NTK-Guided Implicit Neural Teaching

Chen Zhang* † Wei Zuo* Bingyang Cheng Yikun Wang Wei-Bin Kou
Yik-Chung Wu Ngai Wong
The University of Hong Kong
{czhang6, weizuo002}@connect.hku.hk {ycwu, nwong}@eee.hku.hk

Project page

Abstract

Implicit Neural Representations (INRs) parameterize continuous signals via multilayer perceptrons (MLPs), enabling compact, resolution-independent modeling for tasks like image, audio, and 3D reconstruction. However, fitting high-resolution signals demands optimizing over millions of coordinates, incurring prohibitive computational costs. To address it, we propose NTK-Guided Implicit Neural Teaching (NINT), which accelerates training by dynamically selecting coordinates that maximize global functional updates. Leveraging the Neural Tangent Kernel (NTK), NINT scores examples by the norm of their NTK-augmented loss gradients, capturing both fitting errors and heterogeneous leverage (self-influence and cross-coordinate coupling). This dual consideration enables faster convergence compared to existing methods. Through extensive experiments, we demonstrate that NINT significantly reduces training time by nearly half while maintaining or improving representation quality, establishing state-of-the-art acceleration among recent sampling-based strategies.

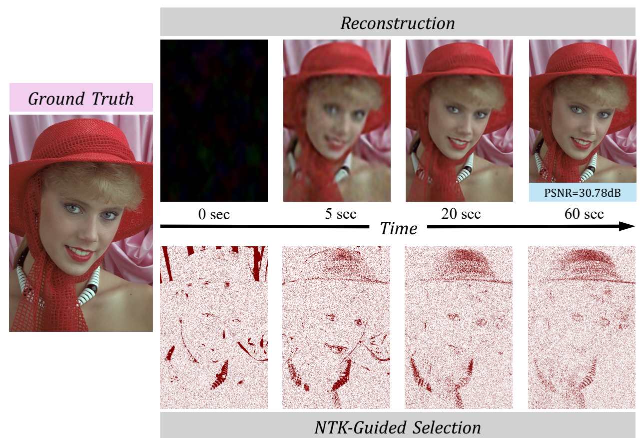

Figure 1. NINT selection and reconstruction on *kodim04* from Kodak [11]. **Top**: INR reconstructions at 0, 5, 20, 60 sec (final PSNR 30.78 dB). **Bottom**: NINT-selected coordinates (red), scored by NTK-augmented loss gradient norm to capture fitting errors and heterogeneous leverage (self-influence and cross-coupling).

1. Introduction

Implicit Neural Representations (INRs) [59, 65] model discrete signals (*e.g.*, audios, images, or 3D scenes) using continuous multilayer perceptrons (MLPs). These MLPs take low-dimensional coordinates as input (*e.g.*, pixel locations or spatiotemporal points) and output the corresponding signal values, effectively learning a continuous function that interpolates the discrete data with high fidelity. This coordinate-based formulation has revolutionized several domains, including high-resolution image representation [50, 59], novel view synthesis [36, 39], and compact signal compression [10, 45, 56, 62].

Nevertheless, training INRs is computationally intensive due to the large number of training examples inherent in high-resolution or high-dimensional signals. For instance, a single 1024×1024 image contains over one million pixels, each treated as an independent training point. Videos and 3D volumes scale this further into billions of coordinates. As a result, standard gradient descent requires repeated full passes over the entire dataset, making training slow and resource-heavy.

Several acceleration strategies have been proposed, each with inherent limitations. Partition-based methods [28, 35, 54] divide the signal domain across multiple specialized MLPs, but this increases architectural complexity and inference overhead. Hybrid explicit-implicit approaches [7, 41, 69] augment MLPs with structured representations such as tensors or voxels, achieving faster convergence at the expense of higher memory consumption. Meta-learning techniques [9, 66] accelerate fitting via pre-trained initialization, yet demand large homogeneous datasets for pre-training,

*Equal contribution
†Corresponding author

limiting flexibility and requiring significant upfront computation.

In contrast, sampling-based methods [12, 75, 79, 80] offer a lightweight alternative by selectively training on a subset of coordinates each iteration. These approaches reduce per-step computation without modifying the model or requiring auxiliary data. However, most rely on static heuristics, such as the magnitude of current output error or local signal variation, to guide selection. While intuitive, these criteria overlook the evolving dynamics of the MLP during training, particularly how different coordinates influence parameter updates and global convergence. This limitation restricts their ability to achieve maximal acceleration.

To address these limitations, we propose NTK-Guided Implicit Neural Teaching (NINT), a novel sampling framework that incorporates the Neural Tangent Kernel (NTK) to capture the MLP's evolving training dynamics. Rather than relying solely on static measures of output error, NINT jointly evaluates two complementary factors: (1) the gradient of the loss with respect to the network output, which identifies regions of high fitting discrepancy, and (2) the NTK-induced influence of each coordinate on parameter updates, which quantifies how strongly a point drives global model changes during training. By prioritizing examples that combine high fitting error with strong dynamic influence, NINT ensures that each training batch maximally accelerates global convergence. Fig. 1 visualizes the evolving reconstructions and NINT-selected coordinates. Through extensive experiments, NINT reduces training time by nearly half compared to full-dataset training, while preserving or improving final representation quality. Among recent sampling-based strategies [12, 19, 75, 79, 80], NINT establishes new state-of-the-art performance in both speed and fidelity. Our key contributions are:

- **NTK-centric analysis of INR dynamics**: We derive the functional evolution under gradient descent, exposing the flaws in error-only sampling due to neglected self-leverage and cross-coordinate coupling (Sec. 3.2).
- **NINT sampling strategy**: A principled, plug-and-play method that selects examples by maximizing the magnitude of the functional update induced by each coordinate, computed as the norm of NTK row (capturing global influence) multiplied by the loss gradients, ensuring every parameter step drives the largest possible improvement across the entire signal (Sec. 3.3).
- **State-of-the-art acceleration**: Extensive experiments demonstrate that NINT reduces training time by nealy half compared to full-batch baselines, while matching or exceeding reconstruction quality, outperforming recent sampling-based strategies (Sec. 4).

