# Learnable Spectral Activations

**Tamir Shor**[1*] **Or Litany**[1,3] **Alex M. Bronstein**[1,2]
[1]Technion
[2]Institute of Science and Technology Austria
[3]NVIDIA
tamir.shor@campus.technion.ac.il orlitany@cs.technion.ac.il
alexander.bronstein@ist.ac.at

## Abstract

Implicit neural representations (INRs) are shaped by the spectral structure induced by their input encodings and activation functions. Existing methods improve fitting primarily by modifying which frequencies are available to the network, through coordinate encodings or periodic nonlinearities. However, frequency access is not the only bottleneck: signals with localized or spatially varying structure require the network to efficiently compose frequencies into multi-harmonic internal responses. We introduce learnable spectral activations (LSA), which replace fixed neuron-level nonlinearities with a residual truncated Fourier series whose harmonic amplitudes are learned during training. LSA does not expand the asymptotic function class. Instead, it changes the factorization of the representation: linear weights select features while activation coefficients control spectral shaping, and the two are updated by separate gradients. Because the activation output is affine in the coefficients given fixed pre-activations, spectral tuning becomes a more direct subproblem compared to architectures where it is entangled with feature selection. Empirically, this factorization concentrates more target-signal energy in the leading eigenmodes of the neural tangent kernel, consistent with improved optimization behavior. Across audio, image, neural radiance field, and neural acoustic field tasks, LSA also improves reconstruction quality. Code is available at https://github.com/tamirshor7/Learnable-Spectral-Activations

## 1 Introduction

Implicit neural representations (INRs), or neural fields [1, 2], parameterize signals as continuous functions, mapping coordinates directly to signal values and replacing discrete grids with compact, resolution-agnostic representations that are amenable to querying, differentiation, and optimization. They have driven progress across neural rendering [3–5], geometric scene understanding [6–11], acoustic modeling [12, 13], physical simulation and scientific computing [14–17], sensor simulation [18], and a growing list of general-purpose use cases [19, 20]. Across all of these, INR behavior is governed by two architectural choices: the input encoding that lifts coordinates into a feature space, and the nonlinear activation through which those features propagate. Together, these choices determine the family of signals the network can fit and how easily it converges to a particular target, making them, in practice, the central design question for INRs.

A productive lens for reading INR design is the structured-dictionary perspective of Yüce et al. [21]: an INR represents a signal as a weighted combination of harmonic atoms drawn from a dictionary $\mathcal{H}(\Omega)$ induced by the architecture. The input encoding determines a base set of frequencies $\Omega$, nonlinear activations generate harmonics of those frequencies, and the network learns coefficients

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

that combine the resulting atoms into the target signal. Under this view, designing an INR amounts to choosing and constraining a dictionary, and the inductive bias of an INR is essentially the bias of that dictionary [22, 23]. Most successful INR methods control this dictionary by adjusting *which* atoms it contains. Fourier features [22] prescribe $\Omega$ at the input. SIREN [2] uses a sinusoidal activation $\sin(\omega_0 u)$ whose frequency scale $\omega_0$ controls the bandwidth the network can resolve. FINER [24, 25] introduces variable-periodic activations whose local frequency depends on the pre-activation magnitude. Meta-learning approaches reshape the dictionary through the initialization [26, 27]. What unifies all of these approaches is their target: they modify *which* frequencies are available, while leaving the neuron-level nonlinearity that *shapes* those frequencies prescribed by the architecture. This creates a structural limitation that becomes visible on heterogeneous signals. Consider reconstructing an image edge – a single spatial feature that requires coordinated energy across many harmonics with specific relative amplitudes. In SIREN, each hidden unit produces a response whose harmonic content is rigidly determined by its weights. In FINER, the effective frequency varies with the input, but at any given input each unit still produces a single local oscillatory mode. To reconstruct the edge, both architectures must assemble it from neurons whose spectral content is locked to the same weights that define their input features. The spectral profile of a neuron cannot be adjusted without simultaneously altering the feature it computes.

We introduce *learnable spectral activations* (LSA), which decouple these two roles. At each layer, the fixed activation is replaced by a residual truncated Fourier series whose harmonic amplitudes are learned alongside the network weights. A residual design ensures the activation starts near the identity and progressively adapts a layer-specific harmonic profile during training. This gives each hidden unit a learned, multi-harmonic response over the same pre-activation, without changing the linear weights that select the input feature.

Crucially, consistent with the sinusoidal equivalence perspective of Morsali et al. [28], LSA does not expand the asymptotic function class: a sufficiently wide periodic INR can represent the same signals. The contribution is a *factorization* change. The linear weights determine which feature each neuron computes while the activation coefficients determine how that feature is spectrally shaped, and the two are updated by different gradients within a jointly optimized parameterization. Because the activation output is affine in the coefficients given fixed pre-activations, spectral tuning is a more direct optimization subproblem than in architectures where spectral control and feature selection are entangled in the same parameters. This factorization has a meaningful effect on training dynamics. The empirical neural tangent kernel (NTK), whose leading eigenfunctions can be interpreted as high-eigenvalue dictionary atoms that gradient descent exploits most efficiently [22, 23], concentrates substantially more target-signal energy in its top modes under LSA than under SIREN or FINER (fig. 1b).

Experimentally, we evaluate LSA across audio fitting (speech and musical signals), image reconstruction (pixel-domain and gradient-only Poisson reconstruction), neural radiance fields, and neural acoustic fields. Across these controlled activation comparisons, LSA improves reconstruction quality and reduces structured spatial and spectral residuals. Audio experiments also show more stable off-grid behavior, supporting the central conclusion that INR performance depends on how available frequencies are composed inside the network.

## 2 Related Work

This work builds on the structured dictionary perspective of INRs, presented by Yüce et al. [21]. In this view, a dictionary is a collection of elementary functions (*atoms*), whose weighted combination represents a target signal. For INRs, these atoms are induced by the coordinate mapping and activation function. Coordinates $x$ are often first uplifted to an input representation $\gamma(x)$, such as Fourier [22] or positional features, to mitigate spectral bias [29]. As shown by Yüce et al. [21], when $\gamma$ contains sinusoidal components with base frequencies $\Omega$, nonlinear activations generate harmonics, schematically, $\rho(e^{j\omega x}) = \sum_k \alpha_k e^{jk\omega x}$. As a result, many INR architectures can be viewed as representing signals as

$$f(x) = \sum_{\omega' \in \mathcal{H}(\Omega)} c_{\omega'} \sin(\langle \omega', x \rangle + \phi_{\omega'}), \tag{1}$$

where $\mathcal{H}(\Omega)$ denotes the harmonic frequencies induced by the architecture and $c_{\omega'}$ are learned coefficients. This perspective connects INR design to the atoms available to the model: input

encodings, activations, and initialization schemes determine the dictionary from which the signal is assembled.

Many INR methods can be interpreted as mechanisms for controlling this dictionary. Fourier features [22] modify the input mapping directly, using $\gamma(x) = [\sin(2\pi Bx), \cos(2\pi Bx)]$, where $B$ determines the frequencies exposed to the MLP. In random Fourier features, sampling $B_{ij} \sim \mathcal{N}(0, \sigma^2)$ makes $\sigma$ a bandwidth parameter. Larger values expose higher frequencies but can also increase aliasing risk off the training grid [21]. Positional encodings in neural rendering follow the same principle by prescribing coordinate-frequency bases before the network [3]. Activation-based INRs instead shape the dictionary inside the network. SIREN [2] uses hidden layers $z^{(\ell)} = \sin(\omega_0(W^{(\ell)}z^{(\ell-1)} + b^{(\ell)}))$, where $\omega_0$ controls the effective frequency scale. This enables high-fidelity fitting and well-behaved derivatives, since derivatives of sinusoids remain sinusoidal. FINER [24] further tunes spectral bias through a variable-periodic activation $\rho_{\mathrm{FINER}}(u) = \sin((|u| + 1)u)$. By changing the bias initialization range, different activation regions with different effective local frequencies are selected. FINER++ [25] extends this idea to a broader family of variable-periodic functions. SIREN and the FINER family show that activation design is central to INR performance, but remain primarily frequency-centric: they tune which effective frequencies are available or emphasized by a prescribed nonlinear operator. WIRE [30] and sinc activations [31] provide localized Gabor and sampling-theoretic alternatives. Scaling initialization [32] improves the training of sinusoidal fields through their initial parameter scale. SAPE [33] and modulated periodic activations [34] address related limitations through adaptive input-frequency schedules and latent-code-conditioned modulation, respectively. Multiplicative filter networks [35] compose learned filters through multiplicative interactions. These mechanisms act on different parts of INR design, including the activation, initialization, input-frequency schedule, conditioning, and network composition.

Recent methods also make the activation or internal basis itself learnable. Among them, STAF [28] is the closest conceptual baseline to LSA. STAF learns layer-shared sinusoidal mixtures with trainable amplitudes, frequencies, and phases, allowing both the sinusoidal basis and its weighting to adapt during training. LSA adopts a different inductive bias: it fixes an integer-harmonic basis of each pre-activation and learns its relative amplitudes. This choice is motivated by the structured-dictionary analysis of Yüce et al. [21], where periodic nonlinearities already induce rich harmonic dictionaries and optimization depends not only on available frequency support, but also on how the induced atoms are weighted and aligned with the target. LSA therefore focuses adaptation on the composition of coordinated harmonics within a structured basis rather than on learning the sinusoidal basis itself. The residual form supports initialization near the identity, and layer sharing keeps this adaptation low-dimensional with only $K$ additional coefficients per layer. Fourier KAN [36], SL$^2$A [37], and MIRE [38] explore learned Fourier expansions, a Chebyshev activation block within a modulated architecture, and layer-wise activation dictionary matching, respectively. These methods span both activation substitutions and broader architectural changes. We therefore distinguish controlled activation comparisons under matched backbones from broader comparisons with alternative INR formulations.

The dictionary view[21] also connects representation to optimization through the empirical neural tangent kernel (NTK)[23]: leading NTK eigenfunctions can be interpreted as high-eigenvalue atoms that gradient descent learns more rapidly [21, 23]. Thus, improving an INR is not only a matter of expanding frequency support, but also of aligning the induced basis with the target signal. Our work follows this line of reasoning but modifies a different axis of the representation. Rather than further tuning which frequencies are available, LSA learns how a structured harmonic basis is composed within each hidden response, adapting neuron-level spectral shaping while retaining a compact layer-wise parameterization.

## 3 Method

The dictionary view in Section 2 suggests that fixed INR activations prescribe the neuron-level waveform family through which hidden features are transformed. We make this operator learnable. At layer $\ell$, we parameterize the activation as a residual truncated Fourier series:

$$\phi^{(\ell)}(u) = u + \sum_{k=1}^{K} a_k^{(\ell)} \sin(2\pi k u), \tag{2}$$

where $u$ denotes a scalar pre-activation and $a_k^{(\ell)}$ are learnable layer-wise coefficients. When applied to a vector pre-activation $u^{(\ell)} = W^{(\ell)} z^{(\ell-1)} + b^{(\ell)}$, $\phi^{(\ell)}$ is applied elementwise with the same coefficients shared across channels. We integrate this activation into a standard INR pipeline. Given coordinates $x$, optionally encoded by $\gamma$, the network computes

$$z^{(0)} = \gamma(x), \qquad z^{(\ell)} = \phi^{(\ell)}\big(W^{(\ell)} z^{(\ell-1)} + b^{(\ell)}\big), \tag{3}$$

for $\ell = 1, \ldots, L-1$, followed by a linear output layer. The activation parameters $a^{(\ell)} \in \mathbb{R}^K$ are shared across neurons within each layer and learned jointly with the network weights. They are initialized with small variance and zero mean so that the model starts close to the identity-residual MLP and progressively adapts its nonlinear basis during training. Notably, like SIREN [2], LSA provides analytic derivatives through automatic differentiation, since the activation remains a smooth harmonic residual operator. Thus, the model retains the derivative-based advantages of periodic INRs for gradient supervision and physical modeling, while allowing the waveform itself to adapt during training.

