

Closing the Capacity–Convergence Gap: Globally Optimal Configuration of Implicit Neural Representations

Sipeng Chen¹, Yan Zhang¹, and Shibo Li¹

Florida State University, Tallahassee, FL 32306, USA
{sc25bg,yzhang18,shibo.li}@fsu.edu

Abstract. Implicit Neural Representations (INRs) achieve state-of-the-art results across image representation, 3D modeling, and novel-view synthesis, yet their performance hinges on tightly coupled design decisions—activation family, initialization scheme, and per-layer hyperparameters—whose interdependencies render manual tuning unreliable and greedy search provably suboptimal. We introduce OptiINR, the first framework that recasts INR configuration as a global optimization problem over a mixed-variable space of discrete activation choices (e.g., SIREN, WIRE, FINER) and their continuous parameters. OptiINR consists of a structured layer-wise INR configuration space and an adaptive Bayesian optimization procedure for navigating it. The search space spans activation families, activation-specific continuous parameters, initialization/scaling choices, per-layer learning rates, and positional-encoding usage. Across five tasks—1D audio, 2D image, 3D occupancy, spatiotemporal PDE, and neural radiance field synthesis—OptiINR consistently outperforms hand-tuned baselines and prior automated methods, revealing task-dependent activation patterns that challenge prevailing one-size-fits-all design conventions. Code is available at <https://github.com/Sippengg/OptiINR>.

Keywords: Implicit Neural Representations · Bayesian Experimental Design

1 Introduction

Implicit Neural Representations (INRs), also referred to as coordinate-MLPs, have fundamentally reshaped how continuous signals are represented and processed across domains from computer vision to computational physics [17, 43]. In contrast to traditional discrete representations tied to fixed spatial resolutions, INRs parameterize signals as continuous functions via neural networks, yielding resolution-independent representations with exceptional expressiveness. This paradigm has unlocked capabilities that were previously unattainable, powering applications such as Neural Radiance Fields (NeRF) for photorealistic view synthesis [22], signed distance functions for high-fidelity 3D reconstruction [21, 23], advanced medical imaging, and even neural solvers for partial differential equations [26, 36]. The strength of INRs lies in their ability to exploit the

universal approximation property of neural networks [4, 11] to learn complex, high-dimensional mappings from coordinate space to signal values. Landmark works such as DeepSDF [23] demonstrated that MLPs can learn continuous signed distance functions for representing 3D geometry, while NeRF showed that similar architectures can capture view-dependent radiance fields with high fidelity [22]. Together, these advances established coordinate-MLPs as flexible neural parameterizations for continuous signal representation.

Despite substantial progress, the practical effectiveness of implicit neural representations (INRs) remains constrained by a *capacity–convergence gap* rooted in the tight coupling between activation families and their initialization schemes. High-capacity activations—sinusoidal (SIREN), wavelet-based/Gabor (WIRE), Gaussian, and variable-periodic (FINER) [18, 27, 28, 36] — provide rich spectral control but can be acutely sensitive to initialization; conversely, simpler, more stable choices converge reliably yet underfit high-frequency content. Initialization strategies (e.g., SIREN’s scale-preserving design) are therefore not interchangeable: the optimal settings depend on activation-specific properties, yielding a high-dimensional, non-convex search landscape where activation and initialization cannot be tuned independently. In practice, small hyperparameter changes can shift performance by over 10 dB PSNR on the same task, yet prevailing workflows still rely on manual, heuristic-driven tuning or coarse grid search. These observations underscore that bridging the capacity–convergence gap requires **joint, principled optimization** of activation selection and initialization to achieve stable training, strong generalization, and robust performance.

To bridge the capacity–convergence gap and move beyond heuristic tuning, we introduce OptiINR (Optimal INR Configuration via Bayesian Optimization), a unified framework that recasts INR configuration as a formal global-optimization problem over a high-dimensional, mixed-variable space. Because each evaluation entails end-to-end training, we employ Bayesian optimization [13, 38] — designed for expensive black-box objectives — to navigate a comprehensive search space spanning activation families (e.g. SIREN, WIRE, FINER, Gauss, FR) [12, 18, 27, 28, 36] and their conditional hyperparameters (e.g. base frequency, spread/scale, initialization scaling) [36, 39]. Activation selection is modeled as categorical, while associated parameters are continuous and conditional on the chosen family, enabling sample-efficient exploration of the complex, non-linear performance landscape and discovery of high-performing, robust configurations for specific INR tasks. Unlike fragmented trial-and-error, OptiINR provides an automated, scientifically grounded procedure for configuring state-of-the-art INRs. Our contributions are as follows:

- We introduce OptiINR, a structured layer-wise INR configuration space covering activation families, activation-specific parameters, initialization, per-layer learning rates, and positional-encoding choices.
- We formulate INR configuration as a mixed discrete–continuous black-box optimization problem and use Bayesian optimization to adaptively navigate this space. Our analysis further shows that the proposed search space itself is a

- central source of performance gains, while BO provides additional refinement over grid and random search under the same budget.
- Across audio, image, occupancy, PDE, and NeRF tasks, OptiINR improves over hand-designed and automated baselines, while revealing task-dependent activation patterns that provide insight into INR inductive biases.

2 Background

Implicit Neural Representations. An Implicit Neural Representation (INR) parameterizes a continuous signal $g : \mathcal{X} \subset \mathbb{R}^d \rightarrow \mathcal{Y} \subset \mathbb{R}^m$ as a neural network f_θ , typically an L -layer MLP [17, 37], encoding the signal within its parameters θ . This paradigm offers fundamental advantages over discrete representations: resolution independence and memory efficiency [7], as storage scales with network complexity rather than sampling density. The forward pass through the network is defined recursively: $\mathbf{z}^{(0)} = \gamma(\mathbf{x})$, $\mathbf{z}^{(\ell)} = \sigma_{\mathbf{p}}(\mathbf{W}^{(\ell)}\mathbf{z}^{(\ell-1)} + \mathbf{b}^{(\ell)})$ for $\ell = 1, \dots, L-1$, and $f_\theta(\mathbf{x}) = \mathbf{W}^{(L)}\mathbf{z}^{(L-1)} + \mathbf{b}^{(L)}$, where $\theta = \{\mathbf{W}^{(\ell)}, \mathbf{b}^{(\ell)}\}_{\ell=1}^L$ are the learnable parameters with $\mathbf{W}^{(\ell)} \in \mathbb{R}^{h_\ell \times h_{\ell-1}}$ and $\mathbf{b}^{(\ell)} \in \mathbb{R}^{h_\ell}$, $\sigma_{\mathbf{p}}$ is an element-wise activation function with parameters $\mathbf{p}$, and γ is an optional coordinate encoding. Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ sampled from the ground truth signal, we optimize $\theta^* = \arg \min_\theta \frac{1}{N} \sum_{i=1}^N \ell(f_\theta(\mathbf{x}_i), \mathbf{y}_i) + \mathcal{R}(\theta)$, where ℓ is a task-specific loss function and $\mathcal{R}$ is an optional regularization term. While the Universal Approximation Theorem [4, 11] guarantees theoretical expressivity, a fundamental practical challenge is spectral bias [3, 25]: neural networks trained with gradient descent inherently learn low-frequency components before high-frequency ones, yielding overly smooth reconstructions that fail to capture fine-grained details and sharp transients in natural signals. Consequently, the performance of f_{θ^*} depends critically on architectural choices made prior to training—particularly the activation function family and parameter initialization strategy—which together determine optimization stability, frequency expressivity, and generalization capacity.