2. Related Works

2.1. Implicit Neural Representations

Implicit Neural Representations (INRs) [3, 14, 44, 59] encode discrete signals as continuous functions using coordinate-based MLPs, delivering memory-efficient, high-resolution representations across diverse domains, spanning images [10, 69], videos [8, 20, 34], 3D shapes [6, 25], scenes [7, 38, 41], and unconventional data types [4, 15]. Beyond reconstruction, INRs drive breakthroughs in compression [62], generative modeling [43], novel view synthesis [38], inverse imaging [16], copyright protection [31–33, 60], and physics-informed PDE solving [47]. Progress in architectural design, including sinusoidal activations [48, 55], Fourier positional encodings [30, 65], and structured priors [13, 23, 24, 27, 40, 67, 72], has significantly boosted expressiveness and alleviated spectral bias [46]. Yet, training on complex signals remains computationally demanding due to massive coordinate counts, highlighting the critical need for efficient, model-agnostic acceleration methods.

2.2. Acceleration for INR Training

To mitigate the high computational cost of training over vast coordinate sets, prior works have explored diverse acceleration paradigms while trading off model complexity, memory, or auxiliary resources. Partition-based techniques [28] segment the domain into subregions managed by specialized MLPs, leveraging regular grids [51], adaptive partitioning [35], Voronoi diagrams [49], or pyramidal structures [54], yet they introduce inference latency from ensemble coordination. Explicit caching methods integrate structured priors, such as hash grids [41], low-rank tensors [7], tree hierarchies [29, 71], or point clouds [70], to expedite convergence at the cost of increased memory. Meta-learning approaches [9, 66] provide task-specific initializations through pre-training on vast corpora [66] or hypernetworks [59, 63], but demand substantial upfront data and computation. Additional efficiency gains have been pursued via modulators [37], reparameterizations [58], input transformations [57], and normalization layers [5].

In contrast, sampling-based methods [12, 75, 79, 80] preserve architectural simplicity by subsampling coordinates per iteration, including edge-aware EGRA [12], frequency-prior-driven Expansive Supervision [80], Monte-Carlo Soft Mining [19], and evolutionary EVOS [79]. Most closely related, INT [75] recasts INR acceleration as a nonparametric teaching problem (*i.e.*, strategically selecting the most informative coordinates to guide MLP convergence, akin to a teacher curating high-impact examples) [73, 74, 76, 77], using greedy functional algorithms for coordinate selection. While effective, these heuristics typically ignore the MLP's parameter update dynamics, constraining convergence speed. As discussed in §3.2, INT is essentially using a matrix similar to an identity matrix to approximate the NTK, with nearly identical values along the diagonal. This inaccurately models the true direction of parameter updates unless the NTK is nearly diagonal, which is rarely the case in typical MLP training. [17, 22]. Consequently, the selected coordinates may not optimally drive convergence. Differently, our NINT integrates the full NTK to explicitly model training dynamics in the selection process, jointly optimizing for high output-gradient error and strong parameter-update influence, thereby maximizing per-batch progress toward global convergence.

3. Method

We begin by formulating the implicit neural representation (INR) as an MLP that maps coordinates to signal attributes, trained via empirical risk minimization. We then analyze INR training dynamics through the lens of the neural tangent kernel (NTK), revealing heterogeneous self-leverage and functional coupling that prior error-only sampling strategies fail to account for. To address this, we propose NTK-Guided Implicit Neural Teaching (NINT), which selects influential coordinates by maximizing NTK-augmented gradients for faster global convergence. We detail the formulation in Sec. 3.1, dynamics in Sec. 3.2, and NINT in Sec. 3.3.

3.1. Formulation

Let $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ denote a natural signal consisting of N observation pairs, where $\mathbf{x} \in \mathbb{R}^m$ represents an m-dimensional coordinate (*e.g.*, spatial, temporal, or spatiotemporal location) and $\mathbf{y}_i \in \mathbb{R}^n$ denotes the corresponding n-dimensional attribute vector (*e.g.*, audio intensity, pixel values, density, or feature embedding).

The goal of INR is to learn a continuous mapping $f_\theta : \mathbb{R}^m \to \mathbb{R}^n$ parameterized by an MLP with parameters θ, such that $f_\theta(\mathbf{x}_i) \approx \mathbf{y}_i$ for all $i \in \{1, \dots, N\}$, and ideally, f_θ generalizes to unobserved coordinates $\mathbf{x} \notin \{\mathbf{x}_i\}_{i=1}^N$. This is achieved by minimizing an empirical risk over the observed data:

$$\theta^\star = \arg\min_\theta \frac{1}{N} \sum_{i=1}^N \mathcal{L}\left(f_\theta(\mathbf{x}_i), \mathbf{y}_i\right), \tag{1}$$

where $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_+$ is a task-specific loss function (typically the ℓ_2 norm for regression). A regularizer $\mathcal{R}(\theta)$ with coefficient $\lambda \geq 0$ may be included to enhance generalization [26].

The MLP f_θ typically consists of L fully connected layers with non-linear activations (*e.g.*, sine [59]):

$$\begin{aligned} &\mathbf{h}_0 = \mathbf{x} \\ &\mathbf{h}_\ell = \sigma_\ell(W_\ell \mathbf{h}_{\ell-1} + \mathbf{b}_\ell), \quad \ell = 1, \dots, L-1, \\ &f_\theta(\mathbf{x}) = W_L \mathbf{h}_{L-1} + \mathbf{b}_L \end{aligned} \tag{2}$$

where $\{W_\ell, \mathbf{b}_\ell\} \equiv \theta$ are learnable weights and biases, and σ_ℓ denotes the activation function at layer ℓ. To enhance high-frequency reconstruction, positional encoding may be applied to the input [38, 58, 65].

During inference, f_θ enables continuous evaluation at arbitrary resolutions without storing the full signal, making INRs particularly suitable for high-dimensional, large-scale, or irregularly sampled data. The learned representation is *implicit*, as the signal is encoded entirely within the network parameters θ, rather than in discrete voxels or vertices.

3.2. INR Training Dynamics

Training an INR as defined in Eq. 1 is typically performed via gradient descent on the network parameters [53]. For a dataset of size N (*i.e.*, N observation pairs), the full-batch parameter update at iteration t is:

$$\theta^{t+1} \leftarrow \theta^t - \frac{\eta}{N} \sum_{i=1}^N \nabla_\theta \mathcal{L}\big(f_{\theta^t}(\mathbf{x}_i), \mathbf{y}_i\big), \tag{3}$$

where $\eta > 0$ is the learning rate.