Under this formulation, input encodings such as Fourier features determine the *input spectral support*, while LSA controls *internal spectral processing*. The linear weights determine which hidden features are formed, whereas the activation coefficients determine how those features are spectrally shaped. Because the learned operator is applied pointwise to each pre-activation, LSA can adapt the harmonic response of hidden units without changing their underlying feature projections.

The key distinction from prescribed INR activations such as SIREN, WIRE, and the FINER family [2, 24, 25, 30] is the factorization of spectral shaping (and, importantly, not asymptotic expressivity). Competitive INR architectures can represent complex waveforms by combining many units, but their neuron-level nonlinear operator remains prescribed. In contrast, LSA in eq. (2) applies multiple harmonic transformations to the same pre-activation $u = w^\top z$ and learns their relative amplitudes, allowing each hidden response to adapt a coordinated multi-harmonic waveform directly rather than relying solely on composition across units with predefined nonlinearities. More generally, a parameterization $u + \sum_k a_k b_k(u)$ with any fixed basis remains affine in the coefficients $a_k$. The harmonic basis specifies the structured inductive bias studied in this work.

Figure 1a illustrates this distinction on a controlled 1D phantom with a limited atom budget and matched optimization budget. SIREN and FINER improve as more atoms are added, showing that the target is representable by sufficiently many fixed units. LSA reaches the target structure with fewer atoms because each unit can synthesize a harmonic mixture. This experiment illustrates the parameter efficiency of the harmonic factorization.

To characterize these learning directions, let $w$ collect ordinary weights and biases, $a$ collect the harmonic coefficients, and $r = f_{w,a}(X) - y$ be the residual over fixed samples $X$, with output channels flattened. Define $J_w = \partial f(X)/\partial w$ and $\Psi = \partial f(X)/\partial a$. Each column $\psi_{\ell k}$ of $\Psi$ propagates $\sin(2\pi k u^{(\ell)})$ through the remaining layers. For $\mathcal{L} = \frac{1}{2}\|r\|_2^2$, full-batch gradient flow with positive group rates $\eta_w, \eta_a$ gives the exact finite-width identity

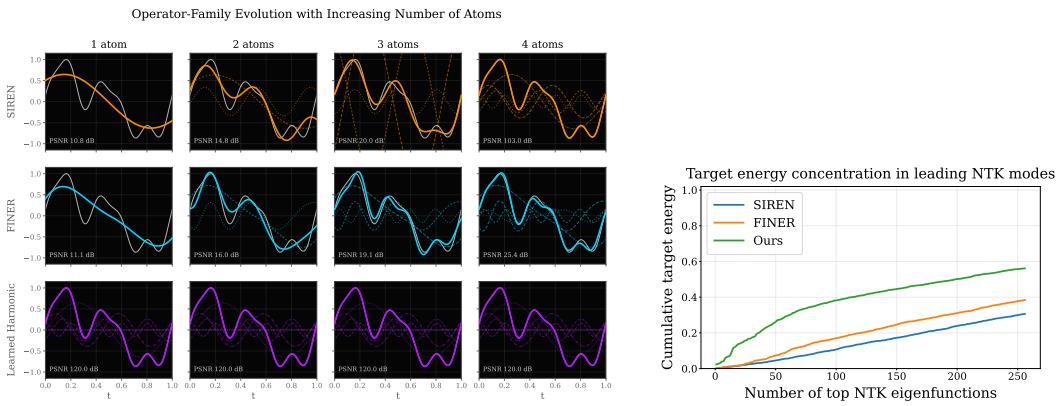
$$-\dot{\mathcal{L}} = \eta_w \|J_w^\top r\|_2^2 + \eta_a \|\Psi^\top r\|_2^2, \qquad \|\Psi^\top r\|_2^2 = \sum_{\ell,k} \langle r, \psi_{\ell k}\rangle^2. \tag{4}$$

Thus coefficient learning follows residual correlations with propagated harmonic features. At a shared initialization, the rate-weighted empirical tangent kernel [23] satisfies

$$\kappa_{\mathrm{LSA},0} = \kappa_{\mathrm{frozen},0} + \eta_a \Psi_0 \Psi_0^\top, \qquad \kappa_{\mathrm{frozen},0} = \eta_w J_{w,0} J_{w,0}^\top. \tag{5}$$

Here the frozen model holds $a$ at that same initialization. The added positive-semidefinite component contributes $\eta_a \|\Psi_0^\top r_0\|_2^2$ to the initial loss-decay rate. At $a = 0$, $\phi(u, 0) = u$ and $\partial\phi/\partial a_k = \sin(2\pi k u)$, retaining harmonic tangent directions whenever their downstream propagation is nondegenerate. Frequency and phase derivatives of $A\sin(\omega u + \varphi)$ scale with $A$. Thus the parameterization admits exact identity initialization while retaining active harmonic tangent directions. The optimization benefit is local and conditional on residual alignment. Derivations, the fixed-NTK interpretation, and controlled validations are given in Section A.6.

To further probe this NTK-based optimization view, we analyze the empirical NTK induced by each architecture. We eigendecompose the empirical NTK on a fixed set of sampled coordinates and measure how much zero-mean target energy lies in the span of the top-$m$ eigenfunctions. These




(a) (b)

Figure 1: **(a)** Operator families and parameter efficiency. Dashed curves show individual atoms and solid curves show the fitted signal. SIREN and FINER use prescribed single-unit responses, whereas LSA synthesizes multi-harmonic waveforms and reaches accurate fits with fewer atoms. **(b)** Target energy concentration in leading empirical NTK modes. LSA places more target energy in the top modes than SIREN or FINER, indicating better alignment between the induced dictionary and the target.

leading eigenfunctions correspond to high-eigenvalue directions that gradient descent learns most efficiently [21, 23]. As shown in Figure 1b, LSA captures substantially more target energy using the same number of leading modes than SIREN or FINER. This suggests that the learned activation reshapes the local optimization geometry so that the target is represented more efficiently by the high-eigenvalue subspace. Full definitions and computation details are provided in Section A.5.

# 4 Results

We evaluate learned spectral activations across characteristic INR settings. In Section 4.1 we study direct signal fitting on one-dimensional audio, where the target contains transients, harmonic structure, and time-varying envelopes. In Section 4.2 we evaluate image fitting, including both pixel-domain reconstruction and a gradient-only Poisson reconstruction task, to test whether the learned activation improves function values and derivative consistency. Next, in Section 4.3 we move from fitting individual signals to learning neural fields, using novel view synthesis and acoustic-field prediction to test whether LSA improves generalization to unseen spatial queries. In Section B.5 we further demonstrate that the same spectral parameterization can also be used beyond INR activations, by applying it to positional encodings in a sparse-to-dense reconstruction task.

**Baselines.** Our primary controlled comparisons use SIREN [2] and FINER [24], strong prescribed-activation INR baselines that allow us to isolate the effect of replacing the neuron-level nonlinear operator under matched MLP backbones and task settings. We additionally compare with STAF [28], the closest learnable-activation baseline to LSA, on both image and audio fitting using independently tuned, task-appropriate configurations. Broader INR comparisons include $SL^2A$[37], WIRE[30], MFN GaborNet[35], and fractional Jacobi blocks (fJNB) [39] on compatible tasks. Hyperparameters are selected separately for each method, with protocols reported alongside the comparisons and in Section C.

## 4.1 Audio Data

### 4.1.1 Audio Signal Fitting

We evaluate our LSA on one-dimensional audio INR fitting. We use LibriSpeech [40], a corpus of read English speech derived from LibriVox audiobooks, and NSynth [41], a dataset of four-second monophonic musical notes from a large collection of instruments. The selected clips are resampled to 48 kHz as detailed in Section A.1. These two datasets probe complementary regimes: speech

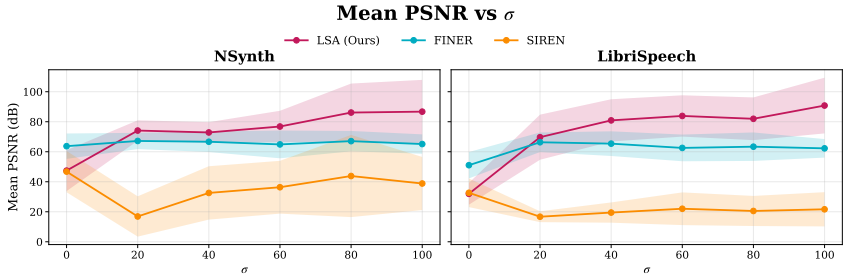



Figure 2: SIREN/FINER/LSA comparison of on-grid audio fitting across Fourier-feature bandwidths $\sigma$. LSA achieves the strongest performance at moderate and high bandwidths on both speech and musical-note signals, indicating that learned spectral activations improve the internal processing of Fourier-feature harmonics.

contains transient, nonstationary waveform structure, NSynth contains harmonic musical signals with instrument-dependent timbre and temporal envelopes. Because audio contains strong high-frequency and highly nonstationary structure, in the controlled SIREN/FINER/LSA comparison we use Fourier features [22] at the input layer and include raw coordinates as the $\sigma = 0$ setting. For STAF, we also evaluated matched Fourier-feature settings, but its native raw-coordinate configuration performed best. We sweep the Fourier bandwidth $\sigma$ and tune it independently for each method. The comparison therefore examines how each activation processes the same exposed frequency support at each bandwidth. Full implementation details are available in Section A.1.

Figure 2 reports on-grid PSNR as a function of the Fourier-feature bandwidth $\sigma$. With raw coordinates ($\sigma = 0$), FINER achieves slightly higher PSNR on both LibriSpeech and NSynth. This is consistent with our formulation: LSA does not expose a different input basis, but rather gives each layer a learnable mechanism for reshaping the spectral components made available to the network. Following Tancik et al. [22], we use $\gamma_\sigma(t) = [t, \sin(2\pi Bt), \cos(2\pi Bt)]$ with entries of $B$ drawn from $\mathcal{N}(0, \sigma^2)$ and fixed across methods for a given setting. As $\sigma$ increases, the advantage of LSA becomes clear. On LibriSpeech, LSA improves sharply from the no-FF setting and achieves the best mean performance at positive bandwidths in the sweep, reaching the highest PSNR at large bandwidths. FINER is competitive at low bandwidth but saturates around the mid-60 dB range, while SIREN remains substantially lower and more variable. This supports the view that INR fitting, especially for challenging high-frequency signals (such as speech or audio), benefits not only from exposing selected frequencies at the input, but from learning how hidden responses should synthesize transient and nonstationary waveform structure. On NSynth, the same trend holds. FINER is strong and relatively stable, reflecting the more global harmonic structure of musical notes, but LSA overtakes it once Fourier-feature support is available and maintains the best performance at moderate and high $\sigma$. SIREN is again considerably more sensitive to bandwidth, indicating that under matched Fourier-feature settings, the advantage of LSA comes from processing the exposed frequency support more effectively rather than from a different input basis.

Comparisons with STAF, SL$^2$A, WIRE, and fJNB, including raw-coordinate and Fourier-feature settings, are in Section C.2. Specifically we show that LSA also outperforms STAF in our tested setting - under the same clip manifests and 5,000-step budget, LSA reaches $86.72/90.78$ dB on NSynth/LibriSpeech, compared with $68.79/68.40$ dB for the strongest tested STAF configuration.

### 4.1.2 Continuous Approximation

In Section 4.1.1 we evaluate standard INR fitting on discrete training grids. We now examine the quality of the represented *continuous* signal between sampled coordinates. This distinction is both practical and theoretical: mitigating spectral bias [29] via high-frequency injection (e.g., Fourier features) improves fitting on training coordinates, but can introduce aliasing, where multiple high-frequency functions agree on sampled points while diverging between them [21].

