t\top} \mathbf{a}_{\boldsymbol{\omega}} e^{\mathrm{i}2\pi\boldsymbol{\omega}^\top \mathbf{x}_i}$ before mode synthesis at $\mathbf{x}_i$. Consequently, TaN sensitivity couples the evolving INR reconstruction state with frequency-amplified target structures, emphasizing coordinates where reconstruction residuals coincide with strong high-frequency target variations. This provides a spectral-bias interpretation of TaN sensitivity's informativeness for INR coordinate sampling, as MLPs tend to learn high-frequency components more slowly (Xu et al., 2019; Rahaman et al., 2019; Tancik et al., 2020; Ramasinghe et al., 2022; Lee et al., 2024).

To visualize this behavior, we consider a toy signal fitting task on $f^\star(x) = \sin(2\pi x) + 0.25\sin(2\pi q x)$ over $x \in [0, 2]$, where $q = 6$ controls the high-frequency component. We construct $D = 500$ evenly spaced observations and train an MLP with normalized coordinate inputs using a sampled subset $|\mathcal{B}_t| = 50$. TaN sensitivity is computed with $p = 2$. Figure 2 compares top-K TaN sensitivity sampling with uniform random sampling. While uniform sampling recovers the high-frequency component more slowly, TaN sensitivity concentrates samples where large residuals coincide with rapid target variations reflected by the signal-local Jacobian, accelerating refinement of unresolved oscillatory details. This behavior supports the spectral interpretation and illustrates the dynamic informativeness of TaN sensitivity for INR coordinate sampling.

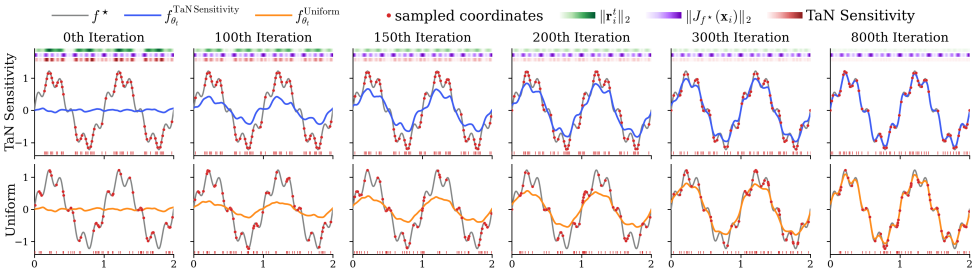


Figure 2: Training snapshots on a toy signal, comparing top-K TaN sensitivity and uniform random sampling. Heat strips visualize the residual norm, signal-local Jacobian norm, and TaN sensitivity.

3.4 TaN Scoring and Probabilistic Sampling

We now derive a stable coordinate-wise TaN score from the closed-form TaN sensitivity in Proposition 1. Since the factor p does not affect coordinate ranking, the TaN score is defined as

$$s_i^t = (\|\mathbf{r}_i^t\|_2 + \varepsilon)^{p-2} \left\| J_{f^\star}(\mathbf{x}_i)^\top \mathbf{r}_i^t \right\|_2, \tag{7}$$

where $\varepsilon > 0$ is a small stability constant and p controls the sharpness of residual-magnitude modulation. At iteration t, the scores $\{s_i^t\}_{i=1}^D$ are sanitized to finite nonnegative values, mean-normalized as $\{\bar{s}_i^t\}_{i=1}^D$, and smoothed using an exponential moving average (EMA) with factor $\beta \in [0, 1)$:

$$w_i^t = \beta w_i^{t-1} + (1-\beta)\bar{s}_i^t. \tag{8}$$

The smoothed weights are then transformed into sampling probabilities:

$$\pi_i^t = (1-\lambda)\frac{(w_i^t + \tau)^\alpha}{\sum_{j=1}^D (w_j^t + \tau)^\alpha} + \lambda\frac{1}{D}, \tag{9}$$

where $\alpha > 0$ controls score concentration, $\tau > 0$ is a small stability constant, and $\lambda \in [0, 1)$ is the uniform exploration rate. Together, $\boldsymbol{\pi}^t = (\pi_1^t, \ldots, \pi_D^t)$ defines the TaN sampling probabilities over all coordinate indices. At iteration t, STaNS forms an index batch $\mathcal{B}_t \subseteq \{1, \ldots, D\}$ by weighted sampling without replacement using $\boldsymbol{\pi}^t$, and updates the INR by

$$\theta_{t+1} = \theta_t - \eta_t \nabla_{\theta_t} \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} \mathcal{L}\left(f_{\theta_t}(\mathbf{x}_i), \mathbf{y}_i\right). \tag{10}$$

4 Experiments

4.1 Implementation Details

STaNS Settings. We set stability constants $\varepsilon = 10^{-3}$ and $\tau = 0.05$. We use linear schedulers $\beta : 0.7 \to 0.98$, $\alpha : 1.0 \to 1.6$, $\lambda : 0.15 \to 0.05$, and $\kappa_t : 20 \to 5$ across training, where κ_t denotes the refresh interval of TaN rescoring and coordinate resampling. We set $p = 4$. For audio and images, $J_{f^\star}(\mathbf{x}_i)$ is estimated once at initialization using finite differences over neighboring temporal samples or pixels; for shapes represented as signed distance fields (SDFs), it is estimated as the mean unit normal of the $k = 3$ nearest surface samples. Additional estimation details are provided in Appendix A.3.1. Sensitivity studies on $\{\beta, \alpha, \lambda, \kappa_t, p\}$ are presented in Sec. 4.6.

Baselines. We benchmark against leading INR sampling methods: **(1)** Uniform Random Sampling (Uniform), **(2)** EGRA (Gai et al., 2023), **(3)** Soft Mining (Kheradmand et al., 2024), **(4)** INT (Zhang et al., 2024a), **(5)** EVOS (Zhang et al., 2025a), **(6)** E.S. (Zhang et al., 2025b), and **(7)** NINT (Zhang et al., 2026), along with **(8)** full-coordinate training (Full). All baselines follow their official settings, with reported times including initialization, scoring, sampling, and optimization costs.

Training Settings. All experiments are conducted on an NVIDIA RTX 4090 (24 GB) and averaged over three random seeds. We employ a 6×256 SIREN MLP (Sitzmann et al., 2020) for audio and image data, and an 8×512 SIREN MLP for shape data, with a cosine learning-rate scheduler $\eta = 10^{-4} \to \eta = 10^{-5}$. We set sampling ratio $|\mathcal{B}_t|/D = 0.3$ for sampling-based methods. Additional evaluations on different network sizes and sampling ratios are provided in Appendix A.1 and A.2.

Table 1: Quantitative results of image fitting task, averaged over 24 images from the Kodak dataset.