When η is sufficiently small, the discrete update can be approximated in continuous time as a differential equation:

$$\frac{\partial \theta^t}{\partial t} = -\frac{\eta}{N} \big[\nabla_\theta f_{\theta^t}(\mathbf{x}_i)\big]_{i=1}^N{}^\top \cdot \big[\nabla_f \mathcal{L}\big(f_{\theta^t}(\mathbf{x}_i), \mathbf{y}_i\big)\big]_{i=1}^N. \tag{4}$$

To understand how the *function* f_{θ^t} evolves, consider its first-order Taylor expansion:

$$f_{\theta^{t+1}}(\mathbf{x}) - f_{\theta^t}(\mathbf{x}) = \Big\langle \frac{\partial f_{\theta^t}(\mathbf{x})}{\partial \theta^t}, \theta^{t+1} - \theta^t \Big\rangle + o(\|\theta^{t+1} - \theta^t\|). \tag{5}$$

Taking the continuous-time limit yields:

$$\frac{\partial f_{\theta^t}(\mathbf{x})}{\partial t} \simeq \Big\langle \frac{\partial f_{\theta^t}(\mathbf{x})}{\partial \theta^t}, \frac{\partial \theta^t}{\partial t} \Big\rangle. \tag{6}$$

Substituting the parameter evolution (*i.e.*, Eq. 4) gives:

$$\frac{\partial f_{\theta^t}(\mathbf{x})}{\partial t} \simeq -\frac{\eta}{N} \big[K_{\theta^t}(\mathbf{x}_i, \mathbf{x})\big]_{i=1}^N{}^\top \cdot \big[\nabla_f \mathcal{L}\big(f_{\theta^t}(\mathbf{x}_i), \mathbf{y}_i\big)\big]_{i=1}^N, \tag{7}$$

where

$$K_{\theta^t}(\mathbf{x}_i, \mathbf{x}) \equiv \Big\langle \frac{\partial f_{\theta^t}(\mathbf{x}_i)}{\partial \theta^t}, \frac{\partial f_{\theta^t}(\mathbf{x})}{\partial \theta^t} \Big\rangle \tag{8}$$

is the Neural Tangent Kernel (NTK) at parameters θ^t, input $\mathbf{x}_i$ and $\mathbf{x}$ [17].

In the infinite-width limit, K_{θ^t} remains *constant* throughout training, transforming gradient descent into kernel regression in function space [17]. For finite-width MLPs, the NTK evolves gradually but remains a reliable descriptor of training dynamics [2, 22].

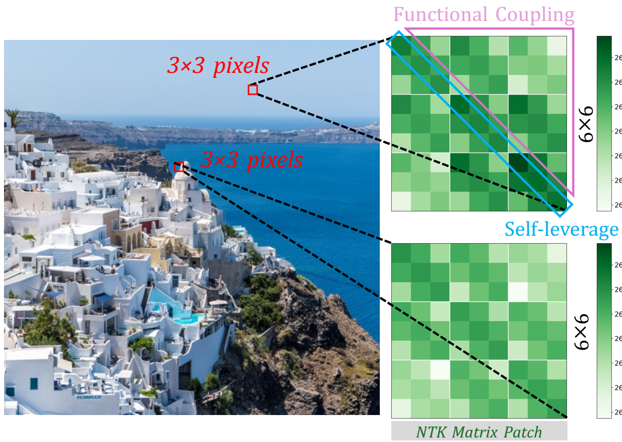

Figure 2. NTK visualization on the image 02 from DIV2K [1]. Two 9×9 NTK patches correspond to two 3×3 pixel regions (red). Strong off-diagonals show significant functional coupling between regions, while diverse diagonals indicate heterogeneous self-leverage. Color denotes magnitude.

The full NTK matrix $K_{\theta^t}(\mathbf{x}_i, \mathbf{x}_j)$ captures **functional coupling** between coordinates via inner products in parameter-gradient space: the off-diagonal entry $K_{\theta^t}(\mathbf{x}_i, \mathbf{x}_j)$ measures the projection of $\frac{\partial f_{\theta^t}(\mathbf{x}_j)}{\partial \theta^t}$ onto the update direction induced by $\mathbf{x}_i$. A large $|K_{\theta^t}(\mathbf{x}_i, \mathbf{x}_j)|$ indicates that a parameter update driven by the loss at $\mathbf{x}_i$ will significantly **co-move** the output at $\mathbf{x}_j$, even if $\mathbf{x}_j$ is not included in the current batch.

Meanwhile, the diagonal trace $K_{\theta^t}(\mathbf{x}_i, \mathbf{x}_i) = \left\|\frac{\partial f_{\theta^t}(\mathbf{x}_i)}{\partial \theta^t}\right\|_2^2$ quantifies the **self-leverage** of $\mathbf{x}_i$, *i.e.*, the squared norm of its output gradient in parameter space. High $K_{\theta^t}(\mathbf{x}_i, \mathbf{x}_i)$ implies that a unit parameter update along the loss gradient at $\mathbf{x}_i$ induces a large change in its own output f_θ and, through off-diagonal coupling, across the global function.

We illustrate this NTK behavior in Fig. 2, where high self-leverage (diagonal) and strong functional coupling (off-diagonal) are evident in local 9×9 NTK matrix patches centered on selected 3×3 pixel regions.

This dual structure, *i.e.*, **varying self-leverage** and **non-negligible functional coupling**, exposes a critical flaw in prior sampling methods [12, 75, 79]. By selecting coordinates solely based on output error $\|\nabla_f \mathcal{L}\|_2$, these approaches implicitly assume the NTK is **diagonal and isotropic**: *i.e.*, no cross-coordinate influence ($K_{\theta^t}(\mathbf{x}_i, \mathbf{x}_j) \approx 0$ for $i \neq j$) and uniform self-leverage ($K_{\theta^t}(\mathbf{x}_i, \mathbf{x}_i) \approx c$ for all i). This is equivalent to approximating the NTK with a scaled identity matrix.

In practice, neither holds: MLPs exhibit strong off-diagonal coupling due to weight sharing [17], and diagonal values vary by orders of magnitude—coordinates in high-frequency or high-curvature regions (*e.g.*, edges, transients) have significantly larger NTK traces than those in smooth or noisy areas [22]. Consequently, error-only sampling frequently prioritizes high-error but low-leverage points, wasting gradient steps on updates with minimal global impact. It prompts us to dynamically select examples for fast convergence as described in the next subsection.