Activation Functions and Spectral Bias The evolution of activation functions in INR literature directly addresses the fundamental challenge of spectral bias. Initial attempts with standard activations like ReLU proved insufficient, necessitating positional encoding[27] - a preprocessing step mapping input coordinates to higher-dimensional Fourier feature spaces to make high-frequency variations accessible. A conceptual breakthrough came with Sinusoidal Representation Networks (SIREN), which integrate periodicity directly into the network architecture by employing $\sigma_{\mathbf{p}}(x) = \sin(\omega_0 x)$ as the primary activation. SIREN demonstrated that appropriately chosen activations could obviate positional encoding; however, their success depends critically on principled initialization schemes that preserve activation distributions across layers, highlighting the tight coupling between activation choice and initialization strategy. Subsequent research questioned the necessity of periodicity itself, producing a powerful toolkit of activation functions with distinct spectral properties. Gaussian activations, $\sigma_{\mathbf{p}}(x) = e^{-(s_0 x)^2}$, offer non-periodic alternatives with controllable spatial extent through scale parameter s_0 . Wavelet Implicit Representations (WIRE) employ Gabor wavelets [28],

$\sigma_{\mathbf{p}}(x) = e^{j\omega_0 x} e^{-|s_0 x|^2}$, valued for their optimal space-frequency concentration that minimizes the uncertainty principle—particularly suitable for visual signal representation. More recent frameworks like FINER and FINER++ [18] introduce variable-periodic functions, $\sigma_{\mathbf{p}}(x) = \sin(\omega_0(|x| + 1)x)$, which modulate local frequency based on input magnitude through adaptive bias initialization, enabling flexible spectral control across different signal regions. While this evolutionary path has produced increasingly sophisticated activation functions, each advancement introduces sensitive hyperparameters (e.g., ω_0 , s_0 , k) requiring specific initialization strategies. This proliferation creates a complex configuration landscape where performance depends critically on joint optimization of activation family, parameter values, and initialization scheme—reinforcing the need for principled, automated configuration strategies.

Automated Model Configuration The challenge of automatically configuring machine learning models is addressed by Automated Machine Learning (AutoML) and Neural Architecture Search (NAS) [8, 9]. Our work, OptiINR, operates within this paradigm to find optimal hyperparameter configurations for single, specific tasks (e.g., representing a given image). This approach is distinct from, yet complementary to, meta-learning for INRs. Meta-learning approaches such as MetaSDF or Meta-SparseINR [35] learn weight initializations from signal distributions, enabling rapid fine-tuning for unseen signals by optimizing network weights for fast adaptation across tasks. In contrast, OptiINR optimizes network hyperparameters (architecture) for maximal performance on individual target signals.

Gaussian Processes A function f drawn from a GP is denoted as $f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$, where $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ is the mean function and $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$ is the covariance (kernel) function. The kernel is a symmetric, positive semi-definite function encoding prior beliefs about function properties such as smoothness and length-scale. For regression with observed data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, a GP infers a posterior distribution over functions. A key property is that any finite collection of function values is jointly Gaussian distributed. The posterior predictive distribution for a test point $\mathbf{x}_*$ is also Gaussian: $p(f(\mathbf{x}_*)|\mathcal{D}, \mathbf{x}_*) = \mathcal{N}(\mu(\mathbf{x}_*), \sigma^2(\mathbf{x}_*))$ with predictive mean $\mu(\mathbf{x}_*) = \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}$ and variance $\sigma^2(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_*$, where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is the $n \times n$ kernel matrix, $\mathbf{k}_* = [k(\mathbf{x}_*, \mathbf{x}_1), \dots, k(\mathbf{x}_*, \mathbf{x}_n)]^T$ is the vector of covariances between test and training points, $\mathbf{y}$ is the vector of observed outputs, and σ_n^2 is the observation noise variance. The predictive mean $\mu(\mathbf{x}_*)$ provides the best estimate of the function value, while the predictive variance $\sigma^2(\mathbf{x}_*)$ quantifies uncertainty—a property fundamental to the intelligent search strategy of Bayesian optimization.

3 Method

The performance of an Implicit Neural Representation is critically sensitive to its architectural configuration, particularly the layer-wise selection of activation

functions and corresponding weight initialization schemes. This sensitivity creates a "capacity-convergence gap," where theoretically powerful architectures fail to realize their potential due to the difficulty of finding stable and effective configurations. Current practices rely on manual tuning, parameter reuse, or greedy layer-wise optimization, none of which guarantee global optimality. We propose a novel framework that recasts this complex, ad-hoc process as a formal global optimization problem, solved efficiently using Bayesian optimization to search the high-dimensional, mixed-variable space of network architectures. This principled approach automates the discovery of globally optimal configurations, moving beyond the limitations of existing methods.

Bayesian Optimization for Expensive Black-Box Functions. Bayesian optimization is a sample-efficient framework for optimizing expensive black-box objectives [13, 31, 38]. In our setting, each objective evaluation corresponds to fully training an INR under a configuration λ and measuring its reconstruction quality, yielding $\lambda^* = \arg \max_{\lambda \in \Lambda} f(\lambda)$. BO maintains a GP surrogate over observed configuration–performance pairs and uses an acquisition function $\alpha(\lambda)$ to select the next configuration by balancing exploration and exploitation: $\lambda_{\text{next}} = \arg \max_{\lambda \in \Lambda} \alpha(\lambda)$.

3.1 INR Configuration as a Global Optimization Problem

The central novelty of our work lies in formalizing the entire INR design process as a single, unified optimization problem. The performance of an INR is critically determined by the interplay between activation functions and weight initialization strategies on a layer-by-layer basis. Previous automated methods, such as MIRE, address this by constructing networks greedily, selecting the optimal activation for each layer in a sequential manner. However, such a layer-wise greedy strategy cannot guarantee global optimality, as the optimal choice for any given layer is inherently conditioned on the choices made for all other layers (see Theorem A.1 in the supplementary materials for details).