To isolate this effect, we train on a fixed grid of 16k samples and evaluate both *on-grid* and an *unseen midpoint grid*, while varying high-frequency injection (governed by Fourier-feature bandwidth $\sigma$). Extending beyond the common INR use-case of memorizing training samples, in this section we evaluate how well INR performs *continuous* reconstructions, necessitating interpolation between

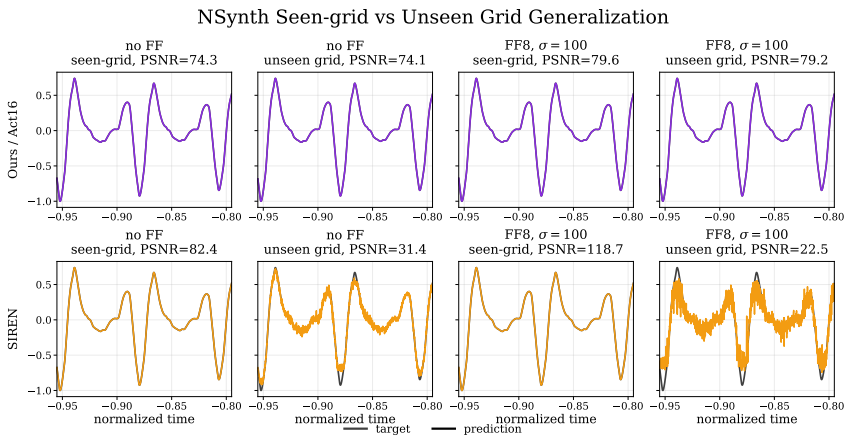



Figure 3: On-grid vs. off-grid reconstruction under increasing Fourier-feature bandwidth. SIREN exhibits a large on-grid/off-grid gap already at $\sigma = 0$ and severe interpolation artifacts for $\sigma = 100$. Our method remains stable under inference of interpolated, unseen grid points.




Figure 4: Per-image Kodak fitting comparison under fixed configurations. Per-image PSNR is shown for LSA, FINER, SIREN, and STAF under 5,000-update fits.

samples seen during INR fitting. We vary Fourier-feature bandwidth via $\sigma$, following standard practice for mitigating spectral bias. Figure 3 compares prediction on seen- and unseen-samples. For SIREN, increasing bandwidth ($\sigma$) improves on-grid PSNR but leads to severe degradation off-grid. This indicates that the model fits sampled points using high-frequency components that do not define a stable continuous function. Full version in Appendix (Fig. 9). In contrast, our method maintains consistent performance across seen and unseen grids, without the same collapse at high bandwidth. From the structured-dictionary perspective [21], increasing $\sigma$ broadens the base frequency support exposed by the input encoding and, consequently, the higher-frequency content available in the induced dictionary. While this alleviates spectral bias, it also introduces ambiguity: multiple high-frequency functions can agree on sampled points while differing between them, leading to aliasing [21]. This effect is visible for SIREN, where high-bandwidth inputs enable accurate interpolation at training coordinates but unstable behavior off-grid. We include a similar comparison with FINER in Figure 9. FINER similarly expands the supported frequency set via activation design[24], but does not inherently resolve this ambiguity when external bandwidth is large.

### 4.2 Image Data

We evaluate learned spectral activations on image fitting using the Kodak dataset [42], consisting of 24 true-color natural images of resolution $768 \times 512$. Each image is represented as a coordinate-based function using an MLP, without any input-level uplifting mechanisms applied. We compare against SIREN, FINER, and STAF. The matched-budget and extended-budget STAF protocols are distinguished in Section C.1.

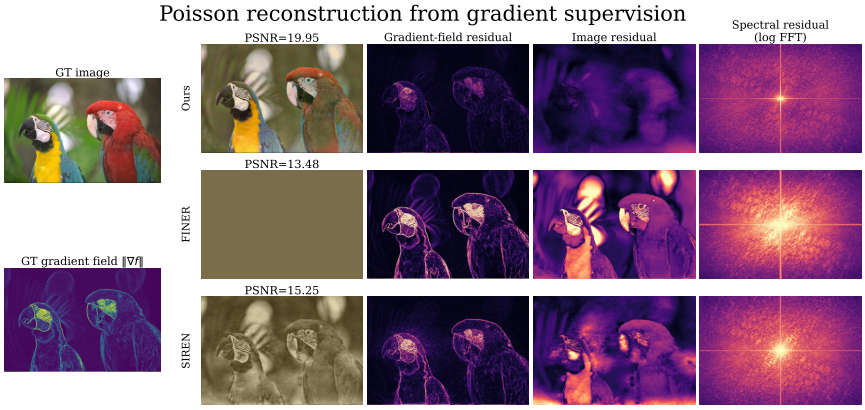



Figure 5: Poisson reconstruction from gradient-only supervision. Our method produces more consistent gradient fields and integrates them into a coherent image, while FINER and SIREN exhibit structured residuals and global reconstruction errors.

Figure 4 compares fixed configurations across all 24 Kodak images. LSA obtains a mean PSNR of 38.110 dB, compared with 37.317 dB for FINER, 35.369 dB for matched-budget global STAF, and 34.990 dB for SIREN. LSA exceeds FINER on 21/24 images and SIREN on all 24. Among the competing matched-backbone baselines, FINER achieves the highest mean PSNR. The full-set means include the six calibration images used to select the LSA and baseline configurations. Selection-independent comparisons use the other 18 images, as described in Section C.1. With per-image tuning, LSA reaches $38.308$ dB and STAF $35.425$ dB, using three and 56 candidate configurations per image, respectively. LSA wins on all 24 images, with a paired mean gain of 2.883 dB and a 95% bootstrap interval of $[2.439, 3.295]$ dB.

We compare to broader INR methods in Section C.1. On the 18-image holdout, LSA reaches $38.045$ dB with 199,043 trainable parameters, while full SL$^2$A reaches $37.857$ dB with 330,243. LSA has a $0.188$ dB higher descriptive mean while using $\sim 40\%$ fewer parameters. At a nearly matched 199,171 parameters, SL$^2$A reaches $35.910$ dB, giving LSA a 2.135 dB mean advantage with 18/18 images where LSA outperforms SL$^2$A. We refer the reader to Section C.1 for complete protocols and comparisons to WIRE[30],MFN[35], and fJNB[39].

**Poisson Reconstruction -** To evaluate derivative modeling, we consider a Poisson-style reconstruction task from image gradients. Given a target image $f_{\mathrm{GT}}$, we compute its spatial gradients and train each INR using only the derivative loss $\|\partial_x f_\theta - \partial_x f_{\mathrm{GT}}\|_2^2 + \|\partial_y f_\theta - \partial_y f_{\mathrm{GT}}\|_2^2$, without any direct pixel supervision. Since gradients determine the image only up to an additive constant, we align the mean of each reconstructed RGB channel to the target after training. This setting isolates the ability of the model to learn a derivative field that is both accurate and globally integrable. As shown in Figure 5, FINER produces highly structured gradient residuals and fails to integrate them into a coherent image, resulting in large low-frequency distortions. SIREN achieves better local structure, but exhibits global drift and inconsistent reconstruction. Our method yields lower gradient-field residuals and a more coherent reconstruction, with reduced structured artifacts in the spectral domain. These results indicate that the learned spectral activation improves the consistency between local derivatives and global signal reconstruction.

Figure 6 shows illustrative reconstructions and residuals for kodim21 and kodim13. In these examples, LSA reduces structured errors around edges and textures, with corresponding reductions in spectral residuals. The examples were selected for qualitative clarity. Dataset-wide conclusions follow from Figure 4 and the quantitative tables. Natural images contain edges, textures, and repeated fine structure that can be expressed through coordinated harmonic mixtures. Equation (2) allows each hidden response to adapt such harmonic composition directly. These observations are consistent with the harmonic-composition interpretation in Figure 1a.

The convergence curves in Section B.2 show that LSA separates from SIREN early, while its advantage over FINER emerges progressively. LSA leads FINER on 5/24 images at step 500, 16/24 at step 1,000, and 21/24 at step 2,000, finishing ahead on 21/24 at step 5,000.

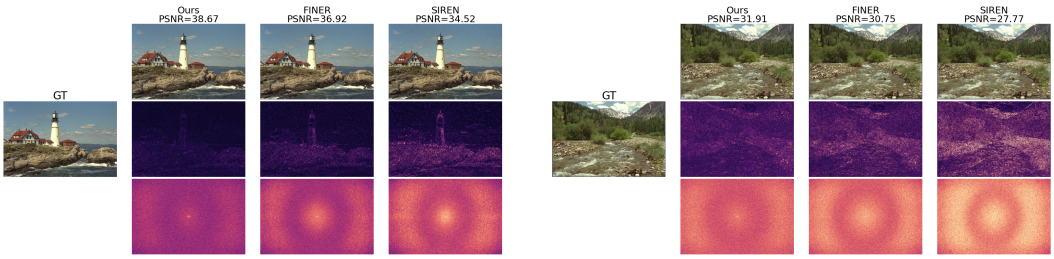



Figure 6: Qualitative Kodak reconstructions for illustrative examples kodim21 (left) and kodim13 (right), selected for qualitative clarity. LSA, FINER, and SIREN use the fixed configurations in Figure 4. We show reconstructions, spatial residuals, and spectral residuals, with a common display scale for each residual type across both images and all methods. Full-dataset comparisons are in Figure 4 and Section C.1.

Table 1: Controlled activation comparison on novel view synthesis, over Blender [3] scenes.

| Metric | Method | Chair | Drums | Ficus | Hotdog | Lego | Materials | Mic | Ship |
|---|---|---|---|---|---|---|---|---|---|
| PSNR ↑ | SIREN [2] | 33.48 | 24.89 | 28.22 | 32.85 | 29.60 | 27.13 | 33.28 | 22.48 |
| | FINER [24] | 33.90 | **24.90** | 28.70 | 33.05 | **30.04** | 27.05 | **33.96** | 22.47 |
| | Ours | **34.18** | 24.79 | **28.85** | **33.36** | 29.39 | **27.22** | 33.91 | **22.60** |
| SSIM ↑ | SIREN [2] | 0.973 | 0.912 | 0.959 | 0.960 | 0.948 | 0.932 | 0.979 | 0.797 |
| | FINER [24] | 0.973 | 0.911 | 0.958 | 0.959 | **0.951** | 0.928 | 0.981 | 0.792 |
| | Ours | **0.977** | **0.922** | **0.967** | **0.967** | 0.950 | **0.936** | **0.983** | **0.803** |
| LPIPS ↓ | SIREN [2] | 0.017 | 0.051 | 0.027 | 0.032 | 0.030 | **0.028** | 0.014 | 0.116 |
| | FINER [24] | 0.015 | 0.052 | 0.024 | 0.033 | **0.025** | 0.032 | **0.010** | 0.108 |
| | Ours | **0.013** | **0.050** | **0.021** | **0.027** | 0.026 | 0.029 | **0.010** | **0.101** |

A six-image ablation improves from 30.45 dB at $K = 4$ to 39.50 dB at $K = 32$, with no further improvement at $K = 64$. Freezing the near-identity coefficients reduces held-out Kodak PSNR from 38.05 to 19.91 dB, showing the contribution of coefficient learning. Full results are in Section C.4.

### 4.3 Neural Fields

We next evaluate LSA in neural-field settings, where the objective is to learn a continuous field that generalizes to unseen spatial queries.

**NeRFs.** For visual neural fields, we evaluate novel view synthesis on the Blender dataset [3]. The backbone INR model in all experiments is an InstantNGP [5]-style neural field. We test whether the proposed activation improves internal spectral processing beyond isolated coordinate-to-value regression, and whether it supports improved neural-field representation in a setting where the task is to generalize to unseen views. Table 1 reports results across eight object scenes. Learned spectral activations achieve the best PSNR on five scenes and the best SSIM on seven scenes, while matching or improving the strongest LPIPS score on most scenes. The gains are moderate compared to the audio and image fitting settings, but they are consistent across complementary metrics: our method improves perceptual quality and structural similarity even when PSNR differences are small. This suggests that the benefit of LSA is not restricted to one-dimensional or two-dimensional fitting, but also appears when the representation is optimized through differentiable rendering. Qualitative results are in Section B.3. Additional fJNB results and published WIRE results under the reduced-view Blender protocol are reported in Section C.3.