Method	Time to PSNR ↓			Time to SSIM ↓			Time to LPIPS ↓			500th Iteration			1000th Iteration		
	25	30	35	0.7	0.8	0.9	0.3	0.2	0.1	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
Full	16.96	53.86	125.95	21.80	44.91	97.77	54.00	82.17	132.43	27.424	0.7316	0.4370	30.450	0.8249	0.2743
Uniform	6.29	28.01	75.28	7.80	18.43	48.96	19.72	30.16	65.88	27.043	0.7201	0.4510	29.512	0.8131	0.3087
EGRA	8.78	35.73	97.43	12.18	22.55	60.58	30.41	46.88	87.65	27.050	0.7205	0.4502	29.540	0.8137	0.3073
Soft	9.66	33.52	85.57	14.49	27.25	69.53	36.99	53.80	88.29	27.852	0.7407	0.4504	31.120	0.8411	0.2716
INT	8.67	28.34	89.56	13.42	25.34	71.05	35.09	51.68	77.06	27.849	0.7393	0.4557	31.149	0.8428	0.2706
EVOS	8.86	26.03	75.45	10.78	23.51	64.46	28.12	47.24	84.40	28.038	0.7458	0.4172	31.336	0.8494	0.2287
E.S.	6.78	20.65	63.36	7.61	19.71	46.16	20.19	30.19	62.72	27.899	0.7345	0.4270	31.596	0.8426	0.2336
NINT	6.28	19.89	61.55	7.09	16.40	52.43	17.10	26.267	56.48	28.073	0.7441	0.4211	31.374	0.8461	0.2360
STaNS	**4.81**	**18.44**	**54.19**	**6.38**	**13.52**	**38.78**	**14.48**	**21.96**	**45.67**	**28.501**	**0.7580**	**0.3935**	**32.054**	**0.8544**	**0.2199**

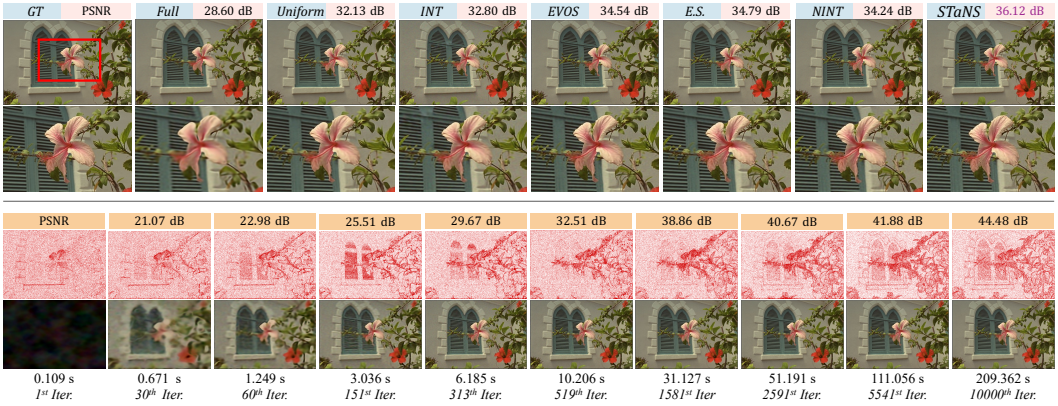


Figure 3: Visual comparison and sampling evolution on *kodim07* from the Kodak dataset. **Upper half**: reconstruction comparison across methods after 20 seconds of training. **Lower half**: STaNS sampled coordinates and reconstructed images across training.

4.2 Image Fitting Task

We conduct image fitting on the Kodak dataset (Eastman Kodak Company, 1999) and evaluate reconstruction quality using PSNR, SSIM (Wang et al., 2004), and LPIPS (Zhang et al., 2018). For each image, we train an INR for 10,000 iterations.

Table 1 reports first-hit times at quality thresholds and reconstruction quality at the 500th and 1000th iterations. STaNS consistently achieves the lowest first-hit times at all quality thresholds and the best iteration-budgeted quality, reducing time-to-quality by at least 23.41% at PSNR $= 25$ dB, 15.99% at SSIM $= 0.9$, and 19.14% at LPIPS $= 0.1$. Compared with INT, a residual-only sampler that greedily selects the largest-residual coordinates, STaNS achieves markedly lower first-hit times, including a 34.93% reduction at PSNR $= 30$ dB, highlighting how TaN sensitivity gains informativeness by coupling the target-neighborhood structure with the evolving INR state. Compared with full-coordinate training, it reduces first-hit time at LPIPS $= 0.2$ from 82.17 s to 21.96 s, representing a $3.74\times$ speedup. These results demonstrate that TaN sensitivity enables STaNS to sample more informative coordinate subsets throughout INR training, thereby establishing substantial acceleration. Later-checkpoint results and additional visualizations are provided in Appendix A.4.

Fig. 3 visualizes reconstruction quality and STaNS sampling dynamics on *kodim07*. In the upper half, STaNS achieves a leading PSNR of 36.12 dB after 20 seconds and preserves sharper local structures than baselines, especially clearer petal details in the zoomed region. The lower half further reveals the evolving nature of TaN sensitivity sampling. *STaNS first concentrates samples on the shutter before the 151st iteration, shifts toward petal regions around the 313th iteration after the shutter is well fitted, and later attends to window-frame blocks near the 5541st iteration.* This transition confirms that TaN sensitivity dynamically couples target-neighborhood structure with underfitted regions, yielding informative sampling for INR acceleration.

Table 2: Quantitative results of shape fitting task on the Stanford 3D Scanning Repository, reporting IoU and CHD at time and iteration (Iter.) checkpoints. CHD values are reported in 10^{-3} units.

Method	Metric	*Armadillo*				*Lucy*				*Happy Buddha*				*Dragon*			
		5 s	10 s	15 s	1000$^{\text{th}}$ Iter.	5 s	10 s	15 s	1000$^{\text{th}}$ Iter.	5 s	10 s	15 s	1000$^{\text{th}}$ Iter.	5 s	10 s	15 s	1000$^{\text{th}}$ Iter.
Full	IoU ↑	0.900	0.952	0.958	0.969	0.694	0.784	0.810	0.859	0.832	0.903	0.917	0.945	0.846	0.921	0.934	0.951
	CHD↓	7.819	6.333	6.194	6.083	22.764	18.710	17.808	13.616	9.199	7.326	7.036	6.168	8.822	6.867	6.693	6.260
Uniform	IoU ↑	0.931	0.953	0.962	0.966	0.793	0.817	0.841	0.836	0.901	0.915	0.920	0.934	0.920	0.935	0.940	0.961
	CHD↓	6.918	6.064	5.987	6.064	17.901	17.365	15.552	14.113	7.381	6.807	6.624	6.177	6.878	6.597	6.426	6.301
INT	IoU ↑	0.938	0.960	0.969	0.975	0.799	0.836	0.851	0.871	0.897	0.914	0.919	0.939	0.918	0.937	0.942	0.964
	CHD↓	6.687	6.045	6.013	5.941	16.405	14.371	14.212	12.168	7.426	6.737	6.603	6.236	6.880	6.622	6.410	6.206
EVOS	IoU ↑	0.929	0.958	0.964	0.972	0.810	0.818	0.854	0.852	0.896	0.919	0.923	0.938	0.912	0.937	0.945	0.969
	CHD↓	6.842	6.095	6.046	5.979	15.409	14.754	**13.598**	12.990	7.332	6.640	6.532	6.014	7.071	6.578	6.420	6.241
NINT	IoU ↑	0.930	0.961	0.965	0.977	0.786	0.820	0.837	0.845	0.896	0.917	0.925	0.941	0.916	0.932	0.944	0.968
	CHD↓	6.786	6.086	5.949	**5.888**	16.190	15.448	15.099	13.877	7.445	6.623	6.613	5.953	6.923	6.633	6.415	6.198
STaNS	IoU ↑	**0.959**	**0.970**	**0.973**	**0.991**	**0.810**	**0.850**	**0.860**	**0.879**	**0.919**	**0.932**	**0.941**	**0.948**	**0.942**	**0.956**	**0.961**	**0.973**
	CHD↓	**6.072**	**5.924**	**5.911**	5.905	**14.923**	**14.131**	13.638	**11.520**	**6.580**	**6.335**	**5.991**	**5.867**	**6.544**	**6.234**	**6.181**	**6.163**

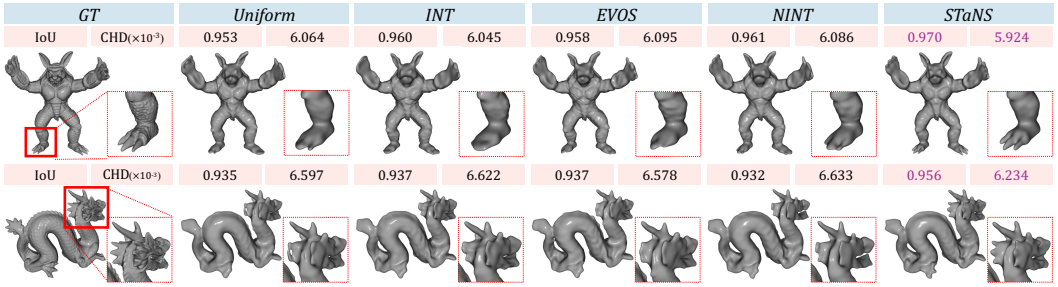


Figure 4: Visual comparison of shape reconstruction on *Armadillo* and *Dragon* from the Stanford 3D Scanning Repository after 10 seconds of training. STaNS preserves finer geometric details, especially around the claw and horn regions.