We instead define a global configuration vector $\mathbf{A}$ that simultaneously parameterizes the choices for all L layers of the network. For each layer $l \in \{1, \dots, L\}$, we specify a configuration tuple $\lambda_l = (\sigma_l, \mathcal{I}_l, \mathbf{p}_l)$, where σ_l is a categorical variable denoting the activation function and $\mathcal{I}_l \in \{0, 1\}$ is a binary variable indicating whether SIREN-style initialization is employed. The continuous hyperparameter vector $\mathbf{p}_l \in \mathbb{R}^{d_p}$ encapsulates several critical per-layer parameters: activation-specific values (e.g., frequency ω_0 or scale s_0 , conditional on the choice of σ_l), the initial range of the layer’s weights, and a per-layer learning rate. The complete network configuration is then the concatenation of these layer-wise tuples: $\mathbf{A}_{\text{network}} = (\lambda_1, \lambda_2, \dots, \lambda_L) \in \mathcal{L}$, where $\mathcal{L}$ denotes the resulting high-dimensional, mixed-type configuration space. Our objective is to find the optimal configuration $\mathbf{A}^* = \arg \max_{\mathbf{A} \in \mathcal{L}} f(\mathbf{A})$, where $f(\mathbf{A})$ measures the performance of the INR (e.g., validation PSNR) after full training under the specified configuration. This evaluation constitutes the expensive black-box function that we seek to optimize.

3.2 Surrogate Modeling of INR Configuration

A Product Kernel for Mixed-Variable Spaces. Our configuration vector $\mathbf{A}$ lives in a product space $\mathcal{X} = \mathcal{X}_{\text{cont}} \times \mathcal{X}_{\text{cat}}$, comprising continuous and categorical variables [15, 32]. To model the correlation structure over this space, we design a product kernel that separates contributions from each variable type: $k(\mathbf{A}, \mathbf{A}') = k_{\text{cont}}(\mathbf{A}_c, \mathbf{A}'_c) \times k_{\text{cat}}(\mathbf{A}_{\text{cat}}, \mathbf{A}'_{\text{cat}})$. For the continuous components $\mathbf{A}_c$, we use the Matérn kernel [6, 29], which generalizes the popular Squared Exponential (RBF) kernel and provides control over the smoothness of the surrogate function via parameter ν : $k_{\text{cont}}(\mathbf{A}_c, \mathbf{A}'_c) = \frac{2^{1-\nu}}{\Gamma(\nu)} (\sqrt{2\nu} \frac{\|\mathbf{A}_c - \mathbf{A}'_c\|_2}{\ell})^\nu K_\nu(\sqrt{2\nu} \frac{\|\mathbf{A}_c - \mathbf{A}'_c\|_2}{\ell})$, where ℓ is the length-scale and K_ν is the modified Bessel function. We use a Matérn kernel as a standard BO choice for potentially non-smooth black-box objectives; its lengthscale and noise parameters are fitted by maximizing the GP marginal likelihood. For the categorical components $\mathbf{A}_{\text{cat}}$, we first transform them into a continuous space using one-hot encoding, where a categorical variable with M levels is mapped to an M -dimensional binary vector. We then define k_{cat} as a Squared Exponential kernel with Automatic Relevance Determination (ARD): $k_{\text{cat}}(\mathbf{A}_{\text{cat}}, \mathbf{A}'_{\text{cat}}) = \exp(-\sum_{j=1}^M \frac{(\mathbf{A}_{\text{cat},j} - \mathbf{A}'_{\text{cat},j})^2}{2\ell_j^2})$, where each dimension has a unique length-scale ℓ_j . This design ensures that our kernel is valid and that, as the number of evaluations grows, the posterior variance of the GP concentrates around the true function $f(\mathbf{A})$ (see Theorem A.3 in the supplementary materials for details).

3.3 Acquisition Optimization and Posterior Sampling

The search for the next point to evaluate is guided by an acquisition function $\alpha : \mathcal{X} \rightarrow \mathbb{R}$ that balances exploration of uncertain regions with exploitation of promising areas. We adopt a Monte Carlo-based Empirical Expected Improvement (EEI) to overcome limitations of the analytic Expected Improvement (EI) function. While analytic EI admits a closed form for Gaussian posteriors in sequential settings, it becomes intractable for batch queries and exhibits sensitivity to model misspecification.

Expected Improvement Let $f : \mathcal{X} \rightarrow \mathbb{R}$ denote our objective function with GP prior $f \sim \mathcal{GP}(m_0, k_0)$. Given observations $\mathcal{D}_n = \{(\mathbf{A}_i, y_i)\}_{i=1}^n$ where $y_i = f(\mathbf{A}_i) + \epsilon_i$ with $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$, and current best observation $f_{\text{best}} = \max_{i \in [n]} y_i$, the improvement function is defined as: $I(\mathbf{A}) = \max\{0, f(\mathbf{A}) - f_{\text{best}}\} = [f(\mathbf{A}) - f_{\text{best}}]_+$. The Expected Improvement [13] is the expectation of this improvement under the posterior measure:

$$\text{EI}(\mathbf{A}) = \mathbb{E}_{f(\mathbf{A}) \sim p(\cdot | \mathcal{D}_n)}[I(\mathbf{A})] = \int_{\mathbb{R}} [t - f_{\text{best}}]_+ p(f(\mathbf{A}) = t | \mathcal{D}_n) dt$$

Under the GP posterior $f(\mathbf{A}) | \mathcal{D}_n \sim \mathcal{N}(\mu_n(\mathbf{A}), \sigma_n^2(\mathbf{A}))$, this admits the analytic form: $\text{EI}(\mathbf{A}) = \sigma_n(\mathbf{A})[\phi(Z)\Phi(Z) + Z]$ where $Z = \frac{\mu_n(\mathbf{A}) - f_{\text{best}}}{\sigma_n(\mathbf{A})}$, and ϕ, Φ denote the standard normal PDF and CDF respectively.

Monte Carlo Approximation. For batch optimization and robustness to model misspecification, we employ a Monte Carlo estimator. Let $\{f^{(s)}\}_{s=1}^S$ be i.i.d. samples from the posterior process. The Empirical Expected Improvement [42] is:

$$\widehat{\text{EI}}(\boldsymbol{\lambda}) = \frac{1}{S} \sum_{s=1}^S [f^{(s)}(\boldsymbol{\lambda}) - f_{\text{best}}]_+$$

By the Strong Law of Large Numbers, $\widehat{\text{EI}}(\boldsymbol{\lambda}) \xrightarrow{a.s.} \text{EI}(\boldsymbol{\lambda})$ as $S \rightarrow \infty$. The convergence rate follows $\mathbb{E}[|\widehat{\text{EI}}(\boldsymbol{\lambda}) - \text{EI}(\boldsymbol{\lambda})|^2] = \mathcal{O}(S^{-1})$ by the Central Limit Theorem.