**Neural-Acoustic Fields.** We then test the same principle in acoustic-field modeling. Following Neural Acoustic Fields (NAF) [13], the task is to represent spatial sound propagation by mapping listener-emitter configurations to room impulse responses. Unlike the audio-fitting experiments in section 4.1.1, where the model is asked to fit one waveform in isolation, here it must learn a spatially varying field whose responses change with listener position, emitter position, room geometry, occlusion, and reflections. This makes room-impulse-response prediction a complementary field-generalization test to NeRF. In both cases, the network represents a continuous physical field,

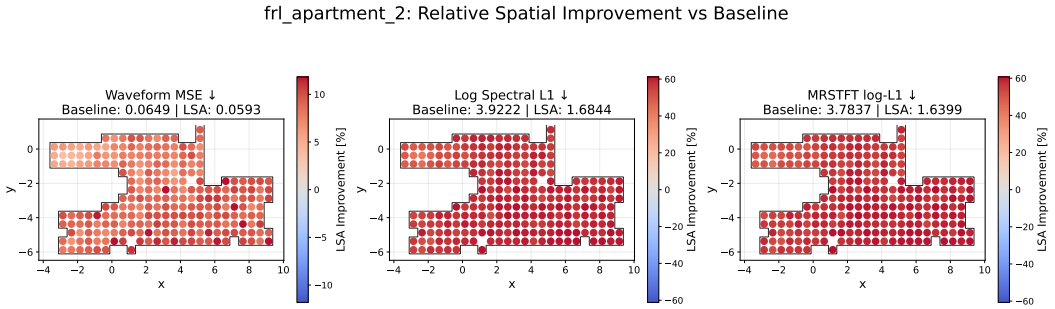



Figure 7: Spatial improvement of LSA over the NAF baseline. Each point corresponds to a spatial query in the scene, and color indicates relative improvement. LSA yields broad spatial gains, especially for log-spectral and multi-resolution STFT metrics, across the queried locations.

but the forward structure differs: NeRF predicts visual radiance accumulated along rays, while the acoustic field predicts time-frequency impulse-response structure over spatial acoustic queries.

We keep the NAF architecture and training protocol fixed and replace its baseline activation with LSA using $K = 8$ harmonics. We evaluate on benchmark scenes used in NAF [13]. Full results across the benchmark are reported in Section B.3. Figure 7 shows the spatial improvement in common acoustic reconstruction metrics after replacing standard original NAF activations with LSA. Replacing the baseline activation with LSA yields consistent gains over the original NAF architecture, especially on spectral and time-frequency metrics. Across the floor plan, LSA improves the predicted acoustic field, with particularly strong and spatially consistent gains in log-spectral and multi-resolution STFT errors. This matters because acoustic fields are highly heterogeneous: nearby walls, openings, occlusions, and reflections induce different impulse-response structures at different positions. As in NeRF, this requires composing spectral components into spatially varying field responses that remain consistent at unseen queries. Implementation details and metric definitions are in Section A.4

Taken together, the gains across NeRF and NAF settings indicate that beyond advantage in signal-fitting, LSAs also provide a more effective internal parameterization for continuous neural fields, where localized spectral structure must be composed consistently across space.

## 5 Conclusion

In this work we presented learnable spectral activations (LSA) - a simple activation-level parameterization for INRs. Our formulation makes the neuron-level waveform operator trainable by representing it as a residual harmonic expansion. This changes the granularity at which spectral structure is modeled. Hidden units can directly adapt their harmonic response, while the network weights continue to determine the features on which these responses operate. The resulting formulation provides a practical mechanism for learning internal spectral bases that are better aligned with heterogeneous and nonstationary signals. Across controlled activation comparisons in audio, images, and neural fields, LSA improves reconstruction quality and reduces structured spatial and spectral residuals. Audio experiments also show more stable off-grid behavior. Our results support the role of learned harmonic composition as a structured inductive bias for INR fitting and spatial-query prediction.

Alongside positive results, a current limitation of our method is increased computational overhead. Although LSA adds very few trainable parameters, evaluating multiple harmonic terms increases activation runtime and memory. In a representative full-grid audio setting, LSA adds 64 parameters but incurs roughly $7\times$ training-time and $9\times$ peak-memory overhead relative to FINER, with similar inference overheads. Full measurements are reported in Section D.4. More efficient fused or checkpointed implementations may reduce this cost. Future work should study more efficient and expressive variants of LSA, including neuron-specific, channel-grouped, or input-conditioned harmonic coefficients, as well as regularization of the learned spectra. Overall, our results suggest that learning the internal spectral operators of coordinate networks is a promising route toward more adaptive implicit representations.

## Acknowledgements and Disclosures

Or Litany acknowledges support from the Israel Science Foundation (grant 624/25) and the Azrieli Foundation Early Career Faculty Fellowship. The authors gratefully acknowledge this support. This research was supported by the Council for Higher Education in Israel under the Moonshot Project. The authors have no competing financial interests to report.

## References

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

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

[3] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65 (1):99–106, 2021.

[4] Sida Peng, Yuanqing Zhang, Yinghao Xu, Qianqian Wang, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9054–9063, 2021.

[5] 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.

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

[7] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[8] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6001–6010, 2020.

[9] Francis Williams, Zan Gojcic, Sameh Khamis, Denis Zorin, Joan Bruna, Sanja Fidler, and Or Litany. Neural fields as learnable kernels for 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18500–18510, 2022.

[10] Jiahui Huang, Zan Gojcic, Matan Atzmon, Or Litany, Sanja Fidler, and Francis Williams. Neural kernel surface reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4369–4379, 2023.

[11] Jiawei Yang, Boris Ivanovic, Or Litany, Xinshuo Weng, Seung Wook Kim, Boyi Li, Tong Che, Danfei Xu, Sanja Fidler, Marco Pavone, and Yue Wang. Emernerf: Emergent spatial-temporal scene decomposition via self-supervision. *arXiv preprint arXiv:2311.02077*, 2023.

[12] Zitong Lan, Chenhao Zheng, Zhiwei Zheng, and Mingmin Zhao. Acoustic volume rendering for neural impulse response fields. *Advances in Neural Information Processing Systems*, 37:44600–44623, 2024.

[13] Andrew Luo, Yilun Du, Michael Tarr, Josh Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *Advances in Neural Information Processing Systems*, 35:3165–3177, 2022.

[14] Yanjie Zhu, Yuanyuan Liu, Yihang Zhang, and Dong Liang. Implicit neural representation for medical image reconstruction. *Physics in Medicine & Biology*, 70(12):12TR01, 2025.

[15] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 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.

[16] Zahra Kadkhodaie and Eero P Simoncelli. Solving linear inverse problems using the prior implicit in a denoiser. *arXiv preprint arXiv:2007.13640*, 2020.

[17] Xuan Li, Yi-Ling Qiao, Peter Yichen Chen, Krishna Murthy Jatavallabhula, Ming Lin, Chenfanfu Jiang, and Chuang Gan. Pac-nerf: Physics augmented continuum neural radiance fields for geometry-agnostic system identification. *arXiv preprint arXiv:2303.05512*, 2023.

[18] David Borts, Erich Liang, Tim Broedermann, Andrea Ramazzina, Stefanie Walz, Edoardo Palladin, Jipeng Sun, David Brueggemann, Christos Sakaridis, Luc Van Gool, et al. Radar fields: Frequency-space neural scene representations for fmcw radar. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024.

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

[20] Tamir Shor, Ethan Fetaya, Chaim Baskin, and Alex Bronstein. Adversarial robustness in parameter-space classifiers. *arXiv preprint arXiv:2502.20314*, 2025.

[21] Gizem Yüce, Guillermo Ortiz-Jiménez, Beril Besbinar, and Pascal Frossard. A structured dictionary perspective on implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19228–19238, 2022.

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

[23] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[24] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2713–2722, 2024.

[25] Hao Zhu, Zhen Liu, Qi Zhang, Jingde Fu, Weibing Deng, Zhan Ma, Yanwen Guo, and Xun Cao. Finer++: building a family of variable-periodic functions for activating implicit neural representation. *arXiv preprint arXiv:2407.19434*, 2024.

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

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

[28] Alireza Morsali, MohammadJavad Vaez, Mohammad Hossein Soltani, Amirhossein Kazerouni, Babak Taati, and Morteza Mohammad-Noori. A unified theory of sinusoidal activation families for implicit neural representations. *Transactions on Machine Learning Research*, 2026. URL `https://openreview.net/forum?id=ZDmBPYptbL`.

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

[30] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G Baraniuk. Wire: Wavelet implicit neural representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18507–18516, 2023.

[31] Hemanth Saratchandran, Sameera Ramasinghe, Violetta Shevchenko, Alexander Long, and Simon Lucey. A sampling theory perspective on activations for implicit neural representations. *arXiv preprint arXiv:2402.05427*, 2024. URL `https://arxiv.org/abs/2402.05427`.

[32] Taesun Yeom, Sangyoon Lee, and Jaeho Lee. Fast training of sinusoidal neural fields via scaling initialization. In *International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=Sr5XaZzirA`.

[33] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Sape: Spatially-adaptive progressive encoding for neural optimization. *Advances in Neural Information Processing Systems*, 34: 8820–8832, 2021.

[34] 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.

[35] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *International conference on learning representations*, 2021.

[36] Ali Mehrabian, Parsa Mojarad Adi, Moein Heidari, and Ilker Hacihaliloglu. Implicit neural representations with fourier kolmogorov-arnold networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025. URL `https://arxiv.org/abs/2409.09323`.

[37] Reza Rezaeian, Moein Heidari, Reza Azad, Dorit Merhof, Hamid Soltanian-Zadeh, and Ilker Hacihaliloglu. SL2A-INR: Single-layer learnable activation for implicit neural representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 26065–26074, 2025. URL `https://openaccess.thecvf.com/content/ICCV2025/html/Rezaeian_SL2A-INR_Single-Layer_Learnable_Activation_for_Implicit_Neural_Representation_ICCV_2025_paper.html`.

[38] Dhananjaya Jayasundara, Heng Zhao, Demetrio Labate, and Vishal M. Patel. MIRE: Matched implicit neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8279–8288, 2025. URL `https://openaccess.thecvf.com/content/CVPR2025/html/Jayasundara_MIRE_Matched_Implicit_Neural_Representations_CVPR_2025_paper.html`.

[39] Alireza Afzal Aghaei. fKAN: Fractional kolmogorov-arnold networks with trainable jacobi basis functions. *arXiv preprint arXiv:2406.07456*, 2024. URL `https://arxiv.org/abs/2406.07456`.

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

[41] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *International conference on machine learning*, pages 1068–1077. PMLR, 2017.

[42] Eastman Kodak Company. Kodak lossless true color image suite. `http://r0k.us/graphics/kodak/`, 1993. Accessed: 2026-04-29.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

# A Implementation Details

In this section we elaborate on implementation details across main experiments from Section 4. All experiments are conducted on a single NVIDIA RTX A6000 GPU unless otherwise stated.

## A.1 Audio Data

**Datasets and preprocessing.** We evaluate audio INR fitting on fixed subsets of LibriSpeech and NSynth. For LibriSpeech, we use the fixed manifest `pilot40_manifest.csv`, containing 40 one-second clips. For NSynth, we use the fixed manifest `pilot10_convergence_manifest.csv`, containing 10 one-second clips. All reported quantitative curves use these fixed clip sets. The manifests are held fixed across methods and hyperparameter sweeps. Audio is resampled to 48 kHz and each training signal is represented by $N = 48{,}000$ samples. We convert each clip to a single waveform channel and use the waveform amplitudes on the decoded floating-point audio scale. We do not apply per-method renormalization. Coordinates are normalized to $t \in [-1, 1]$, and each model learns a coordinate-to-amplitude function $f_\theta(t) \in \mathbb{R}$.