4.3 Shape Fitting Task

We conduct shape fitting on *Armadillo*, *Lucy*, *Happy Buddha*, and *Dragon* from the Stanford 3D Scanning Repository (Stanford Computer Graphics Laboratory, 2007). Each shape is represented by an SDF and trained for 3000 iterations, with 20,000 training and 100,000 evaluation points sampled around the surfaces. Following EVOS (Zhang et al., 2025a), we compare STaNS with Full, Uniform, INT, EVOS, and NINT, excluding methods incompatible with SDF reconstruction. Under the same SDF sampling protocol, half the points are sampled with coarse Laplacian noise of variance 0.1, and half with fine variance 0.001, covering both coarse and fine surface neighborhoods. The training pool is refreshed every 200 iterations, with all samplers reinitialized on the new pool, while the evaluation pool remains fixed. The pool refresh time is excluded from reported times.

Table 2 reports Intersection over Union (IoU) and Chamfer Distance (CHD) at time and iteration checkpoints. Across four shapes, STaNS achieves the best IoU at all 16 checkpoints and the lowest CHD at 14 of 16, demonstrating consistent gains across both time and iteration budgets. The gains are especially clear on *Dragon* at 10 seconds, where STaNS improves IoU from 0.937 to 0.956 and reduces CHD from 6.578×10^{-3} to 6.234×10^{-3}. These results highlight the effectiveness of STaNS for accelerated implicit surface reconstruction via informative sampling.

Fig. 4 visually compares SDF reconstruction on *Armadillo* and *Dragon* after 10 seconds of training. Compared with baselines, STaNS more clearly preserves thin geometric structures and high-curvature details, especially around the claws of *Armadillo* and the horns of *Dragon*. The clearer local geometry is consistent with TaN sensitivity directing sampling toward regions that benefit fine-scale surface recovery.

Table 3: Quantitative results of the audio fitting task on the LibriSpeech *test-clean* split.

Method	SI-SNR ↑					STOI ↑					PESQ ↑				
	5 s	10 s	20 s	500th Iter.	1000th Iter.	5 s	10 s	20 s	500th Iter.	1000th Iter.	5 s	10 s	20 s	500th Iter.	1000th Iter.
Full	2.280	7.944	12.296	0.318	5.681	0.626	0.723	0.816	0.592	0.682	1.293	1.405	1.445	1.240	1.364
Uniform	2.786	7.112	11.223	-1.461	2.934	0.642	0.724	0.793	0.574	0.644	1.272	1.325	1.364	1.178	1.274
INT	5.051	9.646	13.275	0.263	5.572	0.664	0.757	0.837	0.585	0.672	1.218	1.276	1.344	1.147	1.221
EVOS	5.380	10.350	14.111	-0.875	5.442	0.695	0.788	0.840	0.596	0.696	1.256	1.285	1.325	1.189	1.258
NINT	5.959	10.890	13.970	1.403	6.457	0.676	0.775	0.851	0.604	0.688	1.305	1.405	1.471	**1.211**	1.318
STaNS	**7.927**	**11.885**	**14.118**	**2.811**	**8.114**	**0.717**	**0.806**	**0.856**	**0.630**	**0.721**	**1.310**	**1.411**	**1.486**	1.200	**1.324**

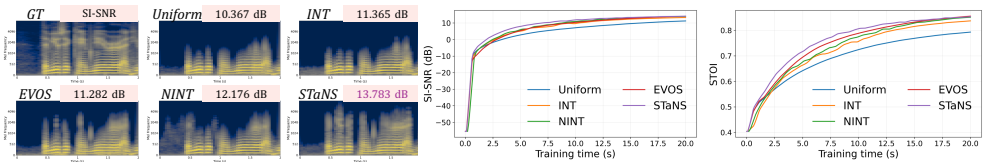


Figure 5: Mel-spectrogram reconstruction after 10 seconds and training curves for both SI-SNR and STOI on utterance *1089-134686-0005* from the LibriSpeech *test-clean* split.

Table 4: Comparison of sampling components and overhead (ms) on Kodak image fitting.

Method	Sampling Components			Overhead (ms)		
	Static Target Structure	Dynamic Fitting Residual	Dynamic INR Structure	Static Precompute	Dynamic Update	Sampling
INT	–	✓	–	–	21.96	0.20
NINT	–	✓	✓	–	256.30	2.72
EVOS	✓	✓	✓	1.84	23.60	22.10
E.S.	✓	–	–	0.87	–	2.12
STaNS	✓	✓	–	1.46	22.11	9.57

4.4 Audio Fitting Task

We conduct audio fitting on the *test-clean* split of LibriSpeech (Panayotov et al., 2015), using the first 2 seconds of each clip. Each signal is trained for 5000 iterations and evaluated by SI-SNR (Le Roux et al., 2019), STOI (Taal et al., 2011), and PESQ (Rix et al., 2001). Following EVOS (Zhang et al., 2025a), we compare STaNS with Full, Uniform, INT, EVOS, and NINT, excluding baselines incompatible with audio. Table 3 reports reconstruction quality averaged over 2,620 clips at different time and iteration checkpoints. Specifically, STaNS achieves the best quality among sampling baselines in most cases, including SI-SNR gains of 1.968 dB and 1.657 dB at the 5-second and 1000-iteration checkpoints, respectively. These results show that STaNS selects more informative temporal coordinates, enabling faster recovery of perceptually meaningful audio structure.

Fig. 5 visually compares audio reconstruction by Mel-spectrograms together with SI-SNR and STOI curves. After 10 seconds, STaNS more faithfully recovers high-frequency spectral bands, especially where baselines exhibit attenuated patterns. The SI-SNR and STOI curves further show that STaNS maintains a consistent advantage in quality improvements across training. These results corroborate the spectral interpretation, showing that TaN sensitivity couples the target-neighborhood structure with the evolving INR state to recover unresolved fine-scale audio dynamics more efficiently.

4.5 Sampling Overhead Evaluation

Table 4 compares criterion components and CUDA-synchronized overhead on Kodak image fitting with NVIDIA RTX 4090 (24 GB), averaged over 24 images (768×512). **Static Precompute** measures one-time target structure computation, while **Dynamic Update** and **Sampling** measure criterion refresh and fresh-subset generation on active sampling iterations. Notably, STaNS incurs only 31.68 ms per active event, 87.77% lower than NINT and 30.68% lower than EVOS. It nearly matches INT's dynamic update cost (22.11 vs. 21.96 ms), while leveraging a 1.46 ms precompute to couple target-neighborhood structure with the evolving INR state for informative INR sampling.

Table 5: Hyperparameter sensitivity and component ablation studies on Kodak image fitting task.