Algorithm 1 OptiNR: Bayesian Optimization for INR Configuration

- 1: **Input:** $f(\cdot)$, search space $\mathcal{L}$, N_{init} , T
 - 2: **Init:** GP surrogate with mixed-variable product kernel k
 - 3: Sample $\{\boldsymbol{A}_i\}_{i=1}^{N_{\text{init}}} \sim \mathcal{L}$ (space-filling) and evaluate $\mathcal{D}_{N_{\text{init}}} = \{(\boldsymbol{A}_i, f(\boldsymbol{A}_i))\}$
 - 4: **for** $t = N_{\text{init}}, \dots, T - 1$ **do**
 - 5: Fit GP on $\mathcal{D}_t$
 - 6: $\boldsymbol{A}_{t+1} = \arg \max_{\boldsymbol{A} \in \mathcal{L}} \widehat{\text{EI}}(\boldsymbol{A} \mid \mathcal{D}_t)$ ▷ Alg. 2
 - 7: $y_{t+1} = f(\boldsymbol{A}_{t+1}); \mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\boldsymbol{A}_{t+1}, y_{t+1})\}$
 - 8: **end for**
 - 9: **Return:** $\boldsymbol{A}^* = \arg \max_{(\boldsymbol{A}, y) \in \mathcal{D}_T} y$
-

Algorithm 2 Empirical Expected Improvement (EEI)

- 1: **Input:** candidate $\boldsymbol{A}$, data $\mathcal{D}_t = \{(\mathbf{X}, \mathbf{y})\}$, y_{best} , samples S
 - 2: Precompute $\mathbf{W} = [k(\mathbf{X}, \mathbf{X}) + \sigma_n^2 \mathbf{I}]^{-1}$
 - 3: $I \leftarrow 0$
 - 4: **for** $s = 1, \dots, S$ **do**
 - 5: Draw prior sample $f^{(s)} \sim \mathcal{GP}(0, k)$ and evaluate $\mathbf{y}_{\text{prior}}^{(s)} = f^{(s)}(\mathbf{X})$, $y_{\text{cand}}^{(s)} = f^{(s)}(\boldsymbol{A})$
 - 6: $y_{\text{post}}^{(s)} = y_{\text{cand}}^{(s)} + k(\boldsymbol{A}, \mathbf{X}) \mathbf{W} (\mathbf{y} - \mathbf{y}_{\text{prior}}^{(s)})$ ▷ Matheron's rule
 - 7: $I \leftarrow I + \max(0, y_{\text{post}}^{(s)} - y_{\text{best}})$
 - 8: **end for**
 - 9: **Return:** $\widehat{\text{EI}}(\boldsymbol{A}) = I/S$
-

Posterior sampling implementation. Direct posterior sampling requires computing the Cholesky decomposition of $\mathbf{K}_n + \sigma_n^2 \mathbf{I} \in \mathbb{R}^{n \times n}$, incurring $\mathcal{O}(n^3)$ cost per sample. For S samples, this yields prohibitive $\mathcal{O}(Sn^3)$ complexity. Alternative approaches such as random Fourier features or sparse GPs sacrifice posterior accuracy, which is critical for our high-dimensional optimization problem. We leverage Matheron's rule [5, 29] (also known as the conditional simulation formula) for exact posterior sampling with dramatically reduced computational

cost. This approach offers critical advantages over alternative methods. First, it provides exceptional computational efficiency by computing the expensive matrix inversion $[\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1}$ only once, reducing complexity from $\mathcal{O}(Sn^3)$ to $\mathcal{O}(n^3 + Sn^2)$ for S samples (see Theorem A.4 in the supplementary materials for details). Second, unlike approximation methods such as inducing points or random features, Matheron’s rule produces exact samples from the true posterior distribution, preserving the GP’s uncertainty quantification that is crucial for balancing exploration and exploitation in our optimization problem. Third, once the weight vector $\mathbf{w}$ is computed, posterior evaluations at different points can be parallelized across samples and query locations, enabling efficient GPU utilization and further accelerating the optimization process. Algorithm 1 outlines the complete workflow for discovering optimal INR configurations with our Bayesian optimization framework.

Theorem (Matheron’s Rule): Let $f \sim \mathcal{GP}(m_0, k_0)$ be a GP prior and $\mathcal{D}_n = \{(\mathbf{X}, \mathbf{y})\}$ be observations. A sample from the posterior process can be expressed as: $f_{\text{post}}(\cdot) \stackrel{d}{=} f_{\text{prior}}(\cdot) + \mathbf{k}(\cdot, \mathbf{X})[\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1}(\mathbf{y} - f_{\text{prior}}(\mathbf{X}))$, where $f_{\text{prior}} \sim \mathcal{GP}(m_0, k_0)$, $\mathbf{K}_{ij} = k_0(\mathbf{x}_i, \mathbf{x}_j)$, and $\stackrel{d}{=}$ denotes equality in distribution. This decomposition enables the following efficient sampling procedure: first, draw one sample path $f_{\text{prior}} \sim \mathcal{GP}(m_0, k_0)$ using random Fourier features or inducing points; second, compute the weight vector $\mathbf{w} = [\mathbf{K} + \sigma_n^2 \mathbf{I}]^{-1}(\mathbf{y} - f_{\text{prior}}(\mathbf{X}))$ once; third, for any query point $\boldsymbol{\lambda}$, compute $f_{\text{post}}(\boldsymbol{\lambda}) = f_{\text{prior}}(\boldsymbol{\lambda}) + \mathbf{k}(\boldsymbol{\lambda}, \mathbf{X})\mathbf{w}$. The computational complexity is $\mathcal{O}(n^3)$ for the initial matrix inversion plus $\mathcal{O}(n)$ per query point evaluation, amortizing the cost across S samples. This methodology provides a principled, globally-aware strategy for exploring the mixed-variable configuration space $\mathcal{X}$, capturing complex interdependencies between layers, activation functions, and initialization schemes to discover high-performing architectures in a fully automated fashion.