**Fourier features.** For coordinate input $t \in \mathbb{R}$, we use Gaussian Fourier features

$$\gamma_\sigma(t) = \left[t,\, \sin(2\pi t B^\top),\, \cos(2\pi t B^\top)\right], \qquad B_i \sim \mathcal{N}(0, \sigma^2),$$

where $B \in \mathbb{R}^{M\times 1}$ and $M$ is the number of Fourier-feature bands. Unless otherwise stated, $M = 16$. The same Fourier-feature width and the same sampled matrix $B$ are used across methods for a given seed and $\sigma$. The setting $\sigma = 0$, equivalently $M = 0$, denotes raw coordinates without Fourier features.

Table 2: Audio fitting hyperparameters. FF16 denotes $M = 16$ Gaussian Fourier-feature bands. The selected $\sigma$ values are those used for the main audio fitting results. Full $\sigma$-sweep curves are reported separately.

| Quantity | Setting |
|---|---|
| Datasets | LibriSpeech fixed 40-clip manifest. NSynth fixed 10-clip manifest |
| Sampling rate | 48 kHz |
| Samples per clip | $N = 48{,}000$ samples, one-second window |
| Channels | Mono waveform fitting |
| Coordinate range | $t \in [-1, 1]$ |
| Output dimension | 1 waveform amplitude |
| Loss | Waveform MSE |
| Optimizer | Adam |
| Training steps | 5,000 |
| Batching | Full grid unless explicitly marked as minibatch. Minibatch runs use 4,096 coordinates |
| Hidden width | 256 |
| Depth | 4 hidden activation layers unless stated otherwise |
| Fourier features | $\gamma_\sigma(t) = [t, \sin(2\pi t B^\top), \cos(2\pi t B^\top)]$ |
| FF width | $M = 16$ bands for FF16 |
| Raw-coordinate setting | $M = 0$, reported as no-FF or $\sigma = 0$ |
| SIREN | $\omega_0 = 30$ |
| FINER | $\omega_0 = 30$, first-layer bias scale $f_{\text{bs}} = 20$ in audio runs |
| LSA | $K = 16$, residual identity included, layer-shared coefficients |
| LSA coefficient init. | $a_k \sim \mathcal{N}(0, 0.01^2)$ |
| Checkpoint selection | Best observed PSNR over the training trajectory |
| Metric | $\text{PSNR} = -10 \log_{10}(\text{MSE})$, dynamic range 1 |
| Curve aggregation | Mean across clips. Shaded regions show standard deviation across clips |

We sweep $\sigma$ and report both the full sweep and the best setting per method. In the main audio results, the selected settings are $\sigma = 100$ for LSA with FF16 and $\sigma = 20$ for FINER with FF16. SIREN is also evaluated under the same matched FF regime when used in compute comparisons.

**Models.** All audio fitting models are coordinate MLPs with scalar output. Unless otherwise stated, models use hidden width 256 and depth 4. SIREN uses sinusoidal activations with $\omega_0 = 30$ for both the first and hidden layers. FINER uses the authors' variable-periodic activation,

$$\phi_{\text{FINER}}(u) = \sin(\omega_0(|u| + 1)u)\,,$$

with $\omega_0 = 30$. We use the FINER-style initialization path used in our experiments: the first-layer bias scale is set by $f_{\text{bs}}$, and the reported audio runs use $f_{\text{bs}} = 20$. Hidden-layer weights and biases follow the implementation's hidden-layer initialization rule. LSA uses the residual harmonic activation from Equation (2),

$$\phi_{\text{LSA}}(u) = u + \sum_{k=1}^{K} a_k \sin(2\pi k u),$$

including the residual identity term exactly as written. The harmonic coefficients $a_k$ are shared within each activation layer, so LSA adds only $K$ scalar parameters per activation layer. The main audio configuration uses $K = 16$. Coefficients are initialized from a zero-mean Gaussian with standard deviation $0.01$.

**Training and model selection.** All methods are trained to minimize waveform mean-squared error over coordinate samples. Unless a minibatch setting is explicitly specified, training uses the full $48{,}000$-sample grid. We use Adam and train for 5,000 optimization steps. Reported PSNR values use the best checkpoint observed along the training trajectory rather than the final step. For all method comparisons, the clip manifests, coordinate grids, Fourier-feature width, and Fourier-feature samples are held fixed across methods for the corresponding seed and $\sigma$.

**Metrics.** For a target waveform $y$ and prediction $\hat{y}$, we compute

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2, \qquad \text{PSNR} = -10\log_{10}(\text{MSE}),$$

using unit dynamic range. Dataset-level values are averages over the fixed clip sets. In Figure 2, each curve reports the mean over clips and the shaded region denotes the standard deviation across clips.

**Continuous approximation.** The continuous-approximation experiment follows the same audio fitting protocol unless stated otherwise, but uses a fixed grid of 16,000 samples, as described in the main text. The seen-grid or on-grid evaluation uses the same coordinate locations used for fitting. The unseen midpoint or inter-sample evaluation constructs a second grid by placing query points halfway between adjacent training-grid coordinates. Thus, if the seen grid is $\{t_i\}_{i=1}^{N}$, the midpoint grid is

$$\tilde{t}_i = \frac{t_i + t_{i+1}}{2}, \qquad i = 1, \ldots, N-1.$$

This measures whether the learned coordinate function interpolates between training samples rather than merely matching the discrete training grid. The same model architecture, optimizer, Fourier-feature construction, and checkpoint-selection rule are used as in fitting. The appendix includes the full continuous-approximation sweep over SIREN, FINER, and LSA, including raw-coordinate and FF16 settings and the corresponding $\sigma$ values.

### A.2 Images and Derivatives

**Kodak data.** We use the Kodak Lossless True Color Image Suite, using the standard `kodim01.png`-`kodim24.png` images. The dataset contains 24 RGB images, each at spatial resolution $768 \times 512$. Images are loaded as 8-bit RGB PNGs and converted to floating point values in $[0, 1]$. For all coordinate-based experiments, pixel locations are represented by continuous coordinates

$$x = (x_1, x_2) \in [-1, 1]^2,$$

obtained by linearly mapping the horizontal and vertical pixel axes to $[-1, 1]$. Direct image-fitting experiments use the full pixel grid as the training domain. Optimization is performed with random minibatches of pixels, while evaluation renders the complete $768 \times 512$ grid. Unless explicitly stated otherwise, direct-fitting train and evaluation grids are therefore the same grid. No held-out pixels are used for this diagnostic setting.

**Image Fitting.** The image-fitting task learns a coordinate-to-RGB function

$$f_\theta : [-1, 1]^2 \to \mathbb{R}^3.$$

For optimization, RGB targets are mapped from $[0, 1]^3$ to $[-1, 1]^3$. Predictions are mapped back to $[0, 1]^3$ before PSNR/SSIM evaluation. All methods in the official direct-fitting comparison are trained in the no-uplifting regime: the input to each network is the raw normalized coordinate $x$, with no Fourier features, positional encoding, or other input lifting.

The matched-backbone comparison uses width 256, four nonlinear layers, and a linear RGB output layer. For SIREN we use the official sinusoidal architecture with width 256, $\omega_0 = 30$ in the first and hidden layers, and RGB output dimension 3. For FINER we use the official FINER architecture with width 256, $\omega_0 = 30$, RGB output dimension 3, and first-layer bias scale

$$\texttt{first_bias_scale} = \sqrt{2}/2 \approx 0.70710678.$$

For LSA, each hidden unit applies the learned spectral activation

$$\phi(u) = u + \sum_{k=1}^{K} a_k \sin(2\pi k u),$$

with $K = 32$ in the main no-Fourier-feature comparison. The coefficients $a_k$ are initialized from a zero-mean Gaussian with standard deviation $10^{-3}$. We also tested an additive learned DC term on a Kodak subset. It did not improve that diagnostic and is not used in the reported main comparison.

The fixed LSA/FINER/SIREN comparison uses Adam for 5,000 updates, with minibatches of 65,536 pixels, seed 0, and full-grid evaluation every 100 steps. SIREN and FINER use an initial learning rate of $5 \times 10^{-4}$. LSA ordinary weights and biases also use $5 \times 10^{-4}$, while its activation coefficients use $10^{-3}$ via the multiplier $\texttt{act_lr_mult} = 2$. All three models follow the same learning-rate multiplier $0.1^{\min(s/5000,1)}$ at step $s$, applied to both parameter groups for LSA. LSA has 198,915 ordinary

MLP weights and biases and $4 \times 32 = 128$ activation coefficients, for 199,043 trainable parameters. The LSA calibration candidates and the matched-budget global STAF configuration are specified in Section C.1.

The reported checkpoint maximizes evaluated PSNR on the complete fitting grid. This is in-sample signal fitting with no held-out pixels. Image-level holdouts in Section C.1 concern configuration selection. PSNR is computed on RGB values in $[0, 1]$ as

$$\mathrm{PSNR} = -10 \log_{10} \left( \frac{1}{3HW} \sum_{p,c} \big(\hat{I}_{p,c} - I_{p,c}\big)^2 + 10^{-12} \right).$$

SSIM is computed on RGB images with data range 1. In Figure 6, spatial residuals are mean absolute RGB errors per pixel. Spectral residuals use the channel-averaged signed prediction error $e$, displayed as $\log_{10}(1 + |\mathcal{F}(e)|)$ with a centered two-dimensional FFT. Each residual type uses one 99th-percentile display scale pooled over both images and all three methods.

**Gradient-only / Poisson reconstruction.** For the derivative-supervision experiment, the network is not trained with direct RGB supervision. Instead, it is trained to match the spatial derivative field of the target image. We run this experiment on the Kodak images using the same coordinate domain $x \in [-1, 1]^2$. The main figure uses representative Kodak examples, and the appendix includes the corresponding residual grids for the full image set.

Target gradients are computed independently for each RGB channel using finite differences via `numpy.gradient`. This gives central finite differences in the interior and one-sided finite differences at the image boundary. Let $I \in [0, 1]^{H \times W \times 3}$ be the target image. For each channel $c$, we compute pixel-index derivatives

$$\partial_x I_c, \qquad \partial_y I_c.$$

Because the INR is differentiated with respect to normalized coordinates in $[-1, 1]^2$, the derivative targets are scaled by the coordinate resolution. The model output is interpreted in the $[-1, 1]$ RGB range during training, so the derivative targets are

$$g^x_{p,c} = (W - 1)\, \partial_x I_{p,c}, \qquad g^y_{p,c} = (H - 1)\, \partial_y I_{p,c}.$$

For a sampled coordinate minibatch, gradients of the network output are computed by automatic differentiation with respect to the input coordinates. The loss is

$$\mathcal{L}_{\mathrm{grad}} = \|\partial_x f_\theta(x) - g^x\|_2^2 + \|\partial_y f_\theta(x) - g^y\|_2^2 ,$$

where the norm is averaged over sampled coordinates and RGB channels. Thus, RGB derivatives are supervised separately, rather than after conversion to grayscale.

Derivative-only supervision leaves an additive constant ambiguity. Therefore, for evaluation and visualization, the predicted image is aligned to the ground truth by matching its mean RGB value channel-wise:

$$\hat{I}_{\mathrm{aligned}} = \hat{I} - \mathrm{mean}_p(\hat{I}_p) + \mathrm{mean}_p(I_p).$$

The aligned output is clipped to $[0, 1]$, and PSNR is computed after this mean alignment.

The residual visualizations in the Poisson figures are computed as follows. The gradient-field residual is

$$r_\nabla(p) = \sqrt{\frac{1}{3} \sum_{c=1}^{3} \Big[ \big(\partial_x \hat{I}_{p,c} - \partial_x I_{p,c}\big)^2 + \big(\partial_y \hat{I}_{p,c} - \partial_y I_{p,c}\big)^2 \Big]}.$$

The image residual is the mean absolute RGB error,

$$r_{\mathrm{pix}}(p) = \frac{1}{3} \sum_{c=1}^{3} \left| \hat{I}_{p,c} - I_{p,c} \right|.$$

The spectral residual is computed from the channel-averaged residual

$$e(p) = \frac{1}{3} \sum_{c=1}^{3} \big(\hat{I}_{p,c} - I_{p,c}\big),$$

using

$$r_{\mathrm{fft}} = \log_{10} \left(1 + |\mathcal{F}(e)|\right),$$

where $\mathcal{F}$ is the centered two-dimensional FFT. For display, each residual type is normalized by a robust 99-th percentile scale computed over the compared methods within the same figure.