Algorithm 1 NTK-Guided Implicit Neural Teaching

Input: Signal $S = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, MLP f_θ, batch size B, learning rate η, iterations T
Output: Trained INR f_{θ_T}

Initialize θ_0
for $t = 0$ to $T-1$ **do**
  Forward pass: $\hat{\mathbf{y}}_i = f_{\theta_t}(\mathbf{x}_i)$ for all i
  Compute $\mathbf{g}^t = \left[\nabla_f \mathcal{L}\big(f_{\theta^t}(\mathbf{x}_i), \mathbf{y}_i\big)\right]_{i=1}^N$
  Compute NTK row: $K_{\theta^t}(\mathbf{x}_i, :)$ for all i
  Select: $\mathcal{B}_t = \underset{\substack{\mathcal{B} \subseteq \{1,\ldots,N\} \\ |\mathcal{B}|=B}}{\arg\max} \left\|[K_{\theta^t}(\mathbf{x}_i, :) \cdot \mathbf{g}^t]_{i \in \mathcal{B}}\right\|_2$
  Update: $\theta_{t+1} \leftarrow \theta_t - \frac{\eta}{B} \sum_{i \in \mathcal{B}_t} \nabla_\theta \mathcal{L}(f_{\theta_t}(\mathbf{x}_i), \mathbf{y}_i)$
end for
return f_{θ_T}

3.3. NTK-Guided Implicit Neural Teaching

To overcome the isotropic-diagonal assumption of error-only sampling, we propose NTK-Guided Implicit Neural Teaching (NINT), a principled sampling strategy that explicitly accounts for the **heterogeneous self-leverage** and **functional coupling** encoded in the NTK. Rather than treating all high-error points equally, NINT prioritizes coordinates that are **both poorly fitted** and **globally influential**, *i.e.*, those that drive large functional changes across the domain via strong self-leverage and cross-coordinate coupling.

Error-only sampling [12, 75, 79] selects a batch $\mathcal{B} \subseteq \{1, \ldots, N\}$ of size B by maximizing the norm of the loss gradient vector, *i.e.*

$$\mathcal{B}^\star = \underset{|\mathcal{B}|=B}{\arg\max} \left\| \left[\nabla_f \mathcal{L}\big(f_{\theta^t}(\mathbf{x}_i), \mathbf{y}_i\big)\right]_{i \in \mathcal{B}} \right\|_2 .$$

This prioritizes points with large local prediction error (*i.e.*, high output disparity between the current MLP prediction and the target signal). While some prior methods, such as EVOS [79], augment this criterion by incorporating a Laplacian operator applied to both the predicted $f_{\theta^t}(\mathbf{x})$ and the given signal $\mathbf{y}$ to better capture high-frequency residuals, they still operate *solely in the output space* and remain agnostic to the parameter-to-function mapping induced by the NTK. However, as highlighted in Sec. 3.2, this ignores the varying self-leverage and functional coupling encoded in the NTK, leading to inefficient updates that may prioritize high-error but low-impact points.

NINT, in contrast, selects samples to maximize the expected magnitude of the functional evolution, as captured by the NTK-augmented gradient. From Eq. 7, the instantaneous change in the function $f_{\theta^t}(\mathbf{x})$ at any point $\mathbf{x}$ is governed by the product of the NTK row $K_{\theta^t}(\mathbf{x}_i, \mathbf{x})$ and the loss gradient at observed points. To accelerate global convergence, NINT prioritizes coordinates $\mathbf{x}_i$ that are *both poorly fitted* and *globally influential*, *i.e.*, those inducing large functional shifts via **strong self-leverage** ($K_{\theta^t}(\mathbf{x}_i, \mathbf{x}_i)$) and **cross-coordinate coupling** ($K_{\theta^t}(\mathbf{x}_i, \mathbf{x}_j)$). Formally:

$$\mathcal{B}^\star = \underset{|\mathcal{B}|=B}{\arg\max} \left\| \left[K_{\theta^t}(\mathbf{x}_i, :) \cdot \left[\nabla_f \mathcal{L}\big(f_{\theta^t}(\mathbf{x}_j), \mathbf{y}_j\big) \right]_{j=1}^{N} \right]_{i \in \mathcal{B}} \right\|_2,$$

where $K_{\theta^t}(\mathbf{x}_i, :) \in \mathbb{R}^{1\times N}$ denotes the row of the NTK matrix corresponding to $\mathbf{x}_i$. This incorporates both the local error magnitude and the coordinate's leverage on the broader function space. Pseudo code is in Algorithm 1.

4. Results

4.1. Experimental Setup

4.1.1. Settings and Metrics

Following [79, 80], a portion ξ of the training set is randomly selected. The remaining portion $1 - \xi$ is divided between NTK-guided sampling and error-based sampling. The NTK-guided ratio is $(1 - \xi)\exp(-\lambda t/\alpha)$, where λ controls the decay rate, and α sets the NTK recomputation interval (reusing prior results otherwise). The error-based ratio constitutes the balance, selecting samples via fitting errors as in prior works [12, 75]. We adopt this hybrid sampling to leverage NTK's global influence while transitioning to effective error-based selection, with hyperparameters tuned for computational balance (see ablations in Sec. 4.3.3). Particularly, default hyperparameters are set as follows: $\xi = 0.7$, $\alpha = 10$, and $\lambda = 1.0$. For baselines, we used their official default settings. The default network is a 5×256 SIREN [62], with further analysis of network size and architecture in Sec. 4.3.1 and Sec. 4.3.2. All runs were on a single NVIDIA RTX 4090 GPU with 24 GB of memory. We measured reconstruction quality using PSNR, SSIM [68], and LPIPS [78].

4.1.2. Baseline Strategies and Datasets

We benchmark NINT against several leading INR acceleration methods: **(1)** Expansive Supervision (Expan.) [80], **(2)** EVOS [79], **(3)** INT [75], **(4)** SoftM. [19], **(5)** EGRA [12], **(6)** Uniform Random Sampling (Unif.), and **(7)** conventional full-coordinate training (Stand.). For EVOS [79], we incorporate the official variant, EVOS (w/o CFS.), where only vanilla L2 loss is implemented without Cross-Frequency Supervision; For Expan. [80], to align with its official uniform random sampling ratio setting ($\xi = 0.5$), we include a matching NINT variant for fair comparison. All methods use a learning rate of $\eta = 1\text{e}-4$ and a batch size of $B = 20$ % of the full sample set (except $B = 100$ % for Stand.). Our 2D image experiments draw from the Kodak [11] and DIV2K [1] datasets. Furthermore, extended experiments on more images and other modalities, including 1D and 3D datasets, can be found in *supplementary materials*.