4 Related Work

Our work builds upon three core areas of research: the development of implicit neural representations, the design of specialized activation functions to overcome spectral bias, and the application of automated machine learning to architectural design. **Implicit Neural Representations.** The paradigm of representing signals as continuous functions parameterized by coordinate-based MLPs has fundamentally reshaped fields like 3D vision and computer graphics [17]. Foundational works such as DeepSDF [23] and Occupancy Networks [21] demonstrated the efficacy of INRs for high-fidelity 3D shape modeling. This was famously extended to novel view synthesis with Neural Radiance Fields (NeRF) [22], cementing INRs as a powerful, resolution-agnostic alternative to traditional discrete representations. We note that Gaussian splatting methods [41, 44] offer an alternative, non-INR paradigm for signal representation; our work focuses specifically on coordinate-MLP architectures. **Activation Functions and Spectral Bias.** A primary challenge in training INRs is the inherent spectral bias of

standard MLPs, which struggle to learn high-frequency functions [25]. Early solutions relied on positional encoding with Fourier features to inject high-frequency information at the input layer [40]. A significant breakthrough came with Sinusoidal Representation Networks (SIRENs) [36], which showed that using periodic activation functions throughout the network could natively represent fine details. The success of SIREN spurred an explosion of research into alternative activation functions, each with a unique inductive bias, including wavelet-based (WIRE) [28], Gaussian [27], and variable-periodic (FINER) [18] activations. While this has created a rich toolkit, it has also transformed INR design into a complex configuration problem where performance is highly sensitive to the choice of activation and its initialization. **Automated Configuration for INRs.** Our work addresses this challenge by drawing from the principles of Automated Machine Learning (AutoML) and Neural Architecture Search (NAS) [8, 9]. We employ Bayesian optimization, a sample-efficient global optimization strategy well-suited for expensive black-box functions like training a neural network [38]. While most INR research relies on manual tuning, the most relevant automated approach is MIRE [12], which uses a greedy, layer-wise dictionary learning method to select activations. However, its sequential nature cannot guarantee global optimality. Kolmogorov–Arnold Networks (KANs) [19] replace fixed scalar nonlinearities with learnable spline-like univariate functions, providing a flexible alternative to standard activation design. Our framework, **OptiINR**, distinguishes itself by performing a global, joint optimization over all layers simultaneously. This approach is also distinct from meta-learning frameworks like MetaSDF [35, 39], which learn priors for fast adaptation to new signals, whereas our goal is to find the single best-performing architecture for a specific, individual signal.

5 Experiments

To rigorously validate the OptiINR framework, we designed a comprehensive suite of experiments aimed at answering three central research questions. First, can a principled, global optimization framework discover configurations that consistently and significantly outperform state-of-the-art, manually-tuned baselines across diverse signal modalities? Second, does the framework’s efficacy scale from low-dimensional signals to more complex, high-dimensional representations? Third, and most critically, can the architectures discovered through automated search reveal novel, generalizable design principles that challenge or refine conventional heuristics in INR design? Through meticulous quantitative and qualitative analysis across multiple canonical tasks [18, 28, 36], we demonstrate that OptiINR not only automates the configuration process but also reveals task-dependent design patterns that challenge conventional heuristics in INR design.

Experimental Protocol All experiments were conducted using PyTorch [24], with the Bayesian optimization component implemented via the BoTorch library [2]. To isolate the impact of network configuration, we employed a consistent base architecture across all evaluated models: a four-layer MLP with 256 hidden

units per layer. For fixed baselines, we use the standard or author-recommended learning rate reported for each method. For OptiINR, the base optimizer is AdamW [20], while layer-wise learning rates are part of the searched configuration space. Unless otherwise specified, all evaluated configurations are trained for 10,000 iterations under the same architecture and data protocol. All evaluations were performed on NVIDIA B200 GPUs. This standardized setup ensures that performance differences are attributable solely to the configuration—the layer-wise combination of activations and initializations—which is the primary variable under investigation.

OptiINR Configuration Space: The core of our method is the structured, mixed-variable search space, which OptiINR navigates to find optimal configurations. This space encompasses critical design choices for a 4-layer network architecture. A binary variable determines whether to use standard Fourier feature positional encoding [40], with a corresponding continuous parameter if PE is used, controlling the scale of input coordinate mapping. For each of the four hidden layers, a categorical variable selects the activation function from task-specific sets: {SIREN, FINER} for audio representation, {SIREN, FINER, FINER++, WIRE} for image representation, and {FINER, Gauss, FINER++, WIRE} for 3D shape representation, enabling discovery of heterogeneous architectures tailored to each signal modality. Each selected activation function has an associated continuous hyperparameter (e.g., ω_0) that is jointly optimized, allowing fine-tuning of the activation’s spectral properties on a per-layer basis. Additionally, to account for varying optimization dynamics across network depth, each of the four layers has its own independent learning rate α_l optimized as a continuous variable.

Baseline Methods: To ensure rigorous and fair comparison, OptiINR was evaluated against a comprehensive set of state-of-the-art INRs using their officially published or standard configurations, measuring OptiINR against methods operating under their ideal, author-optimized conditions. The selected baselines represent diverse inductive biases: SIREN [36], the foundational model employing periodic sinusoidal activations; FINER [18], a recent advance using variable-periodic activations for flexible spectral control; Gauss [27], a representative non-periodic activation based on locality; Wavelet (WIRE) [28], a robust model based on complex Gabor wavelets known for excellent space-frequency localization; FR [34], a recent method based on Fourier reparameterized training; and IGA [33], an improved SIREN variant incorporating inductive gradient adjustment. Our OptiINR optimization process began with 30 initial configurations generated via space-filling Latin Hypercube sampling to ensure broad initial coverage, followed by 100 iterations of Bayesian optimization to refine the search and discover optimal configurations through automated exploration-exploitation balancing.

5.1 Image Representation

Image representation serves as the canonical benchmark for INR capabilities, requiring networks to learn continuous mappings $f : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ from pixel

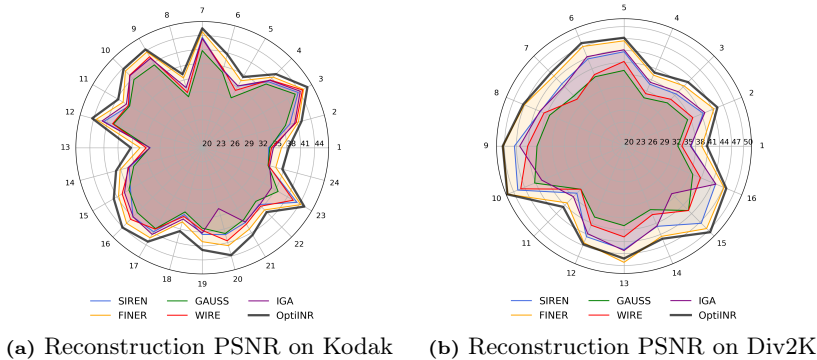


Fig. 1: Detailed per-image PSNR comparisons across all methods on Kodak and Div2K

coordinates to RGB values. This task challenges INRs to capture both smooth gradients and high-frequency details present in natural images, making it an ideal testbed for configuration optimization. Networks are provided with normalized coordinates and trained to predict corresponding RGB values over 10,000 epochs.