Group	Setting	Time to PSNR ↓			Time to SSIM ↓			Time to LPIPS ↓			500th Iteration			1000th Iteration		
		25	30	35	0.70	0.80	0.90	0.3	0.2	0.1	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
	Default	4.81	18.44	**54.19**	6.38	13.52	**38.78**	14.48	21.96	45.67	28.501	0.7580	0.3935	**32.054**	**0.8544**	0.2199
$J_{f^\star}$	*w/o Jac.*	6.63	25.77	76.46	10.95	22.52	65.71	26.82	48.30	71.55	27.792	0.7490	0.4199	30.908	0.8469	0.2311
$\mathbf{r}_i^t$	*w/o Res.*	8.43	31.95	93.16	17.04	25.67	73.56	36.56	56.33	89.78	26.963	0.7206	0.4567	29.553	0.8097	0.2858
p	1§	5.79	22.70	63.15	7.47	16.42	55.55	17.36	27.66	62.38	27.827	0.7345	0.4322	30.757	0.8243	0.2666
	2	5.00	19.66	61.32	6.64	14.35	50.54	15.18	23.55	51.82	28.410	0.7534	0.4039	31.554	0.8416	0.2283
	3	4.81	18.56	55.26	6.38	13.62	44.23	14.60	22.31	48.21	28.550	0.7585	0.3941	31.714	0.8468	0.2186
	6	4.99	19.00	55.23	6.65	14.21	40.70	15.33	23.14	45.66	28.239	0.7507	0.4024	31.236	0.8411	0.2344
β	0.0†	5.65	24.48	68.42	7.30	16.21	48.68	17.14	27.14	63.28	28.032	0.7423	0.4208	30.721	0.8278	0.2632
	0.8	4.86	**17.39**	56.35	6.37	13.54	42.29	**14.41**	**21.73**	**44.80**	28.486	0.7574	0.3941	31.733	0.8483	**0.2155**
λ	0.0‡	**4.78**	17.43	56.35	6.37	13.69	45.95	14.63	23.05	49.85	**28.561**	0.7588	0.3962	31.714	0.8463	0.2227
	0.2	4.81	18.32	54.49	6.43	**13.49**	41.81	14.59	22.01	44.87	28.469	0.7575	0.3928	31.558	0.8472	0.2203
α	0.5	4.99	19.30	56.32	6.47	13.67	39.83	14.68	22.77	46.03	28.395	0.7557	0.3962	31.496	0.8466	0.2242
	1.0	4.84	18.13	55.03	**6.36**	13.50	41.99	14.56	22.22	46.42	28.512	0.7579	0.3936	31.598	0.8470	0.2212
	2.0	5.32	20.67	60.54	7.08	15.35	43.34	16.64	25.65	49.78	27.897	0.7407	0.4168	30.678	0.8303	0.2549
κ_t	5	5.88	21.59	70.76	7.95	16.48	48.51	18.60	27.96	58.65	28.433	0.7581	0.3950	31.668	0.8521	0.2224
	10	5.11	18.87	60.32	6.89	14.46	43.94	16.12	24.22	50.72	28.558	**0.7597**	**0.3917**	31.767	0.8513	0.2196
	20	4.83	18.45	61.17	6.44	13.56	43.70	14.69	22.44	48.53	28.518	0.7588	0.3921	31.687	0.8491	0.2167

† is the variant without EMA. ‡ is the variant without uniform exploration. § is the variant without residual-magnitude modulation.

4.6 Sensitivity and Ablation Studies

We conduct sensitivity and ablation studies on Kodak image fitting, averaged over 24 images. Table 5 shows that the default performs best overall, while variants such as $\beta = 0.8$ remain competitive, demonstrating the robustness to hyperparameter choices and the effectiveness of each component.

Contribution of Target-Neighborhood Structure and Reconstruction Residual. We construct two ablation variants: *w/o Jac.*, which removes the signal-local Jacobian term and uses $s_i^t = (\|\mathbf{r}_i^t\|_2+\varepsilon)^{p-2}\|\mathbf{r}_i^t\|_2$, and *w/o Res.*, which removes residual dependence and uses $s_i^t = \|J_{f^\star}(\mathbf{x}_i)^\top\|_2$, while keeping the sampling pipeline unchanged. Removing either component consistently degrades metrics, with the time to PSNR $= 35$ dB increasing from 54.19 s to 76.46 s and 93.16 s, respectively. These results validate that both target-neighborhood structure and the evolving INR state contribute to informative sampling, with their coupling in TaN sensitivity yielding the strongest acceleration.

TaN Mismatch Order. For $p = 1$ and $\varepsilon \ll \|\mathbf{r}_i^t\|_2$, the score approximates $\|J_{f^\star}(\mathbf{x}_i)^\top \widehat{\mathbf{r}}_i^t\|_2$, largely removing residual magnitude effects and degrading performance. Increasing p strengthens this modulation, with $p = 3$ yielding the highest 500-iteration PSNR and $p = 4$ the best overall balance, while $p = 6$ generally degrades performance. This highlights the importance of balanced residual-magnitude modulation in TaN sensitivity for informative sampling.

Sampling Hyperparameters. The EMA factor β and exploration rate λ stabilize sampling, while concentration α and refresh interval κ_t control adaptivity. $\beta = 0.8$ performs competitively, whereas removing EMA ($\beta = 0.0$) slows quality improvement. Likewise, $\lambda = 0.0$ slightly improves early thresholds but degrades later performance, highlighting the benefit of exploration. Excessive concentration ($\alpha = 2.0$) reduces diversity, while fixed refresh intervals can improve iteration-conditioned metrics but are less consistent across time-to-quality thresholds. Overall, the default schedules balance sampling diversity, rescoring overhead, and adaptation to the evolving INR state.

5 Conclusion

We introduce STaNS, a lightweight INR sampling strategy that views each observed coordinate as a target-neighborhood representative of the underlying continuous target function. By defining TaN mismatch and TaN sensitivity, we derive a closed-form criterion that couples the target-neighborhood structure with the evolving INR state, yielding coordinate-wise TaN scores for probabilistic sampling of informative coordinate subsets for efficient INR training. Across diverse tasks and metrics, STaNS reduces time-to-quality by up to 23.41% over existing INR samplers and achieves up to a $3.74\times$ speedup over full-coordinate training, demonstrating that sampling by TaN sensitivity delivers state-of-the-art INR training acceleration.

AI USE STATEMENT

In this work, we used generative AI tools only to assist with code implementation. All figures and tables were manually designed and assembled by the authors, and all text was written and polished by the authors. Generative AI was not used to conceive the research ideas or generate experimental results. We reviewed all AI-assisted code for accuracy and consistency with the intended scientific content and implementation. The authors take full responsibility for the final manuscript, including all text, claims, code, figures, and tables.

REPRODUCIBILITY STATEMENT

The Supplementary Material provides the implementation source code and the three random seeds used in our experiments. All datasets are publicly available and properly cited. We will publicly release the source code and model checkpoints after the review decisions are announced.