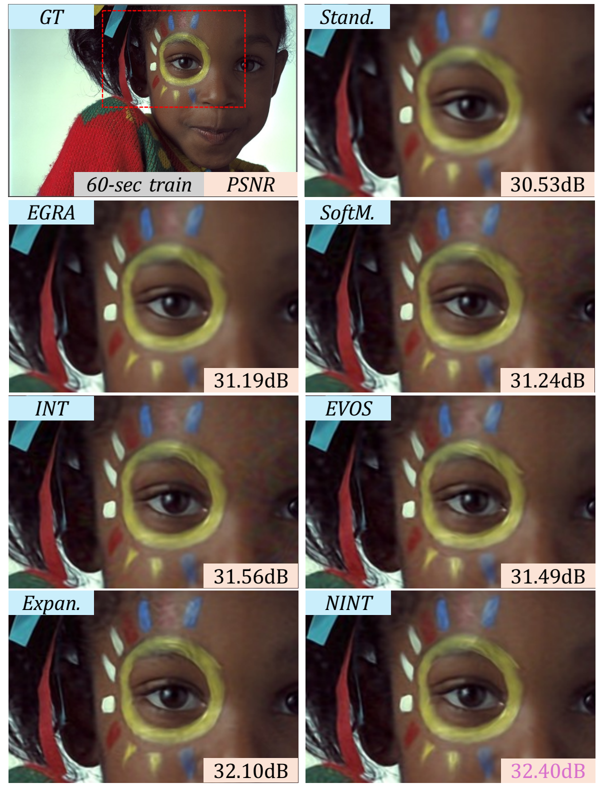

Figure 3. Visual comparison of image reconstructions using different sampling strategies on the *kodim15* image from the Kodak dataset [11], after a fixed training duration of 60 seconds.

4.2. Comparison with State-of-the-art Strategies

Table 1 presents the quantitative results under fixed training iteration budgets. NINT leads across all thresholds (250, 1000, and 5000 iterations) and metrics (PSNR, SSIM, LPIPS). This consistent superiority highlights the effectiveness of NTK-guided functional updates, which enable more informative sample selection and accelerate convergence. Notably, NINT not only achieves higher reconstruction quality at early stages, but also maintains its advantage as training progresses, indicating robust performance throughout the optimization process.

Table 2 examines efficiency by tracking iterations and time needed to hit specific PSNR targets. NINT demonstrates clear improvements in both training speed and resource utilization, outperforming all baselines across multiple target PSNR values (25, 30, and 35 dB). In particular, when compared to standard full-coordinate training (Stand.), NINT reduces the number of required iterations and total training time by up to 26.58 % and **48.99 %**, re-

Strategy	250 Iterations PSNR↑	SSIM↑	LPIPS↓	1000 Iterations PSNR↑	SSIM↑	LPIPS↓	5000 Iterations PSNR↑	SSIM↑	LPIPS↓
Stand.	27.897	0.774	0.438	31.669	0.845	0.278	39.763	0.962	0.022
Unif.	27.660	0.771	0.443	31.139	0.836	0.291	37.137	0.943	0.069
EGRA [12]	27.672	0.772	0.443	31.243	0.838	0.289	37.393	0.945	0.068
SoftM. [19]	27.647	0.747	0.498	31.381	0.816	0.338	35.504	0.920	0.070
INT [75]	27.568	0.747	0.499	31.192	0.815	0.351	39.020	0.943	0.035
EVOS [79] (w/o CFS.)	27.964	0.753	0.459	31.657	0.842	0.260	37.175	0.935	0.054
EVOS [79]	28.016	0.755	0.454	31.719	0.842	0.262	37.558	0.940	0.054
Expan. [80] ($\xi = 0.5$)†	28.034	0.773	0.420	32.273	0.857	0.227	38.350	0.948	0.049
Expan. [80] ($\xi = 0.7$)	27.990	0.772	0.426	32.154	0.855	0.240	38.220	0.947	0.056
NINT ($\xi = 0.5$)	28.720	0.770	0.438	32.591	0.851	0.246	38.082	0.933	0.037
NINT ($\xi = 0.7$)	28.956	0.776	0.414	32.640	0.841	0.238	39.085	0.958	0.029

† denotes the default setting in Expan. [80].

Table 1. Performance metrics (PSNR↑, SSIM↑, LPIPS↓) at fixed training iterations (250, 1000, 5000) across various sampling strategies on image fitting tasks. Purple : the best performance; Pale Purple : the secondary performance.

PSNR	Metric	Stand.	Uni.	EGRA	SoftM.	INT	EVOS	Expan. (ξ=0.5)†	Expan. (ξ=0.7)	NINT (ξ=0.5)	NINT (ξ=0.7)
25	iter↓	80	82	82	99	100	132	90	86	72	70
	Time(s)↓	8.40	5.94	6.09	7.03	6.44	8.90	6.57	6.34	6.07	5.92
30	iter↓	523	626	603	589	589	490	450	454	380	384
	Time(s)↓	49.11	38.85	37.85	37.87	33.01	31.20	28.59	29.16	25.63	25.05
35	iter↓	2043	2760	2619	2624	2023	2302	1802	1988	1706	1644
	Time(s)↓	184.78	168.46	160.41	165.42	111.80	143.20	111.01	123.60	108.90	102.88

† denotes the default setting in Expan. [80].

Table 2. Iterations and runtime (in seconds) required to reach target PSNR thresholds across various sampling strategies on image fitting tasks. Purple : the best performance; Pale Purple : the secondary performance.

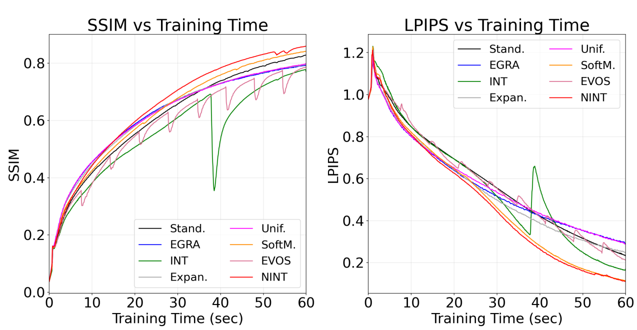

Figure 4. Trends in SSIM [68] and LPIPS [78] metrics over 60 seconds of training across various sampling strategies.

spectively. These savings make NINT ideal for INR tasks with tight time constraints, blending global and local sampling for faster, high-quality results.

Fig. 3 offers a visual comparison after 60 seconds of training. Compared to baseline strategies, NINT produces crisper details, especially in the zoomed eye area, clearly preserving patterns and color boundaries of the facial painting. This aligns with its top PSNR of 32.40 dB among all strategies, confirming that its quantitative advantage translates into meaningful visual improvements in INR tasks.