Datasets and Evaluation Protocol. We evaluate on two complementary benchmarks: the Kodak dataset [10] containing 24 diverse natural images at 768×512 resolution encompassing portraits, landscapes, architecture, and detailed textures; and the DIV2K dataset [1], where we use 16 cropped 512×512 patches selected for varied texture complexities and frequency characteristics, providing a challenging high-resolution testbed.

Quantitative Results. Table 1 summarizes the average PSNR and standard deviation across both datasets, demonstrating OptiINR’s substantial performance gains. On Kodak, OptiINR achieves 41.38 dB average PSNR, surpassing the strongest baseline FINER by 1.14 dB and showing substantial improvements over SIREN (2.91 dB), Gaussian activations (4.02 dB), and Fourier Reparameterization (5.48 dB). Figure 1 presents the detailed per-image PSNR comparisons across all methods, revealing that improvements are consistent across all 24 Kodak images without exception, with per-image gains ranging from 0.91 to 4.14 dB over the best baseline for each image.

On DIV2K’s high-resolution patches, OptiINR demonstrates even more pronounced advantages, achieving PSNR values from 39.99 to an 51.70 dB as shown in Table 1. The average 46.24 dB represents approximately 3–4 dB improvement over the best baselines, with particularly dramatic gains on images containing repetitive patterns or fine details where traditional INR activations fail to capture the full frequency spectrum.

Configuration Adaptation Analysis. The configurations discovered by OptiINR reveal sophisticated adaptation to underlying image characteristics. This behavior is illustrated qualitatively for two representative images from the Kodak dataset in Figure 2, which presents the final reconstructions. To further highlight the performance differences across methods, Figures 3 and 4 in the supplementary

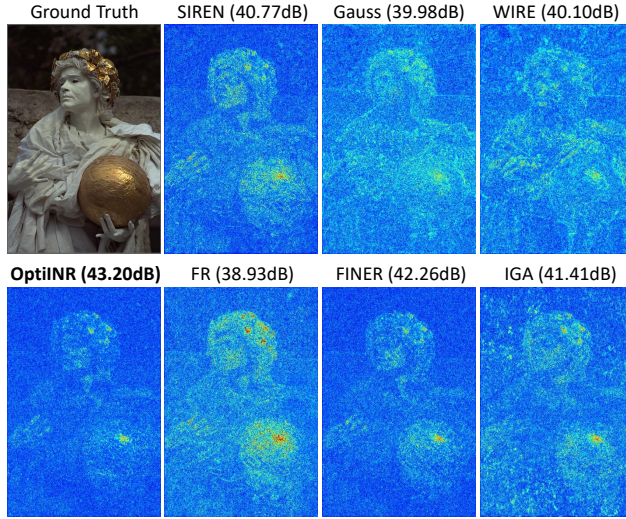


Fig. 2: Residual heatmap visualization on Kodak 17 with respect to the reference image. For each baseline, we compute per-pixel absolute differences to the reference (averaged over RGB), normalize them to $[0, 1]$, and enhance visibility using gain (GAIN=16) and gamma ($\gamma=0.6$). The residuals are colorized using the jet colormap, where blue indicates low error and red indicates high error, and they are overlaid on the reconstructed image with an opacity of 0.85.

materials display the corresponding error maps for all evaluated approaches. For smooth, low-frequency content, OptiNR tends to select Gaussian or FINER++ activations in early layers to promote smooth interpolation, followed by periodic activations (e.g., SIREN) in deeper layers to capture residual high-frequency components. For texture-rich images with prominent edges, OptiNR favors wavelet-based activations (e.g., WIRE) throughout the network, leveraging their superior space-frequency localization. This automatic adaptation eliminates the need for manual parameter tuning, where even a single misconfiguration can degrade performance by several dB. Notably, OptiNR discovers novel activation combinations unexplored in prior work, such as employing band-limited functions in intermediate layers to bridge spatially-localized early features with globally periodic representations in the final layers. These design choices yield the superior reconstructions shown in Figure 2, with the reduction in reconstruction error further evidenced by the significantly attenuated error fields in Figures 3 and 4 of the supplementary materials.

Optimization Strategy Comparison. To disentangle the contribution of Bayesian optimization from that of the configuration space itself, we compare OptiNR against grid search, random search, FreSh [14], and SPDER [30] on the Kodak dataset under identical evaluation budgets. OptiNR achieves the highest average PSNR (41.38 dB), while grid search (41.02 dB) and random search (41.23 dB)—when operating over OptiNR’s configuration space—already

substantially outperform FreSh (37.41 dB) and SPDER (35.03 dB), underscoring that the design of the search space is itself a principal contribution. Full results, including per-image PSNR breakdowns and a detailed computational cost analysis demonstrating that the BO overhead is negligible relative to INR training cost, are provided in Section F of the supplementary materials.

Table 1: Quantitative comparison of coordinate-based representation models across four modalities: (i) natural image representation on Kodak and DIV2K (PSNR in dB, mean $\pm$ std), (ii) audio signal reconstruction on Bach/Count/Two Spk (PSNR in dB), (iii) high-resolution 3D occupancy reconstruction at 512³ (IoU) on Dragon and Thai Statue, and (iv) NeRF-based view synthesis reconstruction quality (PSNR in dB). Higher is better for all metrics. OptiINR achieves the best performance across all evaluated tasks.

Method	Image PSNR (dB) $\uparrow$		Audio PSNR (dB) $\uparrow$			3D IoU $\uparrow$		NeRF PSNR (dB) $\uparrow$
	Kodak	DIV2K	Bach	Count	Two Spk	Dragon	Thai Statue	NeRF
SIREN	38.47 $\pm$ 3.47	42.75 $\pm$ 3.91	52.59	34.39	41.59	0.9881	0.9778	23.87
Gauss	37.36 $\pm$ 3.11	38.48 $\pm$ 3.13	16.49	21.32	17.21	0.9934	0.9871	22.25
WIRE	38.69 $\pm$ 3.50	39.85 $\pm$ 3.81	17.54	21.54	24.16	0.9924	0.9861	24.63
FR	35.90 $\pm$ 2.42	38.87 $\pm$ 2.27	54.94	36.93	56.36	0.9919	0.9650	—
FINER	40.24 $\pm$ 3.23	45.56 $\pm$ 3.84	36.67	39.35	42.27	0.9897	0.9804	24.25
GF	38.47 $\pm$ 4.50	40.57 $\pm$ 5.54	—	—	—	—	—	—
IGA	38.27 $\pm$ 3.43	41.77 $\pm$ 3.24	52.35	34.41	42.39	0.9919	0.9834	—
ReLU+PE	—	—	—	—	—	—	—	25.05
OptiINR (ours)	41.38 $\pm$ 3.05	46.24 $\pm$ 3.49	60.84	49.60	68.39	0.9936	0.9884	25.63

5.2 Audio Reconstruction

Datasets and Evaluation Protocol. We evaluate on three standard audio signals from the SIREN [36] benchmark: Bach (complex polyphonic composition with intricate harmonic structures), Counting (speech with distinct phonetic transitions), and Two Speakers (overlapping voices requiring separation of distinct characteristics). Following established protocols, the output layer was initialized with $\mathcal{U}(-10^{-4}, 10^{-4})$ distribution and zero biases for stable training, input coordinates were mapped to $[-100, 100]$, and models were trained for 10,000 iterations.