### A.3 Novel View Synthesis

We evaluate novel view synthesis on the Blender synthetic dataset [3], using the eight scenes *Chair*, *Drums*, *Ficus*, *Hotdog*, *Lego*, *Materials*, *Mic*, and *Ship*. We follow the reduced-view protocol used by FINER [24]: from the standard Blender training split, we retain every fourth training image and evaluate on the held-out test views. Images are rendered and evaluated at downscale factor $4$, corresponding to $200 \times 200$ resolution for the original $800 \times 800$ Blender images. All methods are implemented in the same Instant-NGP-style differentiable volume-rendering codebase, where an MLP predicts density and view-dependent color from 3D position and viewing direction, and pixel colors are obtained by alpha compositing along sampled camera rays.

For all methods we use the same NeRF backbone: four density layers, four color layers, hidden width $182$, geometry-feature dimension $182$, and $512$ samples per ray. Each optimization step samples $1024$ rays from the retained training views, with gradient accumulation over four steps, giving an effective ray batch of $4096$. Models are trained from scratch for $150{,}000$ optimization steps using Adam, and reported values use the final checkpoint after this fixed budget. Only the nonlinear activation family is changed across methods.

SIREN [2] uses sinusoidal activations with first-layer frequency $\omega_0 = 30$ and hidden-layer frequency $\omega_0 = 1$. FINER [24] uses the variable-periodic activation with the same frequency settings and bias scale $0.577$. LSA replaces the fixed nonlinearities in the same NeRF MLP with $K = 24$ harmonic components, layer-wise coefficients shared across neurons, and activation-coefficient initialization standard deviation $5 \times 10^{-3}$. We use learning rate $3 \times 10^{-4}$ for LSA and $1 \times 10^{-4}$ for SIREN and FINER, based on the best stable settings for the shared reduced-view protocol. No scene-specific LSA hyperparameter tuning is used.

We report PSNR, SSIM, and LPIPS on the held-out test views. PSNR is computed from mean squared RGB error, SSIM from the evaluation code's standard structural-similarity implementation, and LPIPS with the AlexNet backbone. Qualitative residual maps are computed from saved evaluation tensors as the per-pixel mean absolute RGB error.

### A.4 Neural acoustic fields

We evaluate LSA in a neural acoustic field setting based on the Neural Acoustic Fields (NAF) formulation [13]. Our experiments use the NAF codebase with minimal architectural modification: the baseline model, data loading, spatial feature grid, coordinate embeddings, training protocol, and evaluation pipeline are kept unchanged, except that the hidden activation in the NAF decoder is replaced by LSA. We evaluate on the benchmark SoundSpaces/Replica scenes `frl_apartment_2`, `frl_apartment_4`, `room_4`, and `office_2`. For each scene, the train/test split follows the NAF protocol, where impulse responses are split by emitter-listener spatial queries and held-out queries are used for evaluation. Each query consists of a listener position, an emitter position, listener orientation, and binaural channel index. Listener and emitter coordinates are represented in the horizontal plane and are combined with sinusoidal embeddings of the time-frequency coordinates. As in NAF, the model additionally uses a learned local geometric feature grid queried at the listener and emitter positions. The target room impulse responses are represented in the NAF time-frequency output format. The model predicts the normalized spectrogram-domain quantities used by the original NAF implementation, and evaluation converts predictions and ground truth to the corresponding waveform or spectral representations as required by each metric. All experiments use the same RIR sample rate, impulse-response length, and normalization files as the NAF benchmark preprocessing.

**Architecture.** The baseline is the shared-grid NAF architecture: listener and emitter positions query a learned spatial feature grid, these features are concatenated with coordinate, orientation, ear, time, and frequency embeddings, and an MLP decoder predicts the NAF output at the queried time-frequency coordinate. Let $u$ denote a scalar hidden pre-activation in this decoder. In the LSA variant, we replace the fixed baseline activation by

$$\phi(u) = u + \sum_{k=1}^{K} a_k \sin(2\pi k u), \qquad K = 8. \tag{6}$$

The LSA coefficients are shared across channels within each layer and are initialized from a zero-mean Gaussian with standard deviation $10^{-2}$. All other NAF modules, including the learned spatial feature

grid, coordinate embeddings, orientation and binaural embeddings, output head, and normalization procedure, are unchanged. Thus, the comparison isolates the effect of replacing the hidden nonlinear operator while preserving the NAF parameterization and acoustic-field conditioning.

**Training.** We use the original NAF training objective, namely the spectrogram-domain reconstruction loss used by the baseline implementation. Models are trained independently per scene. Unless otherwise stated, all NAF and LSA-NAF runs use Adam with learning rate $5 \times 10^{-4}$, batch size $10$, $256$ sampled time-frequency coordinates per impulse response, hidden width $512$, and $50$ epochs. For each scene and method, we report the final checkpoint at epoch $50$. The same train/test split, sampled query distribution, preprocessing statistics, and optimization budget are used for the baseline and LSA variants. All random seeds are fixed within each paired baseline/LSA comparison whenever the codebase exposes seed control. Otherwise, both methods are run with the same data split and deterministic preprocessing artifacts.

**Metrics.** Metrics are computed on held-out spatial queries and then averaged over spatial positions and scenes. Let $\hat{x} \in \mathbb{R}^{2\times T}$ and $x \in \mathbb{R}^{2\times T}$ denote the predicted and ground-truth binaural impulse responses after conversion to the waveform domain, with two channels corresponding to the left and right ears.

Waveform MSE is

$$\mathrm{MSE}(\hat{x}, x) = \frac{1}{2T}\|\hat{x} - x\|_2^2. \tag{7}$$

PSNR is computed from waveform MSE as

$$\mathrm{PSNR} = -10 \log_{10}\left(\max(\mathrm{MSE}, 10^{-12})\right), \tag{8}$$

assuming unit-normalized waveform scale. MAE is the mean absolute waveform error,

$$\mathrm{MAE}(\hat{x}, x) = \frac{1}{2T}\|\hat{x} - x\|_1. \tag{9}$$

Correlation is the normalized inner product between flattened predicted and ground-truth binaural waveforms,

$$\mathrm{Corr}(\hat{x}, x) = \frac{\langle \mathrm{vec}(\hat{x}), \mathrm{vec}(x)\rangle}{\|\mathrm{vec}(\hat{x})\|_2 \|\mathrm{vec}(x)\|_2 + \epsilon}. \tag{10}$$

For frequency-domain metrics, we compute the real FFT independently for each binaural channel. Let $P = |\mathrm{rFFT}(\hat{x})|$ and $G = |\mathrm{rFFT}(x)|$. Spectral MSE and spectral L1 are

$$\mathrm{SpectralMSE} = \frac{1}{|P|}\|P - G\|_2^2, \qquad \mathrm{SpectralL1} = \frac{1}{|P|}\|P - G\|_1. \tag{11}$$

Log spectral L1 is

$$\mathrm{LogSpectralL1} = \frac{1}{|P|}\left\|\log(P + \epsilon) - \log(G + \epsilon)\right\|_1. \tag{12}$$

For the multi-resolution STFT metric, we compute log-magnitude STFT errors at FFT sizes $\{256, 512, 1024, 2048\}$, using a Hann window and hop size $n_{\mathrm{fft}}/4$ for each resolution, and average the resulting log-L1 errors:

$$\mathrm{MRSTFT\text{-}LogL1} = \frac{1}{4}\sum_{n\in\{256,512,1024,2048\}} \left\|\log(|\mathrm{STFT}_n(\hat{x})| + \epsilon) - \log(|\mathrm{STFT}_n(x)| + \epsilon)\right\|_1. \tag{13}$$

TDOA error measures binaural timing-cue error. For each waveform, we estimate the interaural time delay by computing the full cross-correlation between the left and right channels and taking the lag of the maximum response:

$$\tau(x) = \arg\max_{\ell} \sum_t x_L(t) x_R(t - \ell). \tag{14}$$

The TDOA error is

$$\mathrm{TDOA}(\hat{x}, x) = |\tau(\hat{x}) - \tau(x)|, \tag{15}$$

reported in samples. Lower is better for MSE, MAE, spectral MSE, spectral L1, log spectral L1, MRSTFT log-L1, and TDOA. Higher is better for PSNR and correlation.

We report two aggregate summaries. Macro-scene aggregation first averages a metric over spatial positions within each scene and then averages the resulting scene means, giving each scene equal weight. Micro-position aggregation pools all evaluated spatial positions across scenes and computes a single average. When reporting signed improvements, positive values always indicate that LSA is better: for lower-is-better metrics we use $\mathrm{baseline} - \mathrm{LSA}$, while for higher-is-better metrics we use $\mathrm{LSA} - \mathrm{baseline}$.

**Spatial improvement maps.** Figure 7 visualizes spatially resolved improvements for `frl_apartment_2`. Each point corresponds to an evaluated listener/emitter spatial query location projected onto the scene floor plan. For lower-is-better metrics, the relative improvement at spatial position $p$ is

$$\Delta_{\%}(p) = 100 \cdot \frac{m_{\mathrm{baseline}}(p) - m_{\mathrm{LSA}}(p)}{\max(|m_{\mathrm{baseline}}(p)|, \epsilon)}, \tag{16}$$

so positive values indicate lower error under LSA. For higher-is-better metrics, the numerator is reversed. Metrics are first averaged over all held-out examples associated with the same spatial position and then visualized on the floor-plan grid. For readability, color ranges are clipped symmetrically using a robust percentile of the absolute improvement values for each metric, with a separate colorbar per metric.

### A.5 NTK Energy

In this section we include implementation details for the experiment from Figure 1 Let $f_\theta(x)$ denote the INR and $\{x_i\}_{i=1}^N$ a set of sampled coordinates. The empirical neural tangent kernel is defined as

$$\kappa_{ij} = \langle \nabla_\theta f_\theta(x_i), \nabla_\theta f_\theta(x_j) \rangle \,. \tag{17}$$

Let $\kappa = Q\Lambda Q^\top$ be its eigendecomposition, with orthonormal eigenvectors $\{q_i\}$. Given a target signal $y \in \mathbb{R}^N$ centered to zero mean, we project it onto the NTK eigenbasis as

$$\alpha_i = \langle q_i, y \rangle. \tag{18}$$

We measure the normalized cumulative target energy captured by the top $m$ modes as

$$E(m) = \frac{\sum_{i=1}^m \alpha_i^2}{\|y\|^2}. \tag{19}$$

All methods use the same coordinate subset and target preprocessing. All trainable parameters are included in the Jacobian, including LSA activation coefficients.

### A.6 Local Optimization

The harmonic coefficients expose a structured family of learning directions. We characterize their contribution to squared-loss descent at a fixed network state, then examine the amplitude parameterization near identity initialization. The results concern the residual on the coordinates used to form the Jacobian.

#### A.6.1 Tangent Features and Descent

Let $X = (x_1, \ldots, x_N)$ be fixed input coordinates and let $f_{w,a}(X) \in \mathbb{R}^{Nd}$ stack the network outputs, where $d$ is the output dimension. The vector $w$ contains all ordinary weights and biases, and $a$ contains the coefficients $a_k^{(\ell)}$, for $\ell = 1, \ldots, L-1$ and $k = 1, \ldots, K$. Write $r = f_{w,a}(X) - y$ and $\mathcal{L}(w, a) = \frac{1}{2}\|r\|_2^2$. A mean-squared-error normalization can be absorbed into the group learning rates.

For one input, let $B_\ell(x) = \partial f(x)/\partial z^{(\ell)}(x)$ be the downstream Jacobian from layer $\ell$ to the output. Since $a_k^{(\ell)}$ is shared across channels, its output tangent feature is

$$\psi_{\ell k}(x) = B_\ell(x) \sin\Big(2\pi k u^{(\ell)}(x)\Big), \tag{20}$$

where the sine acts elementwise and the matrix multiplication sums the channel contributions. Stacking these vectors over $X$ gives the corresponding column of $\Psi = \partial f(X)/\partial a$. Let $J_w = \partial f(X)/\partial w$.