REFERENCES

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5855–5864, 2021. 2

Yizhak Ben-Shabat, Chamin Hewa Koneputugodage, Sameera Ramasinghe, and Stephen Gould. Neural experts: Mixture of experts for implicit neural representations. *Advances in Neural Information Processing Systems*, 37:101641–101670, 2024. 2

Amandine Brunetto, Sascha Hornauer, and Fabien Moutarde. Neraf: 3d scene infused neural radiance and acoustic fields. In *International Conference on Learning Representations*, volume 2025, pp. 92876–92899, 2025. 2

Zhicheng Cai, Hao Zhu, Qiu Shen, Xinran Wang, and Xun Cao. Batch normalization alleviates the spectral bias in coordinate networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 25160–25171, 2024. 2

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European conference on computer vision*, pp. 333–350. Springer, 2022. 1, 2

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. 2

Xiang Chen, Jinshan Pan, and Jiangxin Dong. Bidirectional multi-scale implicit neural representations for image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25627–25636, 2024. 2

Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pp. 170–187. Springer, 2022. 1

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*, 2022. 2

Eastman Kodak Company. Kodak lossless true color image suite. http://r0k.us/graphics/kodak/, 1999. [Accessed 14-08-2023]. 6

Zhenbiao Gai, Zhenyang Liu, Min Tan, Jiajun Ding, Jun Yu, Mingzhao Tong, and Junqing Yuan. Egra-nerf: Edge-guided ray allocation for neural radiance fields. *Image and Vision Computing*, 134:104670, 2023. 2, 5

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 2

Jeffrey Gu and Serena Yeung. Foundation models secretly understand neural network weights: Enhancing hypernetwork architectures with foundation models. In *International Conference on Learning Representations*, volume 2025, pp. 58565–58577, 2025. 2

Xin Huang, Qi Zhang, Ying Feng, Hongdong Li, and Qing Wang. Inverting the imaging process by learning an implicit camera model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21456–21465, 2023. 2

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pp. 2525–2534. PMLR, 2018. 1

Shakiba Kheradmand, Daniel Rebain, Gopal Sharma, Hossam Isack, Abhishek Kar, Andrea Tagliasacchi, and Kwang Moo Yi. Accelerating neural field training via soft mining. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20071–20080, 2024. 1, 2, 5

Jonathan Le Roux, Scott Wisdom, Hakan Erdogan, and John R Hershey. Sdr–half-baked or well done? In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 626–630. IEEE, 2019. 8

Joo Chan Lee, Daniel Rho, Seungtae Nam, Jong Hwan Ko, and Eunbyung Park. Coordinate-aware modulation for neural fields. In *International Conference on Learning Representations*, volume 2024, pp. 22374–22391, 2024. 4

Ruofan Liang, Hongyi Sun, and Nandita Vijaykumar. Coordx: Accelerating implicit neural representation with a split mlp architecture. *arXiv preprint arXiv:2201.12425*, 2022. 2

Ke Liu, Feng Liu, Haishuai Wang, Ning Ma, Jiajun Bu, and Bo Han. Partition speeds up learning implicit neural representations based on exponential-increase hypothesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5474–5483, 2023. 1, 2

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *Advances in Neural Information Processing Systems*, 33:15651–15663, 2020. 2

Qiang Liu, Mengyu Chu, and Nils Thuerey. Config: Towards conflict-free training of physics informed neural networks. In *International Conference on Learning Representations*, volume 2025, pp. 59531–59566, 2025. 2

Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14214–14223, 2021. 1, 2

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. 1

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

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*, 2015. 8

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019. 1

Francesca Pistilli, Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Signal compression via neural implicit representations. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3733–3737. IEEE, 2022. 2

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. 4

Sameera Ramasinghe, Lachlan E MacDonald, and Simon Lucey. On the frequency-bias of coordinate-mlps. *Advances in Neural Information Processing Systems*, 35:796–809, 2022. 4

Daniel Rebain, Wei Jiang, Soroosh Yazdani, Ke Li, Kwang Moo Yi, and Andrea Tagliasacchi. Derf: Decomposed radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14153–14161, 2021. 2

Antony W Rix, John G Beerends, Michael P Hollier, and Andries P Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pp. 749–752. IEEE, 2001. 8

Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representation. In *European Conference on Computer Vision*, pp. 318–333. Springer, 2022. 2

Jonathan Richard Schwarz, Jihoon Tack, Yee Whye Teh, Jaeho Lee, and Jinwoo Shin. Modality-agnostic variational compression of implicit neural representations. *arXiv preprint arXiv:2301.09479*, 2023. 2

Junwon Seo, Sangyoon Lee, Kwang In Kim, and Jaeho Lee. In search of a data transformation that accelerates neural field training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4830–4839, 2024. 2

Liming Shen, Liang Deng, Chongke Bi, Yu Wang, Xinhai Chen, Yueqing Wang, and Jie Liu. Peinr: A physics-enhanced implicit neural representation for high-fidelity flow field reconstruction. In *Forty-second International Conference on Machine Learning*, 2025. 1

Kexuan Shi, Xingyu Zhou, and Shuhang Gu. Improved implicit neural representation with fourier reparameterized training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25985–25994, 2024. 2

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, 2020. 1, 5

Kevin D Smith, Francesco Seccamonte, Ananthram Swami, and Francesco Bullo. Physics-informed implicit representations of equilibrium network flows. *Advances in Neural Information Processing Systems*, 35:7211–7221, 2022. 2

Stanford Computer Graphics Laboratory. The stanford 3d scanning repository. https://graphics.stanford.edu/data/3Dscanrep/, 2007. 7

Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *European conference on computer vision*, pp. 74–91. Springer, 2022. 1

Kun Su, Mingfei Chen, and Eli Shlizerman. Inras: Implicit neural representation for audio scenes. *Advances in Neural Information Processing Systems*, 35:8144–8158, 2022. 1

Cees H Taal, Richard C Hendriks, Richard Heusdens, and Jesper Jensen. An algorithm for intelligibility prediction of time–frequency weighted noisy speech. *IEEE Transactions on audio, speech, and language processing*, 19(7):2125–2136, 2011. 8

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. 4

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. 1

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(11):9426–9437, 2024. 2

Shu Wang, Yanbo Gao, Shuai Li, Chong Lv, Xun Cai, Chuankun Li, Hui Yuan, and Jinglin Zhang. Metricgrids: Arbitrary nonlinear approximation with elementary metric grids based implicit neural representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21381–21391, 2025. 2

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6

Dejia Xu, Peihao Wang, Yifan Jiang, Zhiwen Fan, and Zhangyang Wang. Signal processing for implicit neural representations. *Advances in Neural Information Processing Systems*, 35:13404–13418, 2022a. 1

Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5438–5448, 2022b. 2

Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *International Conference on Neural Information Processing*, pp. 264–274. Springer, 2019. 4

Maizhe Yang, Kaiyuan Tang, and Chaoli Wang. Meta-inr: Efficient encoding of volumetric data via meta-learning implicit neural representation. In *2025 IEEE 18th Pacific Visualization Conference (PacificVis)*, pp. 246–251. IEEE, 2025. 2

Runzhao Yang. Tinc: Tree-structured implicit neural compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18517–18526, 2023. 1

Shuzhou Yang, Moxuan Ding, Yanmin Wu, Zihan Li, and Jian Zhang. Implicit neural representation for cooperative low-light image enhancement. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 12918–12927, 2023. 2

Chen Zhang, Steven Tin Sui Luo, Jason Chun Lok Li, Yik-Chung Wu, and Ngai Wong. Nonparametric teaching of implicit neural representations. *arXiv preprint arXiv:2405.10531*, 2024a. 1, 2, 5

Chen Zhang, Wei Zuo, Bingyang Cheng, Yikun Wang, Wei-Bin Kou, Yik-Chung Wu, and Ngai Wong. Ntk-guided implicit neural teaching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17248–17258, 2026. 1, 2, 3, 5

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 586–595, 2018. 6

Tianjing Zhang, Yuhui Quan, and Hui Ji. Cross-scale self-supervised blind image deblurring via implicit neural representation. In *NeurIPS*, 2024b. 2

Weixiang Zhang, Shuzhao Xie, Chengwei Ren, Siyi Xie, Chen Tang, Shijia Ge, Mingzi Wang, and Zhi Wang. Evos: Efficient implicit neural training via evolutionary selector. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30472–30482, 2025a. 1, 2, 5, 7, 8

Weixiang Zhang, Wei Yao, Shuzhao Xie, Shijia Ge, Chen Tang, and Zhi Wang. Expansive supervision for neural radiance fields. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2025b. 1, 2, 5

A APPENDIX

A.1 ADAPTABILITY ACROSS NETWORK SIZES

Table 6: Quantitative comparison across network sizes on image fitting task, reporting first-hit times (seconds) to reach quality thresholds and reconstruction quality at iteration checkpoints. Results are averaged over 24 images from the Kodak dataset.