Quantitative Results. Table 1 demonstrates substantial gains across all audio signals. OptiINR attains 60.84 dB on Bach, surpassing FR by 5.90 dB and SIREN by 8.25 dB; 49.60 dB on Counting versus FINER’s 39.35 dB (a 10.25 dB gain); and 68.39 dB on Two Speakers versus FR’s 56.36 dB (a 12.03 dB gain). These improvements correspond to orders-of-magnitude reductions in reconstruction error, with OptiINR attaining near-machine-precision loss ($\approx 10^{-6}$) while baselines exhibit losses 3–4 orders of magnitude higher. Figures 5 and 6 in the supplementary materials present waveform and spectral comparisons: the signal reconstructed by OptiINR is visually indistinguishable from the ground truth, whereas baselines exhibit pronounced distortions. The spectral residual for OptiINR is nearly uniform across all frequencies, confirming its ability to

preserve subtle audio characteristics—including room acoustics, instrumental timbres, and vocal inflections—that are largely lost in baseline reconstructions.

5.3 3D Shape Representation: Occupancy Reconstruction

Three-dimensional shape representation through occupancy fields tests INRs’ ability to model complex geometric structures and maintain topological consistency across multiple spatial scales. This task involves learning a function $f : \mathbb{R}^3 \rightarrow \{0, 1\}$ following the occupancy network formulation [21] that maps voxel coordinates to binary occupancy values, where 1 indicates object presence and 0 denotes empty space, effectively acting as a 3D point classifier.

Dataset and Experimental Setup. We evaluate on high-resolution models from the Stanford 3D Scanning Repository [16]: the Dragon and Thai Statue, chosen for their intricate geometric details and varied surface characteristics. Both models were voxelized at 512^3 resolution, providing a challenging testbed for precise boundary representation. Performance is measured using Intersection over Union (IoU), which captures occupancy quality while ignoring the large number of trivial true negatives.

Quantitative Results. Table 1 demonstrates OptiINR’s consistent gains in geometric accuracy. On Dragon, OptiINR attains an IoU of 0.9936 compared to 0.9934 for the strongest baseline (Gaussian activations); although the absolute improvement of 0.0002 may appear modest, on a 512^3 grid this corresponds to approximately 2.7×10^4 additional correctly classified voxels, predominantly concentrated in high-curvature regions such as scales, wing membranes, and facial details. On Thai Statue, OptiINR achieves an IoU of 0.9884 versus 0.9871, with improvements primarily localized to carved motifs and thin protrusions that demand precise spatial resolution. Reconstruction visualizations are provided in Figures 7 and 8 in the supplementary materials.

Extension to PDE Signals. To further assess generality, we evaluate OptiINR on a spatiotemporal PDE reconstruction task (advection–diffusion tracer field from the Shear Flow benchmark). OptiINR achieves 57.02 dB, surpassing the strongest baseline (SIREN, 53.96 dB) by 3.06 dB; full details are provided in Appendix E.

5.4 Neural Radiance Fields

Dataset and Experimental Setup. We evaluate OptiINR on the Lego scene from the synthetic NeRF dataset using the vanilla implementation of NeRF [22]. All methods use the original 8-layer, 256-width MLP. Following standard NeRF practice, we apply positional encoding (PE) with 10 frequencies for 3D coordinates (`multires=10`) and 4 frequencies for view directions (`multires_views=4`). To ensure a fair comparison across INR activations, we unify the optimizer settings by training all baselines with Adam at a fixed learning rate of 5×10^{-5} for 30,000 iterations while keeping the sampling pipeline, ray-marching hyperparameters, and dataset splits unchanged (100 train views, 13 validation views, 25 test views, `testskip=8`). ReLU uses the standard NeRF positional encoding, while other

INR activations follow the established practice in INR reconstruction benchmarks and are evaluated both without PE (their default setting) and under OptiINR’s BO-driven configuration search. OptiINR performs BO over the configuration space consisting of PE usage, per-layer activation families and initialization parameters, and layerwise learning rates.

Quantitative Results. Table 1 reports PSNR on held-out test views. When trained with unified settings, ReLU+PE reaches 25.05 dB, substantially stronger than the classical INR baselines without PE (SIREN 23.87 dB, Gauss 22.25 dB, FINER 24.25 dB, and WIRE 24.63 dB). This aligns with prior observations that PE is essential for ReLU-based NeRFs but does not trivially benefit INR activations designed to encode high frequencies directly. OptiINR achieves 25.63 dB, outperforming the strongest baseline (WIRE, 24.63 dB) by 1.00 dB and surpassing the improved ReLU+PE baseline by 0.58 dB. In NeRF benchmarks—where architectural or sampling changes typically yield only 0.3–0.5 dB improvements—this 0.6–1.0 dB margin obtained purely from configuration optimization is substantial. These results indicate that activation and initialization choices are a critical yet under-explored component of radiance-field modeling.

Configuration Adaptation Analysis. The configurations discovered by OptiINR show a consistent pattern across trials. Early layers tend to adopt smoother or band-limited activations to improve coarse geometry stability under volumetric rendering, whereas deeper layers select more oscillatory or directional activations (e.g., SIREN- or WIRE-like families) to capture fine view-dependent effects. BO also identifies a non-uniform layerwise learning-rate pattern, assigning larger rates to shallow layers for rapid global structure fitting and smaller rates to deeper layers for stable refinement of high-frequency appearance. These adaptations arise automatically without modifying the NeRF architecture or sampling process, illustrating OptiINR’s ability to uncover non-trivial, task-dependent INR configurations.

6 Conclusion

We recast INR configuration as a global optimization problem, replacing manual tuning and ad-hoc heuristics with a principled, automated alternative. OptiINR jointly selects activation functions and initialization schemes via Bayesian optimization in a unified, sample-efficient manner. Across five tasks—2D image, 1D audio, 3D occupancy, spatiotemporal PDE, and neural radiance field synthesis—OptiINR consistently outperforms hand-tuned baselines and prior automated methods. The discovered configurations are strongly task-dependent, exposing the limits of one-size-fits-all design rules. OptiINR provides an extensible foundation for INR design that scales with evaluation budgets and helps close the capacity–convergence gap.