Fig. 4 tracks SSIM [68] and LPIPS [78] over 60 seconds. Across both metrics and over the full training duration, NINT consistently achieves the highest structural and perceptual fidelity and outclasses all baselines. Notably, NINT demonstrates superior convergence behavior, attaining higher SSIM and lower LPIPS values faster in training and maintaining this advantage through the final iteration. This indicates that NINT not only yields better image reconstruction quality but also does so more efficiently, underscoring its superior effectiveness.

4.3. Ablation Studies

4.3.1. Adaptability for Network Sizes

Since INR tasks differ in complexity, sampling strategies must adapt well. We tested NINT across SIREN [62] networks from 1×64 to 5×256 to check its fit for varying ca-

Figure 5. Visual comparison of reconstruction quality across various network sizes, after training for 150 seconds on image *07* from DIV2K [1].

Network Size	PSNR ↑			Time (s) ↓
	500	1000	2500	
1×64	11.835	17.137	18.305	N/A
1×64 (NINT)	13.890	17.606	18.527	N/A
3×64	22.291	23.527	25.110	159.04
3×64 (NINT)	22.023	23.611	25.257	141.14
3×128	23.172	24.169	26.140	92.16
3×128 (NINT)	23.195	24.506	26.521	72.14
5×128	24.486	26.036	30.114	45.34
5×128 (NINT)	24.948	27.363	30.259	31.47
3×256	24.075	25.785	28.852	57.68
3×256 (NINT)	24.404	27.761	29.444	39.75
5×256	25.613	28.685	33.693	35.42
5×256 (NINT)	26.851	31.274	35.100	22.16

Table 3. Comparison of PSNR at fixed training iterations (500, 1000, 2500) and elapsed time for 3000-iteration training across different network sizes, with and without NINT. Better performance is marked as purple.

pacities. Specifically, we compare: **1)** PSNR at fixed training iterations including 500, 1000 and 2500; **2)** the elapsed time for reaching the PSNR $=$ 25 dB threshold. As Table 3 indicates, NINT's time savings grow with network size, hitting 37.44 % at 5×256. Even for the small 1×64 network, which cannot always hit the threshold (marked as N/A) due to its limited capacity, NINT still brings up to 17.36 % PSNR improvement at fixed iteration checkpoints under such setting. Furthermore, NINT demonstrates consistent superiority of PSNR at given training iterations, which clearly confirms the effectiveness of incorporating NTK-guided sampling for training acceleration while satisfying distinct expressiveness requirements of INR tasks.

Fig. 5 presents a visual comparison of network sizes under a fixed 150-second training budget. While both strategies benefit from increased capacity, NINT consistently outperforms Stand., with PSNR improving from 25.58 dB (3×64) to 34.91 dB (5×256). Notably, the performance gap widens as the network scales, indicating that NINT more effectively leverages larger capacity for higher-fidelity reconstructions while maintaining its advantage even with limited model capacity. This highlights NINT's scalability and robustness across model sizes.

Network Structure	PSNR ↑			Time (s) ↓
	20s	60s	120s	
MLP	14.91	18.13	20.67	374.64
MLP + NINT	15.11	19.72	21.68	266.32
FFN [65]	16.75	26.9	31.44	54.19
FFN + NINT	18.88	27.39	31.48	48.75
FINER [30]	28.12	31.15	33.01	6.90
FINER + NINT	29.54	31.28	35.61	6.76
GAUSS [48]	21.51	25.87	33.05	56.23
GAUSS + NINT	21.46	27.11	33.38	48.24
PEMLP [65]	18.41	25.09	29.11	59.02
PEMLP + NINT	19.95	25.91	30.45	49.13
SIREN [62]	27.45	30.51	32.44	8.25
SIREN + NINT	29.28	32.40	35.47	5.81
WIRE [55]	16.88	23.86	27.17	83.30
WIRE + NINT	17.62	26.62	29.13	47.23

Table 4. Comparison of PSNR at fixed training times (20s, 60s, 120s) and elapsed time to reach PSNR=25 threshold across various INR network structures, with and without NINT. Better performance is highlighted in purple.

4.3.2. Adaptability for Network Structures

To probe NINT's adaptability for network structures, we conduct experiments across a diverse set of neural architectures, including plain MLP and its extensive variants:

Setting	Iterations ↓						Time (s) ↓					
	PSNR		SSIM		LPIPS		PSNR		SSIM		LPIPS	
	30	35	0.8	0.9	0.2	0.1	30	35	0.8	0.9	0.2	0.1
NINT (default)	389	1644	399	1639	1172	2082	25.09	102.88	26.00	102.56	73.51	130.21
$\alpha=1$	411	1682	424	1685	1258	2297	27.08	107.43	27.82	107.65	80.42	146.21
$\alpha=20$	401	1698	400	1717	1338	1941	26.19	106.24	26.15	107.44	83.74	121.31
$\alpha=50$	383	1650	394	1651	1155	2384	25.39	104.07	26.08	104.12	73.09	149.60
$\xi=0.4$	391	1659	431	1832	1327	2001	26.03	104.55	28.46	115.36	83.77	129.41
$\xi=0.5$	380	1706	414	1762	1200	1973	25.63	108.89	27.65	112.44	76.98	125.69
$\xi=0.6$	376	1656	401	1693	1287	2491	24.71	103.50	26.22	105.80	80.51	155.08
$\xi=0.8$	400	1647	399	1644	1181	2331	26.16	102.95	26.04	120.73	74.04	145.26
$\lambda=0.1$	390	1700	404	1711	1202	1962	28.48	110.37	29.37	110.99	79.42	126.50
$\lambda=0.5$	394	1731	402	1742	1239	1951	26.28	109.43	26.79	110.10	78.68	123.02
$\lambda=2.0$	380	1648	404	1643	1208	2028	24.82	102.85	26.22	102.55	75.57	126.24

Table 5. Ablation study on the impact of hyperparameters α, ξ, and λ in NINT, showing iterations and runtime (in seconds) required to reach target thresholds for PSNR, SSIM, and LPIPS on image fitting tasks. Purple : the best performance; Orange : the worst performance.