Network	Method	Time to PSNR ↓			Time to SSIM ↓			Time to LPIPS ↓			500th Iteration			1000th Iteration		
		24	26	28	0.6	0.7	0.8	0.5	0.4	0.3	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
6x256*	Uniform	4.64	9.33	18.82	3.68	7.80	18.43	7.94	12.83	19.72	27.043	0.7201	0.4510	29.512	0.8131	0.3087
	INT	7.18	12.68	19.87	6.78	13.42	25.34	12.30	19.06	35.09	27.849	0.7393	0.4557	31.149	0.8428	0.2706
	EVOS	7.34	11.92	17.28	5.10	10.78	23.51	10.39	16.19	28.12	28.038	0.7458	0.4172	31.336	0.8494	0.2287
	E.S.	4.37	7.38	11.99	3.81	7.61	19.71	7.62	12.54	20.19	27.899	0.7345	0.4270	31.596	0.8426	0.2336
	NINT	4.60	7.19	12.11	3.75	7.09	16.40	7.05	11.96	17.10	28.073	0.7441	0.4211	31.374	0.8461	0.2360
	STaNS	**3.67**	**6.20**	**10.40**	**3.11**	**6.38**	**13.52**	**6.49**	**9.75**	**14.48**	**28.501**	**0.7580**	**0.3935**	**32.054**	**0.8544**	**0.2199**
6x128	Uniform	3.69	6.35	10.09	2.63	5.98	14.96	6.11	9.97	17.36	25.714	0.6668	0.5514	27.503	0.7384	0.4313
	INT	6.34	10.52	16.20	5.02	9.14	21.16	10.98	17.97	31.45	26.379	0.6828	0.5554	28.435	0.7588	0.4235
	EVOS	6.31	9.49	15.53	4.36	8.99	20.85	8.77	15.54	26.60	26.309	0.6789	0.5446	28.551	0.7600	0.3964
	E.S.	3.77	6.27	10.10	2.95	6.79	15.88	6.30	10.06	17.31	26.405	0.6776	0.5347	28.417	0.7485	0.4036
	NINT	3.79	5.99	9.60	2.62	6.06	14.23	5.71	9.38	15.05	26.614	0.6879	0.5259	28.723	0.7625	0.3952
	STaNS	**2.80**	**4.62**	**8.36**	**2.20**	**5.37**	**13.08**	**4.93**	**7.95**	**13.78**	**26.833**	**0.6971**	**0.5069**	**28.933**	**0.7668**	**0.3727**
4x256	Uniform	5.19	10.61	14.79	3.90	8.05	20.95	8.49	12.80	21.55	25.478	0.6602	0.5563	27.801	0.7598	0.3934
	INT	7.24	13.08	17.31	7.34	13.80	26.91	11.80	17.77	34.30	26.216	0.6792	0.5620	29.078	0.7882	0.3680
	EVOS	7.65	14.12	16.92	5.87	11.82	25.41	10.82	15.63	29.58	26.113	0.6723	0.5473	29.049	0.7851	0.3388
	E.S.	4.57	8.94	11.67	3.80	7.53	19.62	7.51	10.71	17.11	26.514	0.6861	0.5132	29.325	0.7884	0.3258
	NINT	4.59	8.71	11.32	3.51	7.33	17.70	6.79	9.84	16.35	26.602	0.6923	0.5119	29.310	0.7949	0.3227
	STaNS	**3.84**	**7.90**	**10.02**	**3.11**	**6.69**	**15.95**	**6.31**	**8.96**	**14.38**	**27.013**	**0.7083**	**0.4765**	**29.731**	**0.7986**	**0.2998**
8x64	Uniform	3.45	4.58	-	2.77	4.02	-	7.50	11.89	-	24.393	0.6252	0.6302	25.978	0.6771	0.5355
	INT	5.36	6.63	-	5.04	6.69	-	11.32	18.65	-	24.757	0.6267	0.6432	26.336	0.6776	0.5481
	EVOS	5.74	6.34	-	4.81	6.34	-	9.41	16.67	-	24.785	0.6282	0.6200	26.523	0.6803	0.5230
	E.S.	4.46	5.15	-	4.09	5.02	-	7.32	11.61	-	24.575	0.6223	0.6294	26.264	0.6700	0.5292
	NINT	5.03	5.89	-	4.66	5.33	-	7.58	10.12	-	24.905	0.6306	0.6183	26.555	0.6806	0.5142
	STaNS	**2.94**	**3.45**	-	**2.75**	**3.70**	-	**6.52**	**9.73**	-	**25.072**	**0.6370**	**0.6019**	**26.687**	**0.6868**	**0.5033**

$*$ indicates the default setting in Table 1.

We further examine the adaptability of STaNS beyond the default 6×256 SIREN MLP size by fitting Kodak images for 5,000 iterations with three smaller architectures: 6×128, 4×256, and 8×64.

Table 6 reports first-hit times at PSNR, SSIM, and LPIPS thresholds and reconstruction quality at the 500th and 1000th iterations, where a dash indicates that the corresponding threshold is not reached within the training budget. Across the default and three reduced configurations, STaNS consistently achieves the lowest first-hit times at every reachable threshold and the best reconstruction quality at both iteration checkpoints. On the 6×128 network, STaNS reduces first-hit times by up to 24.12% and achieves the best 1000th-iteration quality with 28.933 dB PSNR, 0.7668 SSIM, and 0.3727 LPIPS. On the 4×256 network, it reduces first-hit times by up to 15.97%, while on the compact 8×64 network, the reduction reaches 24.67%. These consistent gains exhibit the adaptability of STaNS across network sizes, demonstrating its effectiveness for accelerating INR training under variations in network width and depth.

Notably, at the early PSNR $= 24$ dB threshold, Uniform is faster than INT and EVOS on all three reduced networks, while also outperforming E.S. and NINT on the 6×128 and 8×64 networks. This reveals a trade-off between sampling informativeness and early-stage computational cost, especially for smaller networks where optimization becomes inexpensive and sampling accounts for a larger fraction of the training time. INT relies on residual-only greedy sampling, EVOS introduces additional Laplacian-aware evolutionary operations, NINT incurs costly NTK-weighted updates, and E.S. uses fixed Sobel-derived edge anchors without adapting to the evolving INR state. In contrast, STaNS balances this trade-off via the lightweight yet informative TaN sensitivity. Consequently, STaNS reaches PSNR $= 24$ dB in 2.80 s, 3.84 s, and 2.94 s on the 6×128, 4×256, and 8×64 networks, respectively, while remaining the strongest method at all reachable later thresholds and fixed-iteration checkpoints. These results demonstrate that coupling the once-computed target-neighborhood structure with the evolving INR state enables TaN sensitivity to achieve both efficient and effective coordinate sampling for substantial INR training acceleration.

A.2 Adaptability Across Sampling Ratios

Table 7: Quantitative comparison across sampling ratios on image fitting task, reporting first-hit times (seconds) to reach quality thresholds and reconstruction quality at iteration checkpoints. Results are averaged over 24 images from the Kodak dataset.