Bibliography

- [1] Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 126–135 (2017)
- [2] Balandat, M., Karrer, B., Jiang, D.R., Daulton, S., Letham, B., Wilson, A.G., Bakshy, E.: BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization. In: Advances in Neural Information Processing Systems 33 (2020), <http://arxiv.org/abs/1910.06403>, accessed 30 June 2026
- [3] Canatar, A., Bordelon, B., Pehlevan, C.: Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications* **12**(1), 2914 (2021)
- [4] Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems* **2**(4), 303–314 (1989)
- [5] Daulton, S., Wan, X., Eriksson, D., Balandat, M., Osborne, M.A., Bakshy, E.: Bayesian optimization over discrete and mixed spaces via probabilistic reparameterization. *Advances in Neural Information Processing Systems* **35**, 12760–12774 (2022)
- [6] Daxberger, E., Makarova, A., Turchetta, M., Krause, A.: Mixed-variable bayesian optimization. *arXiv preprint arXiv:1907.01329* (2019)
- [7] Dupont, E., Goliński, A., Alizadeh, M., Teh, Y.W., Doucet, A.: Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123* (2021)
- [8] Elsken, T., Metzen, J.H., Hutter, F.: Neural architecture search: A survey. *Journal of Machine Learning Research* **20**(55), 1–21 (2019)
- [9] Feurer, M., Hutter, F.: Hyperparameter optimization. In: *Automated machine learning: Methods, systems, challenges*, pp. 3–33. Springer International Publishing Cham (2019)
- [10] Franzen, R.: Kodak lossless true color image suite (1999)
- [11] Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural Networks* **2**(5), 359–366 (1989)
- [12] Jayasundara, D., Zhao, H., Labate, D., Patel, V.M.: Mire: Matched implicit neural representations. In: *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 8279–8288 (2025)
- [13] Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *Journal of Global optimization* **13**(4), 455–492 (1998)
- [14] Kania, A., Mihajlovic, M., Prokudin, S., Tabor, J., Spurek, P., et al.: Fresh: Frequency shifting for accelerated neural representation learning. *arXiv preprint arXiv:2410.05050* (2024)
- [15] Konakovic Lukovic, M., Tian, Y., Matusik, W.: Diversity-guided multi-objective bayesian optimization with batch evaluations. *Advances in Neural Information Processing Systems* **33** (2020)

- [16] Levoy, M., et al.: The stanford 3d scanning repository. URL <http://graphics.stanford.edu/data/3Dscanrep> (2000), accessed 30 June 2026
- [17] Li, Y., Mustikovela, S., Tewari, A., Thies, J., Wu, T., Zollhofer, M.: Neural fields in visual computing and beyond. In: ACM SIGGRAPH 2021 Courses. pp. 1–100 (2021)
- [18] Liu, Z., Zhu, H., Zhang, Q., Fu, J., Deng, W., Ma, Z., Guo, Y., Cao, X.: Finer: Flexible spectral-bias tuning in implicit neural representation by variable-periodic activation functions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2713–2722 (2024)
- [19] Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljagic, M., Hou, T., Tegmark, M.: Kan: Kolmogorov–arnold networks. In: International conference on learning representations. vol. 2025, pp. 70367–70413 (2025)
- [20] Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
- [21] Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4460–4470 (2019)
- [22] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021)
- [23] Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: 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)
- [24] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* **32** (2019)
- [25] Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International conference on machine learning. pp. 5301–5310. PMLR (2019)
- [26] Raissi, M., Perdikaris, P., Karniadakis, G.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **378**, 686–707 (2019)
- [27] Ramasinghe, S., Lucey, S.: Beyond periodicity: Towards a unifying framework for activations in coordinate-mlps. In: European Conference on Computer Vision. pp. 142–158. Springer (2022)
- [28] Saragadam, V., LeJeune, D., Tan, J., Balakrishnan, G., Veeraraghavan, A., Baraniuk, R.G.: Wire: Wavelet implicit neural representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18507–18516 (2023)
- [29] Seeger, M.: Gaussian processes for machine learning, vol. 14. World Scientific (2004)

- [30] Shah, K., Sitawarin, C.: Spder: Semiperiodic damping-enabled object representation. arXiv preprint arXiv:2306.15242 (2023)
- [31] Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N.: Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* **104**(1), 148–175 (2015)
- [32] Sheikh, H.M., Marcus, P.S.: Bayesian optimization for mixed-variable, multi-objective problems. *Structural and Multidisciplinary Optimization* **65**(11), 331 (2022). <https://doi.org/10.1007/s00158-022-03382-y>
- [33] Shi, K., Chen, H., Zhang, L., Gu, S.: Inductive gradient adjustment for spectral bias in implicit neural representations. arXiv preprint arXiv:2410.13271 (2024)
- [34] Shi, K., Zhou, X., Gu, S.: 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)
- [35] Sitzmann, V., Chan, E., Tucker, R., Snavely, N., Wetzstein, G.: Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems* **33**, 10136–10147 (2020)
- [36] Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. *Advances in neural information processing systems* **33**, 7462–7473 (2020)
- [37] Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in neural information processing systems* **32** (2019)
- [38] Snoek, J., Larochelle, H., Adams, R.P.: Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems* **25** (2012)
- [39] Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P.P., Barron, J.T., Ng, R.: 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)
- [40] Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems* **33**, 7537–7547 (2020)
- [41] Waczynska, J., Szczepanik, T., Borycki, P., Tadeja, S., Bohné, T., Spurek, P.: Mirage: Editable 2d images using gaussian splatting. In: *Forty-second International Conference on Machine Learning* (2025), <https://openreview.net/forum?id=PQYJMq39gI>, accessed 30 June 2026
- [42] Wilson, J., Hutter, F., Deisenroth, M.: Maximizing acquisition functions for bayesian optimization. *Advances in neural information processing systems* **31** (2018)
- [43] Xie, Y., Gu, J., Tancik, M., Chen, Q., Liu, S., Li, Y., Liu, L., Thies, J., Wu, T., Wu, K., et al.: Neural fields for visual computing. In: *ACM SIGGRAPH 2022 Courses*. pp. 1–100 (2022)
- [44] Zhang, X., Ge, X., Xu, T., He, D., Wang, Y., Qin, H., Lu, G., Geng, J., Zhang, J.: Gaussianimage: 1000 fps image representation and compression

by 2d gaussian splatting. In: European Conference on Computer Vision. pp. 327–345. Springer (2024)