FFN [65], FINER [30], GAUSS [48], PEMLP [65], SIREN [62], and WIRE [55]. These architectures, characterized by heterogeneous designs in frequency-induced encoding and activation functions, encompass a broad spectrum of INR scenarios and pose unique challenges for sampling strategy robustness. As summarized in Table 4, we evaluate performance by comparing PSNR at fixed training times (20, 60, and 120 seconds), as well as the time required to reach a PSNR of 25 dB. The results demonstrate that NINT consistently delivers superior performance across all tested structures, reducing training time by up to 43.30 % and improving PSNR by as much as 11.57 % under comparable training budgets. These improvements are particularly notable in architectures with complex frequency encoding, where traditional sampling strategies often struggle to efficiently allocate computational resources. By consistently delivering acceleration and quality improvements regardless of model type, NINT is well-suited for practical INR applications with diverse network choices.

4.3.3. Adaptability for NINT Settings

We further assess the sensitivity of NINT to its hyperparameters across multiple metrics, considering both iteration- and time-based efficiency. As shown in Table 5, NINT achieves robust performance under the default configuration (see Sec. 4.1.1), consistently ranking at or near the top across all evaluation criteria (*e.g.*, fastest time to PSNR = 30 dB: 25.09 seconds; lowest time to SSIM = 0.8: 26.00 seconds). Importantly, NINT maintains strong performance even when deviating from default settings, demonstrating flexibility in optimization: for instance, $\alpha = 50$ yields the best time for LPIPS = 0.2, $\xi = 0.6$ achieves the lowest iteration count for PSNR = 30 dB, and $\lambda = 2.0$ attains the fastest time for PSNR = 35 dB and SSIM = 0.9. Even in worst-case scenarios (e.g., $\alpha = 1$ or $\lambda = 0.1$), NINT's results remain close to optimal, underscoring its stability across diverse configurations. Overall, these findings highlight that NINT delivers plug-and-play efficiency and robust performance without the need for intricate hyperparameter tuning.

5. Concluding Remarks and Future Work

In this work, we introduced NINT, a sampling-based strategy that accelerate INR training by leveraging NTK to dynamically select coordinates maximizing global functional updates, integrating fitting errors with insights into heterogeneous self-leverage and cross-coordinate coupling to overcome the inefficiencies of error-only methods that ignore the NTK's off-diagonal elements and lead to suboptimal progress. Our NTK-centric analysis revealed these limitations, while NINT prioritizes NTK-augmented gradient norms for faster convergence. Extensive benchmarks show NINT outperforming baselines in PSNR, SSIM, and LPIPS, reducing time and iterations to quality thresholds by up to 49 % and 27 % compared to full-batch training, with visuals confirming sharper details achieved faster. It scales effectively with network size, yielding greater savings on larger models, adapts seamlessly across diverse architectures with up to 43.3 % runtime cuts, and demonstrates robustness to hyperparameter variations, ensuring reliable performance without extensive tuning. Overall, NINT enhances INR practicality by speeding up training without architectural modifications or additional data, surpassing recent samplers [12, 19, 75, 79, 80]. Future work includes NTK approximations to lower overhead, adaptive batching, and integration with hybrid architectures for applications in neural signal processing.

Acknowledgments

This work was supported in part by the Theme-based Research Scheme (TRS) project T45-701/22-R of the Research Grants Council of Hong Kong, and in part by the AVNET-HKU Emerging Microelectronics and Ubiquitous Systems (EMUS) Lab.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 126–135, 2017. 4, 5, 7, 3

[2] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019. 3

[3] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *CVPR*, 2020. 2

[4] Hmrishav Bandyopadhyay, Ayan Kumar Bhunia, Pinaki Nath Chowdhury, Aneeshan Sain, Tao Xiang, Timothy Hospedales, and Yi-Zhe Song. Sketchinr: A first look into sketches as implicit neural representations. *arXiv preprint arXiv:2403.09344*, 2024. 2

[5] 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*, pages 25160–25171, 2024. 2

[6] Rohan Chabra, Jan Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *ECCV*, pages 608–625, 2020. 2

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

[8] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. In *NeurIPS*, pages 21557–21568, 2021. 2

[9] Yinbo Chen and Xiaolong Wang. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pages 170–187. Springer, 2022. 1, 2

[10] Emilien Dupont, Adam Goliński, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. Coin: Compression with implicit neural representations. In *ICLR Neural Compression Workshop*, 2021. 1, 2

[11] Eastman Kodak Company. Kodak lossless true color image suite. `http://r0k.us/graphics/kodak/`, 1999. [Accessed 14-08-2023]. 1, 5

[12] 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, 4, 5, 6, 8, 1

[13] Daniele Grattarola and Pierre Vandergheynst. Generalised implicit neural representations. In *NeurIPS*, 2022. 2

[14] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, 2020. 2

[15] Langwen Huang and Torsten Hoefler. Compressing multidimensional weather and climate data into neural networks. In *ICLR*, 2023. 2

[16] Xin Huang, Qi Zhang, Ying Feng, Hongdong Li, and Qing Wang. Inverting the imaging process by learning an implicit camera model. In *CVPR*, pages 21456–21465, 2023. 2

[17] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018. 3, 4

[18] Mark W Jones, J Andreas Baerentzen, and Milos Sramek. 3d distance fields: A survey of techniques and applications. *IEEE Transactions on visualization and Computer Graphics*, 12(4):581–599, 2006. 1

[19] 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*, pages 20071–20080, 2024. 2, 5, 6, 8, 1

[20] Subin Kim, Sihyun Yu, Jaeho Lee, and Jinwoo Shin. Scalable neural video representations with learnable positional features. In *NeurIPS*, 2022. 2

[21] 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)*, pages 626–630. IEEE, 2019. 1

[22] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *NeurIPS*, 2019. 3, 4

[23] Jason Chun Lok Li, Chang Liu, Binxiao Huang, and Ngai Wong. Learning spatially collaged fourier bases for implicit neural representation. In *AAAI*, 2024. 2

[24] Jason Chun Lok Li, Steven Tin Sui Luo, Le Xu, and Ngai Wong. Asmr: Activation-sharing multi-resolution coordinate networks for efficient inference. In *ICLR*, 2024. 2

[25] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *CVPR*, pages 8456–8465, 2023. 2

[26] Zhemin Li, Hongxia Wang, and Deyu Meng. Regularize implicit neural representation by itself. In *CVPR*, 2023. 3

[27] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *CVPR*, 2022. 2

[28] 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*, pages 5474–5483, 2023. 1, 2

[29] 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

[30] 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 variable-periodic activation functions. *arXiv preprint arXiv:2312.02434*, 2023. 2, 7, 8

[31] Ziyuan Luo, Qing Guo, Ka Chun Cheung, Simon See, and Renjie Wan. Copyrnerf: Protecting the copyright of neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22401–22411, 2023. 2