Ratio	Method	Time to PSNR ↓			Time to SSIM ↓			Time to LPIPS ↓			500th Iteration			1000th Iteration		
		25	30	35	0.7	0.8	0.9	0.3	0.2	0.1	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
0.2	Uniform	4.55	17.17	**49.16**	5.41	12.74	32.37	13.98	22.41	45.65	26.859	0.7146	0.4583	29.063	0.8011	0.3279
	INT	7.50	20.94	69.47	8.97	17.20	48.11	23.97	37.92	62.58	27.738	0.7294	0.4777	30.949	0.8273	0.2959
	EVOS	7.05	18.37	65.63	8.95	18.11	44.45	22.70	35.94	59.92	27.238	0.7209	0.4526	30.195	0.8216	0.2754
	E.S.	4.21	16.05	56.12	5.54	12.90	34.92	12.55	20.21	48.04	27.736	0.7241	0.4412	30.480	0.8165	0.2749
	NINT	3.91	16.87	54.32	5.56	12.60	32.45	11.27	19.74	44.52	27.964	0.7363	0.4345	31.128	**0.8359**	0.2627
	STaNS	**3.67**	**15.99**	49.71	**4.82**	**11.89**	**30.48**	**10.33**	**18.65**	**42.65**	**28.364**	**0.7489**	**0.4002**	**31.432**	0.8319	**0.2427**
0.3*	Uniform	6.29	28.01	75.28	7.80	18.43	48.96	19.72	30.16	65.88	27.043	0.7201	0.4510	29.512	0.8131	0.3087
	INT	8.67	28.34	89.56	13.42	25.34	71.05	35.09	51.68	77.06	27.849	0.7393	0.4557	31.149	0.8428	0.2706
	EVOS	8.86	26.03	75.45	10.78	23.51	64.46	28.12	47.24	84.40	28.038	0.7458	0.4172	31.336	0.8494	0.2287
	E.S.	6.78	20.65	63.36	7.61	19.71	46.16	20.19	30.19	62.72	27.899	0.7345	0.4270	31.596	0.8426	0.2336
	NINT	6.28	19.89	61.55	7.09	16.40	52.43	17.10	26.267	56.48	28.073	0.7441	0.4211	31.374	0.8461	0.2360
	STaNS	**4.81**	**18.44**	**54.19**	**6.38**	**13.52**	**38.78**	**14.48**	**21.96**	**45.67**	**28.501**	**0.7580**	**0.3935**	**32.054**	**0.8544**	**0.2199**
0.5	Uniform	9.22	30.29	78.76	11.81	24.49	53.48	29.50	44.36	80.70	27.247	0.7265	0.4424	30.031	0.8252	0.2893
	INT	11.74	36.69	95.01	16.00	33.93	77.42	42.85	65.06	104.62	27.733	0.7406	0.4358	30.873	0.8443	0.2604
	EVOS	11.66	34.67	89.16	15.45	31.61	71.78	38.80	58.48	100.50	28.071	0.7426	0.4284	31.094	0.8537	0.2595
	E.S.	8.37	26.26	74.59	10.97	22.37	52.46	25.74	38.41	69.61	27.949	0.7424	0.4182	31.147	0.8431	0.2502
	NINT	8.31	27.32	72.26	11.05	23.76	51.22	24.84	38.24	70.64	27.972	0.7454	0.4188	31.184	0.8458	0.2489
	STaNS	**7.61**	**24.49**	**66.85**	**10.11**	**20.66**	**47.51**	**24.06**	**35.80**	**64.58**	**28.203**	**0.7542**	**0.4024**	**31.546**	**0.8559**	**0.2275**
0.7	Uniform	12.38	39.44	98.97	15.90	32.78	70.70	39.34	58.74	100.83	27.334	0.7294	0.4391	30.286	0.8316	0.2787
	INT	14.74	50.91	123.77	20.96	41.59	90.87	52.44	74.55	133.15	27.529	0.7353	0.4349	30.455	0.8336	0.2717
	EVOS	14.28	48.51	117.68	19.84	39.16	86.33	49.19	68.22	124.90	27.814	0.7357	**0.4178**	30.717	0.8394	0.2657
	E.S.	11.72	36.20	94.70	15.19	30.79	67.81	36.13	54.17	93.41	27.756	0.7398	0.4241	31.019	0.8432	0.2529
	NINT	12.11	35.99	94.44	15.36	31.20	67.18	36.34	54.00	92.56	27.788	0.7409	0.4244	31.148	0.8465	0.2521
	STaNS	**11.38**	**35.59**	**91.12**	**14.67**	**30.10**	**65.83**	**35.52**	**52.86**	**90.85**	**27.837**	**0.7440**	0.4194	**31.272**	**0.8582**	**0.2453**

$*$ indicates the default setting in Table 1.

We further examine the adaptability of STaNS beyond the default sampling ratio $|\mathcal{B}_t|/D = 0.3$ by fitting Kodak images for 10,000 iterations with smaller and larger ratios $|\mathcal{B}_t|/D \in \{0.2, 0.5, 0.7\}$.

Table 7 reports first-hit times at PSNR, SSIM, and LPIPS thresholds and quality at the 500th and 1000th iterations. Across sampling ratios, STaNS achieves the lowest first-hit times at 35 of 36 thresholds and the best quality in 22 of 24 fixed-iteration metrics. Its maximum first-hit reductions over the strongest competing baseline reach 10.91%, 23.41%, 8.42%, and 3.52% at ratios 0.2, 0.3, 0.5, and 0.7, respectively. At the 1000th iteration, STaNS achieves the best PSNR and LPIPS at every ratio; at 0.5 and 0.7, it improves PSNR by 0.362 dB and 0.124 dB while reducing LPIPS by 8.60% and 2.70%. These results demonstrate consistent effectiveness across coordinate budgets.

Notably, at the small sampling ratio $|\mathcal{B}_t|/D = 0.2$, Uniform remains competitive at late-stage thresholds because lower optimization cost makes sampling overhead more influential, reaching PSNR $= 35$ dB slightly faster than STaNS (49.16 s vs. 49.71 s). Under this constrained budget, most sampling baselines fail to consistently translate fixed-iteration gains into faster threshold attainment: NINT outperforms Uniform at LPIPS $= 0.1$ but is slightly slower at SSIM $= 0.9$, while INT, EVOS, and E.S. are slower at both. In contrast, STaNS surpasses Uniform at most late-stage thresholds, reaching SSIM $= 0.9$ in 30.48 s vs. 32.37 s and LPIPS $= 0.1$ in 42.65 s vs. 45.65 s, reductions of 5.84% and 6.57%, respectively. Together with its higher fixed-iteration quality, these results show that TaN sensitivity remains sufficiently informative to offset overhead even under tight coordinate budgets by coupling target-neighborhood structure with the evolving INR state.

A.3 Additional Implementation Details

A.3.1 Estimation of Signal-Local Jacobian

For an audio signal, let $f^\star(\mathbf{x}_t) : \mathbb{R} \to \mathbb{R}$ denote the target value at temporal sample t. At an interior sample, we estimate its signal-local Jacobian using a centered finite difference:

$$\widehat{J_{f^\star}} = \frac{f^\star(\mathbf{x}_{t+1}) - f^\star(\mathbf{x}_{t-1})}{2} \in \mathbb{R}^{1\times 1}. \tag{11}$$