Under full-batch gradient flow with fixed positive rates $\eta_w$ and $\eta_a$,

$$\dot{w} = -\eta_w J_w^\top r, \qquad \dot{a} = -\eta_a \Psi^\top r, \qquad \dot{r} = -\left(\eta_w J_w J_w^\top + \eta_a \Psi \Psi^\top\right) r. \tag{21}$$

The chain rule yields

$$-\dot{\mathcal{L}} = \eta_w \|J_w^\top r\|_2^2 + \eta_a \|\Psi^\top r\|_2^2 = \eta_w \|J_w^\top r\|_2^2 + \eta_a \sum_{\ell,k} \langle r, \psi_{\ell k}\rangle^2. \tag{22}$$

This identity is exact at finite width, including when the tangent features change during training. Each coefficient follows the correlation of the residual with its downstream-propagated harmonic feature. At a common network state, the additional instantaneous descent contribution from updating $a$ is exactly $\eta_a \|\Psi^\top r\|_2^2$. It is positive when the residual has a nonzero projection onto $\mathrm{col}(\Psi)$.

In the fixed-NTK approximation, evaluate both Jacobians at the shared initialization $(w_0, a_0)$ and keep them fixed. The model with frozen coefficients trains $w$ while holding $a = a_0$. Its effective kernel and that of the jointly trained model satisfy

$$\kappa_{\mathrm{frozen},0} = \eta_w J_{w,0} J_{w,0}^\top, \qquad \kappa_{\mathrm{LSA},0} = \kappa_{\mathrm{frozen},0} + \eta_a \Psi_0 \Psi_0^\top. \tag{23}$$

The update is positive semidefinite and has rank at most $(L-1)K$, the number of coefficients. For either fixed kernel $\kappa_0$, the residual evolves as $r(t) = \exp(-\kappa_0 t) r_0$. At initialization, the difference in loss-decay rates is $\eta_a \|\Psi_0^\top r_0\|_2^2$. These comparisons hold at a shared state. Subsequent nonlinear trajectories develop their own Jacobians and residuals, so the applicable guarantee is instantaneous descent and its fixed-kernel local approximation. Generalization to unobserved coordinates is evaluated empirically in the audio and neural-field experiments.

The harmonic choice determines the alignment of the added directions. In a simplified local model with orthonormal propagated harmonic features $q_1, \ldots, q_K$, let $r = \sum_k c_k q_k + r_\perp$, where $r_\perp$ is orthogonal to their span. For an equal-weight adapter kernel $\kappa_a = \lambda \sum_k q_k q_k^\top$, its target-weighted contribution is $r^\top \kappa_a r = \lambda \sum_k c_k^2$. This describes how a fixed kernel-trace budget can concentrate on harmonic residual energy. For the actual propagated features, the exact quantity is $\|\Psi^\top r\|_2^2$, which also reflects their scale and correlation. The resulting inductive bias is target-dependent.

### A.6.2 Amplitude Conditioning

Consider one residual sinusoidal term $g(u) = u + A\sin(\omega u + \varphi)$. Its parameter derivatives are

$$\frac{\partial g}{\partial A} = \sin(\omega u + \varphi), \qquad \frac{\partial g}{\partial \omega} = Au\cos(\omega u + \varphi), \qquad \frac{\partial g}{\partial \varphi} = A\cos(\omega u + \varphi). \tag{24}$$

Assume a fixed finite network and evaluation set, bounded pre-activations and downstream Jacobians as $A \to 0$, and fixed frequencies and phases. The chain rule then gives frequency and phase output-Jacobian blocks of order $O(A)$, and their Gram matrices are $O(A^2)$. The amplitude block is $O(1)$, and has a nonzero limit whenever the propagated sine features are nondegenerate. The same argument holds for a finite bank with amplitudes $A_k = \epsilon c_k$ and bounded $c_k$ as $\epsilon \to 0$.

For LSA, $\omega_k = 2\pi k$ and $\varphi_k = 0$. At exact-zero coefficient initialization,

$$\phi(u, 0) = u, \qquad \frac{\partial \phi}{\partial u}(u, 0) = 1, \qquad \frac{\partial \phi}{\partial a_k}(u, 0) = \sin(2\pi k u). \tag{25}$$

The activation and its input derivative preserve the identity, while coefficient learning can access harmonic output directions. This motivates amplitude learning under an identity residual parameterization. STAF also learns amplitudes, and its full parameterization and initialization are evaluated separately in Section C.

### A.6.3 Controlled Validation

We validated the finite-width identity using six Kodak images and three network initializations per image. The model has four coefficient-bearing layers with $K = 32$, giving a coefficient-kernel

rank bound of 128. The learned and frozen variants begin from the same state. For a common outer gradient-descent step size $\alpha$, define the one-step loss difference normalized by its first-order prediction as

$$R(\alpha) = \frac{\mathcal{L}_{\text{frozen}}(\alpha) - \mathcal{L}_{\text{learned}}(\alpha)}{\alpha \eta_a \|\Psi_0^\top r_0\|_2^2}. \tag{26}$$

Here $\mathcal{L}_{\text{learned}}(\alpha)$ follows one joint update from the initial state, and $\mathcal{L}_{\text{frozen}}(\alpha)$ follows the corresponding ordinary-weight update only. Differentiability gives $R(\alpha) \to 1$ as $\alpha \to 0$ whenever the denominator's coefficient is positive. With $\alpha_0 = 0.1/\lambda_{\max}(\kappa_{\text{LSA},0})$ and equal group rates, the measured means are shown in Table 3. The smallest tested step has a positive difference in all 18 image-seed pairs. Exact-zero initialization reaches a mean full-image PSNR within 0.044 dB of small-random initialization on the six-image comparison.

Table 3: One-step validation on six Kodak images and three initializations per image. The normalization in Equation (26) tests the local descent prediction.

| $\alpha/\alpha_0$ | 0.10 | 0.03 | 0.01 |
|---|---|---|---|
| Mean $R(\alpha)$ | 0.806 | 0.942 | 0.981 |

For amplitude conditioning, we evaluated final-output Jacobians of a scalar-input, scalar-output network with width 32, three residual sinusoidal layers, and $K = 8$. We used 512 uniformly spaced coordinates on $[0, 1]$, three ordinary-weight seeds, and float64 Jacobian calculations. Ordinary weights were fixed across the amplitude sweep. With $A_k = \epsilon$, $\omega_k = 2\pi k$, and zero phases, log-log fits over $\epsilon \in [10^{-6}, 10^{-2}]$ gave squared frequency-Jacobian exponents from 2.18 to 2.27 and squared phase-Jacobian exponents near 2.00. Amplitude exponents remained near zero, and the output perturbation scaled approximately linearly with $\epsilon$. At $\epsilon = 0$, the amplitude block remained nonzero for every tested seed.

We also compared amplitude adapters with the same coefficient count and matched kernel trace in this network. All adapters start with zero amplitudes and identical ordinary parameters. The harmonic adapter uses $\sin(2\pi k u)$. The random-frequency adapter uses $\sin(2\pi \nu_k u + \varphi_k)$ with $\nu_k \sim \mathcal{U}[0.5, 8.5]$ and $\varphi_k \sim \mathcal{U}[0, 2\pi]$. The jittered adapter uses $\sin(2\pi(k + \delta_k)u)$ with $\delta_k \sim \mathcal{U}[-0.4, 0.4]$. Five dictionary seeds are evaluated for each stochastic family. For each ordinary-weight seed, coefficient kernels are rescaled to the harmonic kernel's trace. Let $\widetilde{\kappa}_a$ denote this trace-matched kernel and define $D_a = r_0^\top \widetilde{\kappa}_a r_0$.

On the zero-mean, unit-RMS edge target $y(x) \propto \tanh(12 \sin(2\pi\, 3x))$, the harmonic adapter attains $D_a = 0.0611$. The five random dictionaries span $0.00020$ to $0.00481$, and the mean jittered result is $0.0616$. The same ordering holds for the tested spectral-stack target. This comparison associates the observed gain with coordinated harmonic structure under a controlled kernel scale. The local benefit remains determined by the target-weighted alignment of the propagated features.

# B Additional Results

In this section we bring additional results for the experiments discussed in Section 4.

## B.1 Audio Data

### B.1.1 Audio Signal Fitting

Figure 8 shows representative waveform fits and residuals. On LibriSpeech, learned spectral activations nearly overlap the target waveforms in the shown windows, whereas FINER leaves visible errors around transient regions and SIREN often misses amplitude and fine waveform structure. On NSynth, learned spectral activations preserve both periodic structure and decaying envelopes, while FINER captures the coarse oscillatory pattern but leaves larger residuals, and SIREN shows the largest mismatch.

These qualitative differences match the operator-level motivation in Figure 1a. Speech transients and musical timbre are not merely high-frequency signals. They require coordinated harmonic structure whose amplitudes and envelopes vary over time. Frequency warping alone can expose or

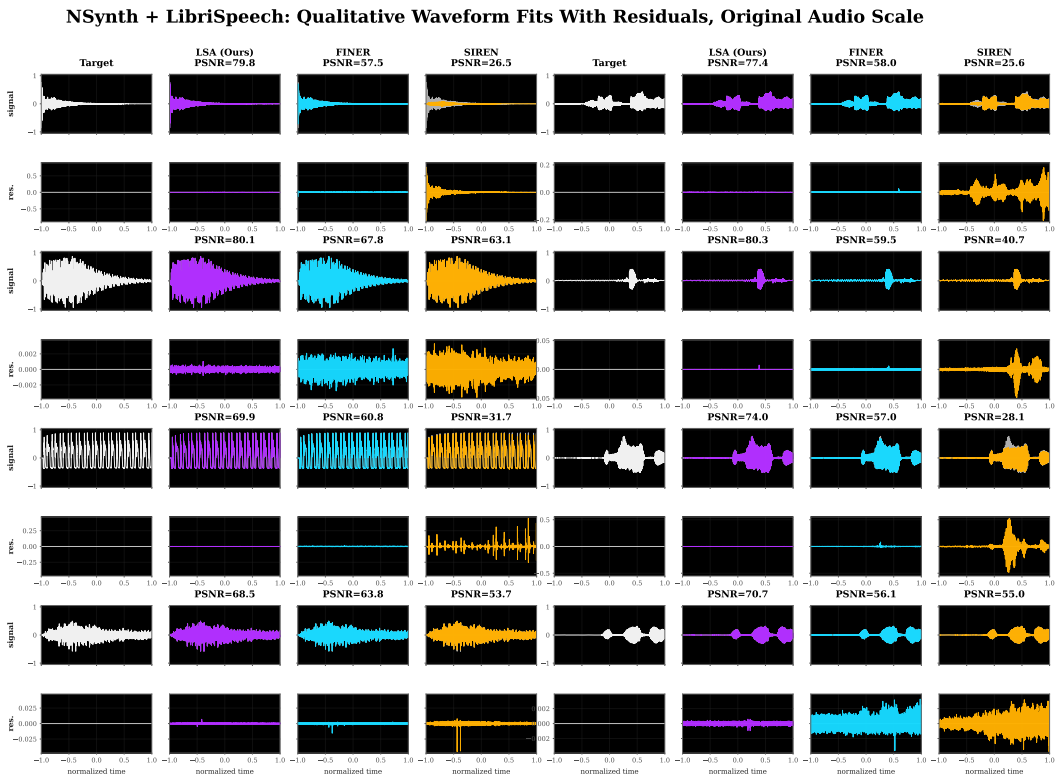



Figure 8: Qualitative waveform fits on LibriSpeech and NSynth. Each column compares the target with a reconstruction and its residual on the original audio scale. Learned spectral activations closely match transient and harmonic waveform structure, leaving substantially smaller residuals than FINER and SIREN.

shift oscillatory components, but it does not directly parameterize their composition. Learned spectral activations, through Equation (2), provide this missing degree of freedom by learning the harmonic mixture of the hidden nonlinear response.