[32] Ziyuan Luo, Anderson Rocha, Boxin Shi, Qing Guo, Haoliang Li, and Renjie Wan. The nerf signature: Codebook-aided watermarking for neural radiance fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[33] Ziyuan Luo, Yangyi Zhao, Ka Chun Cheung, Simon See, and Renjie Wan. Imagesentinel: Protecting visual datasets from unauthorized retrieval-augmented image generation. In *NeurIPS*, 2025. 2

[34] Shishira Maiya, Sharath Girish, Max Ehrlich, Hanyu Wang, Kwot Sin Lee, Patrick Poirson, Pengxiang Wu, Chen Wang, and Abhinav Shrivastava. NIRVANA: Neural implicit representations of videos with adaptive networks and autoregressive patch-wise modeling. In *CVPR*, pages 14378–14387, 2023. 2

[35] Julien NP Martel, David B Lindell, Connor Z Lin, Eric R Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: adaptive coordinate networks for neural scene representation. *ACM Transactions on Graphics (TOG)*, 40(4):1–13, 2021. 1, 2

[36] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 1

[37] 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*, pages 14214–14223, 2021. 2

[38] Ben Mildenhall, Pratul Srinivasan, Matthew Tancik, Jonathan Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, pages 405–421, 2020. 2, 3

[39] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2021. 1

[40] Amirali Molaei, Amirhossein Aminimehr, Armin Tavakoli, Amirhossein Kazerouni, Bobby Azad, Reza Azad, and Dorit Merhof. Implicit neural representation in medical imaging: A comparative survey. In *ICCV*, 2023. 2

[41] 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, 2

[42] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*, 2015. 1

[43] Dogyun Park, Sihyeon Kim, Sojin Lee, and Hyunwoo Kim. DDMI: Domain-agnostic latent diffusion models for synthesizing high-quality implicit neural representations. *arXiv preprint arXiv:2401.12517*, 2024. 2

[44] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 2

[45] Francesca Pistilli, Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Signal compression via neural implicit representations. In *ICASSP*, 2022. 1

[46] 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 *ICML*, 2019. 2

[47] Maziar Raissi, Paris Perdikaris, and George Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019. 2

[48] Sameera Ramasinghe and Simon Lucey. Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In *ECCV*, pages 142–158, 2022. 2, 7, 8

[49] 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*, pages 14153–14161, 2021. 2

[50] Pradyumna Reddy, Zhifei Zhang, Zhaowen Wang, Matthew Fisher, Hailin Jin, and Niloy Mitra. A multi-implicit neural representation for fonts. In *NeurIPS*, 2021. 1

[51] Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 14335–14345, 2021. 2

[52] 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)*, pages 749–752. IEEE, 2001. 1

[53] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 3

[54] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representation. In *European Conference on Computer Vision*, pages 318–333. Springer, 2022. 1, 2

[55] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard Baraniuk.

WIRE: Wavelet implicit neural representations. In *CVPR*, pages 18507–18516, 2023. 2, 7, 8

[56] Jonathan Richard Schwarz, Jihoon Tack, Yee Whye Teh, Jaeho Lee, and Jinwoo Shin. Modality-agnostic variational compression of implicit neural representations. In *ICML*, 2023. 1

[57] 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*, pages 4830–4839, 2024. 2

[58] Kexuan Shi, Xingyu Zhou, and Shuhang Gu. Improved implicity neural representation with fourier bases reparameterized training. In *CVPR*, 2024. 2, 3

[59] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 1, 2, 3

[60] Qi Song, Ziyuan Luo, Ka Chun Cheung, Simon See, and Renjie Wan. Protecting nerfs' copyright via plug-and-play watermarking base model. In *European Conference on Computer Vision*, pages 57–73. Springer, 2024. 2

[61] Stanford Computer Graphics Laboratory. The stanford 3d scanning repository. `https://graphics.stanford.edu/data/3Dscanrep/`, 2007. 2, 3

[62] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *ECCV*, 2022. 1, 2, 5, 6, 7, 8

[63] 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*, pages 661–676. Springer, 2023. 2

[64] 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. 1

[65] 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. In *NeurIPS*, 2020. 1, 2, 3, 7, 8

[66] 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 *CVPR*, 2021. 1, 2

[67] Peihao Wang, Zhiwen Fan, Tianlong Chen, and Zhangyang Wang. Neural implicit dictionary learning via mixture-of-expert training. In *ICML*, 2022. 2

[68] 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. 5, 6

[69] Shaowen Xie, Hao Zhu, Zhen Liu, Qi Zhang, You Zhou, Xun Cao, and Zhan Ma. Diner: Disorder-invariant implicit neural representation. In *CVPR*, 2023. 1, 2

[70] 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*, pages 5438–5448, 2022. 2

[71] Runzhao Yang. Tinc: Tree-structured implicit neural compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18517–18526, 2023. 2

[72] Gizem Yüce, Guillermo Ortiz-Jiménez, Beril Besbinar, and Pascal Frossard. A structured dictionary perspective on implicit neural representations. In *CVPR*, 2022. 2

[73] Chen Zhang, Xiaofeng Cao, Weiyang Liu, Ivor Tsang, and James Kwok. Nonparametric teaching for multiple learners. In *NeurIPS*, 2023. 2

[74] Chen Zhang, Xiaofeng Cao, Weiyang Liu, Ivor Tsang, and James Kwok. Nonparametric iterative machine teaching. In *ICML*, 2023. 2

[75] Chen Zhang, Steven Tin Sui Luo, Jason Chun Lok Li, Yik Chung Wu, and Ngai Wong. Nonparametric teaching of implicit neural representations. In *International Conference on Machine Learning*, pages 59435–59458. PMLR, 2024. 2, 4, 5, 6, 8, 1

[76] Chen Zhang, Weixin Bu, Zeyi Ren, Zhengwu Liu, Yik-Chung Wu, and Ngai Wong. Nonparametric teaching for graph property learners. In *ICML*, 2025. 2

[77] Chen Zhang, Jianghui Wang, Bingyang Cheng, Zhongtao Chen, Wendong Xu, Cong Wang, Marco Canini, Francesco Orabona, Yik-Chung Wu, and Ngai Wong. Nonparametric teaching for attention learners. In *ICLR*, 2026. 2

[78] 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*, pages 586–595, 2018. 5, 6

[79] 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*, pages 30472–30482, 2025. 2, 4, 5, 6, 8, 1, 3

[80] 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)*, pages 1–6, 2025. 2, 5, 6, 8, 1, 3