Method	500th Iteration			1000th Iteration			3000th Iteration			5000th Iteration			10000th Iteration		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Full	27.424	0.7316	0.4370	30.450	0.8249	0.2743	36.353	0.9369	0.0777	38.663	0.9604	0.0354	42.568	0.9789	0.0111
Uniform	27.043	0.7201	0.4510	29.512	0.8131	0.3087	34.559	0.9137	0.1095	36.697	0.9450	0.0695	41.184	0.9702	0.0268
EGRA	27.050	0.7205	0.4502	29.540	0.8137	0.3073	34.661	0.9115	0.1075	36.185	0.9378	0.0610	41.119	0.9699	0.0260
Soft	27.852	0.7407	0.4504	31.120	0.8411	0.2716	35.078	0.9126	0.0915	38.009	0.9460	0.0570	41.622	0.9705	0.0249
INT	27.849	0.7393	0.4557	31.149	0.8428	0.2706	35.272	0.9254	0.0942	38.104	0.9442	0.0491	41.590	0.9703	0.0253
EVOS	28.038	0.7458	0.4172	31.336	0.8494	0.2287	35.361	0.9275	0.0908	38.055	0.9428	0.0489	41.150	0.9711	0.0268
E.S.	27.899	0.7345	0.4270	31.596	0.8426	0.2336	35.480	0.9237	0.0927	37.082	0.9399	0.0683	40.197	0.9648	0.0347
NINT	28.073	0.7441	0.4211	31.374	0.8461	0.2360	35.660	0.9235	**0.0867**	38.102	0.9467	0.0493	41.662	0.9710	0.0256
STaNS	**28.501**	**0.7580**	**0.3935**	**32.054**	**0.8544**	**0.2199**	**35.791**	**0.9295**	0.0879	**38.215**	**0.9563**	**0.0473**	**42.043**	**0.9717**	**0.0237**

Table 8: Extended iteration-conditioned results of image fitting task, reporting reconstruction quality at iteration checkpoints. Results are averaged over 24 images from the Kodak dataset.

We use a forward difference at the first sample and a backward difference at the last sample. These differences are computed in sample-index units.

For an RGB image, let $f^\star_{i,j} : \mathbb{R}^2 \to \mathbb{R}^3$ denote the target value at pixel (i, j). At an interior pixel, we estimate its signal-local Jacobian using centered finite differences:

$$\widehat{J_{f^\star}} = \left[\frac{f^\star_{i,j+1} - f^\star_{i,j-1}}{2}, \frac{f^\star_{i+1,j} - f^\star_{i-1,j}}{2} \right] \in \mathbb{R}^{3\times 2}. \tag{12}$$

At image boundaries, we use the one-sided difference. These differences are computed in pixel-index units.

For a shape represented by an SDF, i.e., $f^\star : \mathbb{R}^3 \to \mathbb{R}$, surface points and normals are obtained from the oriented point cloud or PLY mesh data used to construct the SDF training set, and we estimate its signal-local Jacobian by averaging the unit normals of the $k = 3$ surface samples nearest to $\mathbf{x}_i$:

$$\widehat{J_{f^\star}} = \left(\frac{1}{k} \sum_{m=1}^{k} \mathbf{n}_{q_m} \right)^\top \in \mathbb{R}^{1\times 3}, \tag{13}$$

where $\mathbf{n}_{q_m}$ is the unit normal of the m-th neighbor.

A.3.2 Mean-Normalization of TaN Scores

Before applying EMA smoothing, we mean-normalize each TaN score as

$$\bar{s}_i^t = \frac{s_i^t}{\max\left(\frac{1}{D} \sum_{j=1}^{D} s_j^t, \epsilon_{\mathrm{n}} \right)}, \tag{14}$$

where $\epsilon_{\mathrm{n}} = 10^{-12}$ prevents division by zero.

A.4 Additional Results

Table 8 extends the iteration-conditioned comparison in Table 1 to later training stages. Among sampling methods, STaNS achieves the best PSNR and SSIM at all five checkpoints and the lowest LPIPS at four of them. Although E.S. performs strongly at early checkpoints, its Sobel-derived edge anchors do not condition sampling on the evolving INR state, causing its performance to degrade noticeably at later stages and reach only 40.197 dB PSNR at the 10,000th iteration. In contrast, STaNS reaches 42.043 dB PSNR, 0.9717 SSIM, and 0.0237 LPIPS, improving over the strongest sampling baseline by 0.381 dB in PSNR and reducing LPIPS by 4.82%. These results demonstrate that dynamically coupling target-neighborhood structure with the evolving reconstruction state allows STaNS to maintain its advantage throughout long-horizon training.

Fig. 6 visualizes STaNS sampling dynamics on *kodim16*, *kodim20*, and *kodim23* from Kodak dataset. On *kodim16*, STaNS first concentrates on cloud boundaries and the tree-lined island before the 201st iteration, shifts toward the textured water surface around the 1409th iteration once the dominant skyline is well reconstructed, and later refines cloud contours and water ripples near the 5561st iteration. On *kodim20*, sampling initially traces the wing boundaries, engine cowling, and propeller outlines before the 151st iteration, then expands toward the flaps, landing gear, and grassland

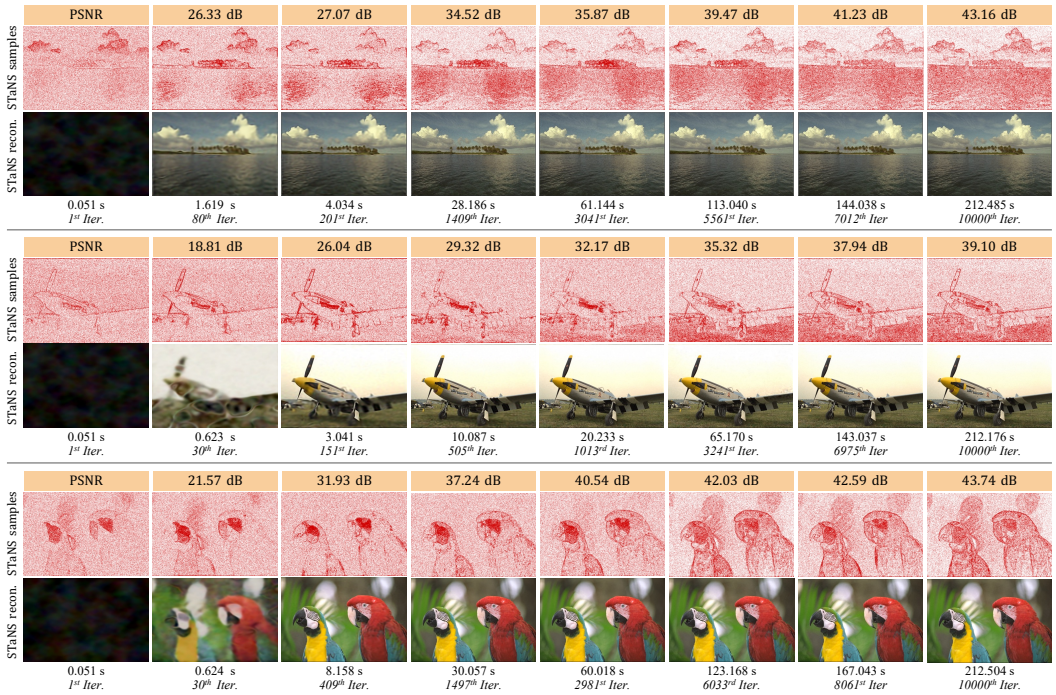


Figure 6: Visualizations of STaNS sampling and reconstruction evolution on *kodim16*, *kodim20*, and *kodim23*, shown from top to bottom. For each image, the upper and lower rows show sampled coordinates highlighted in red and the corresponding reconstructions, illustrating how TaN sensitivity progressively redirects sampling toward unresolved target structures.

textures around the 1013th iteration, while continuing to attend to fuselage markings throughout training. On *kodim23*, STaNS first emphasizes the parrots' heads and beaks before the 409th iteration, follows their feather contours around the 1497th iteration, and later focuses on fine plumage and background foliage from the 6033rd iteration onward. Eventually, the reconstructions on three images progressively improve to 43.16, 39.10, and 43.74 dB PSNR at the 10,000-th iteration. These transitions further show how TaN sensitivity dynamically redirects selection toward unresolved target structures as training progresses, enabling informative sampling for accelerated INR training.