### B.1.2 Continuous Approximation

For the main audio fitting experiments, one-second clips are represented at 48 kHz, yielding $48{,}000$ samples per clip. The continuous-approximation experiment is separate: it uses a fixed grid of $16{,}000$ training samples and evaluates both on-grid and at unseen midpoints between adjacent grid points.

## B.2 Image Fitting

**Additional convergence diagnostics.** Figure 10 shows PSNR trajectories for all Kodak images under the fixed configurations and 5,000-update budget in Section 4.2. At steps 500, 1,000, 2,000, 3,000, 4,000, and 5,000, LSA leads FINER on 5, 16, 21, 22, 21, and 21 of 24 images, respectively. It leads SIREN on 23/24 images at step 500 and all 24 at the remaining listed steps. Thus LSA separates from SIREN early, while its advantage over FINER emerges progressively. Early-stage behavior varies across images.

## B.3 Neural Fields

**NeRFs.** The qualitative comparison in Figure 11 supports the interpretation from Section 4.3. On Chair and Ficus, all methods recover the coarse object appearance, but the residual maps show that SIREN and FINER retain more structured errors around thin geometry, object boundaries, and textured regions. LSA reduces these residuals while preserving comparable global appearance. This aligns with the mechanism in Figure 1a: neural rendering does not only require access to high

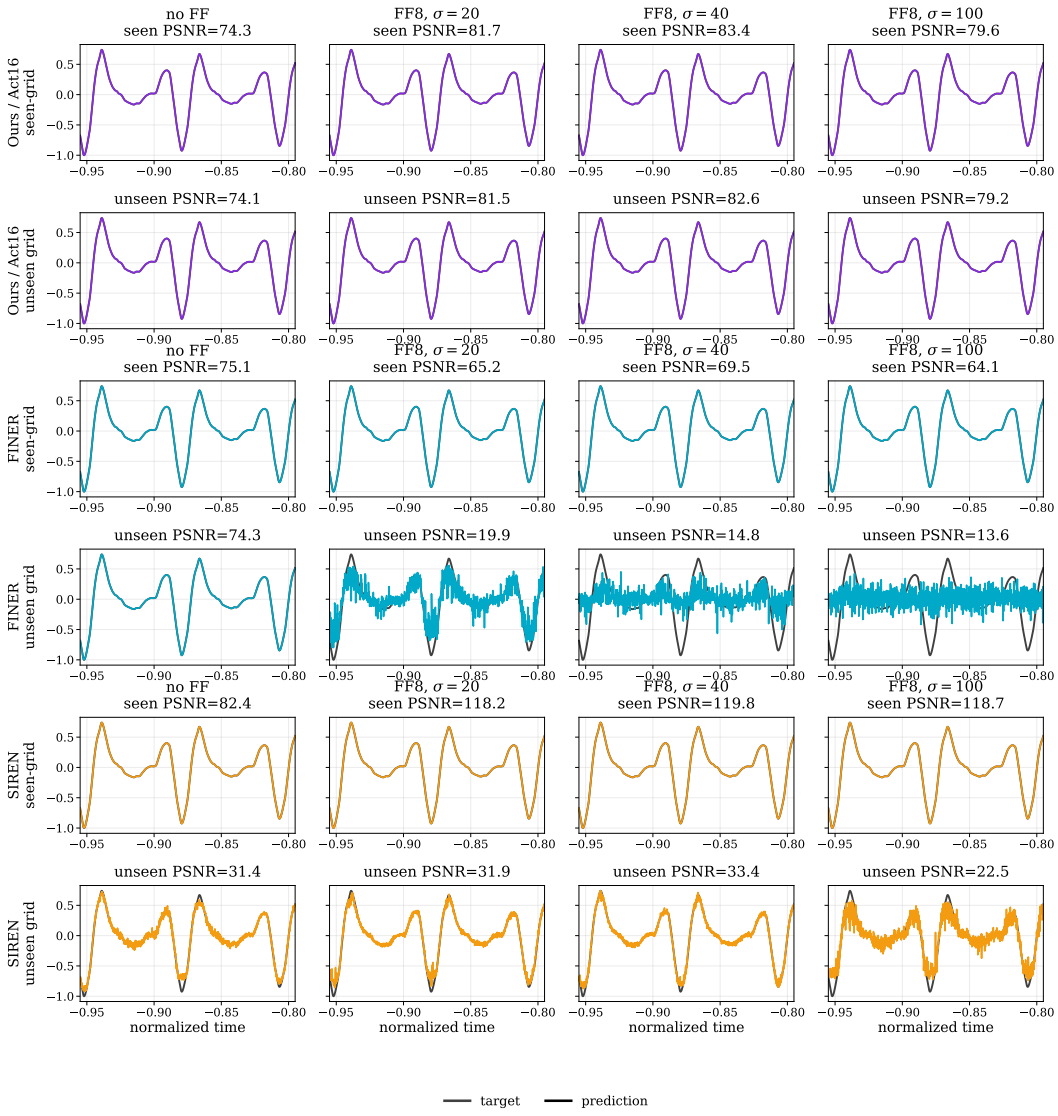



Figure 9: Full NSynth on-grid vs. off-grid diagnostic across bandwidth settings.

frequencies, but efficient composition of localized appearance and geometric detail within the internal representation.

### B.4 Neural-Acoustic Fields

We evaluate on benchmark scenes used in NAF [13], including `frl_apartment_2`, `frl_apartment_4`, `room_4`, and `office_2`. Table 4 reports aggregate results using two complementary summaries: macro-scene averaging, which weights each scene equally, and micro-position averaging, which aggregates over all evaluated spatial positions.

LSA improves every reported metric under both aggregation schemes. The largest gains appear in metrics that are sensitive to the time-frequency structure of the impulse response: log spectral L1 improves by $44.90\%$ under macro-scene aggregation and $50.84\%$ under micro-position aggregation, while MRSTFT log-L1 improves by $45.97\%$ and $51.54\%$, respectively. The improvement in TDOA

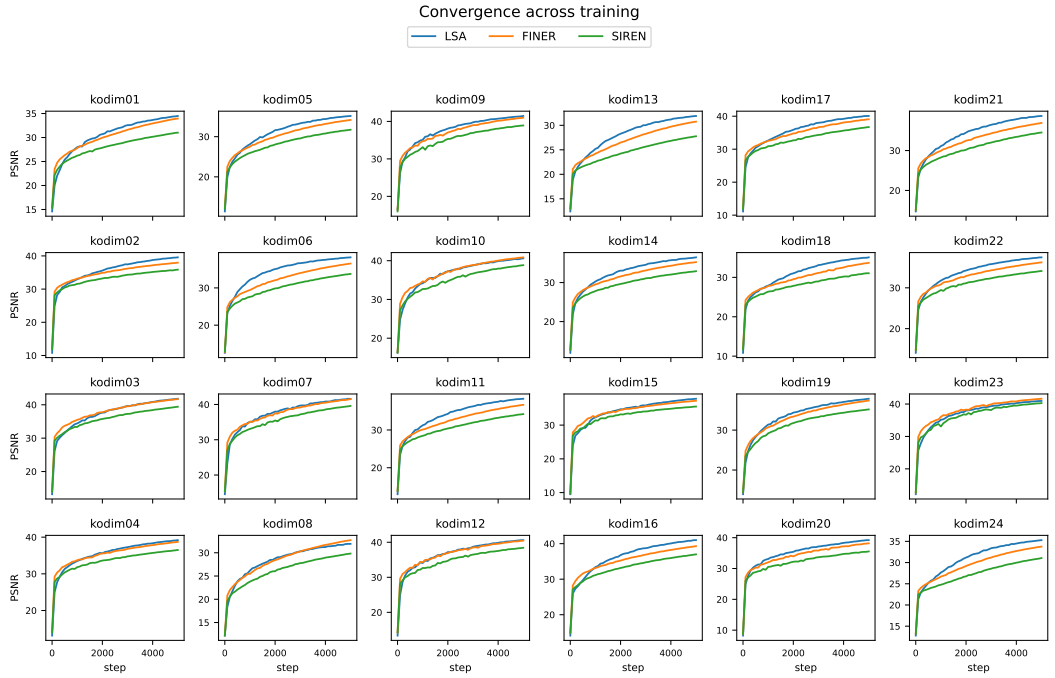



Figure 10: Convergence curves for all 24 Kodak images under fixed LSA, FINER, and SIREN configurations and matched 5,000-update budgets. LSA separates from SIREN early, while its advantage over FINER emerges progressively. Early-stage behavior varies across images.

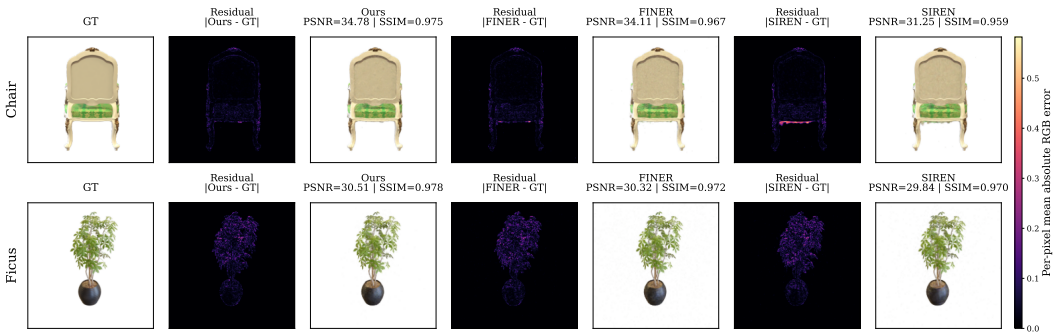



Figure 11: Qualitative novel view synthesis results. We show target views, reconstructions, and absolute RGB residuals for representative scenes. Learned spectral activations reduce structured residuals around fine object boundaries and high-detail regions compared to SIREN and FINER.

is even larger, reaching $67.55\%$ and $72.41\%$, indicating better recovery of spatial timing cues. Waveform-domain metrics also improve consistently, with MSE reduced by approximately $9$-$10\%$ and correlation increasing by roughly $40$-$43\%$.

**Additional Spatial Improvement Maps.** We bring results similar to Figure 7, conclusions from Section 4.3 stably reproduce across different scenes.

### B.5 Additional Use-cases

While this paper focuses on the application of learned spectral activations for INRs, in this section we display initial evidence for the potential of our method beyond this scope. We evaluate positional encodings (PEs) in a sparse-to-dense reconstruction task, where a Transformer predicts the full image from a small set of observed pixels and their coordinates. Unlike dense INR fitting, this setting requires reasoning about spatial relationships between observed and query locations, making

Table 4: Neural acoustic field results. We replace the activation in the NAF model with LSA ($K = 8$) while keeping the architecture and training protocol fixed. Improvements are signed so that positive values indicate better performance.

| | Macro-scene aggregation | | | Micro-position aggregation | | |
|---|---|---|---|---|---|---|
| Metric | Baseline | LSA | Improvement | Baseline | LSA | Improvement |
| PSNR from MSE ↑ | 11.417 | 11.846 | +0.429 / +3.76% | 11.550 | 11.993 | +0.443 / +3.84% |
| MSE ↓ | 0.07238 | 0.06572 | +0.0067 / +9.20% | 0.07 | 0.063 | +0.0068 / +9.71% |
| Log Spectral L1 ↓ | 4.053 | 2.233 | +1.820 / +44.90% | 3.981 | 1.957 | +2.024 / +50.84% |
| MRSTFT Log-L1 ↓ | 4.008 | 2.165 | +1.842 / +45.97% | 3.925 | 1.902 | +2.023 / +51.54% |
| TDOA ↓ | 122.406 | 39.727 | +82.679 / +67.55% | 114.675 | 31.634 | +83.041 / +72.41% |
| Spectral L1 ↓ | 25.558 | 22.061 | +3.497 / +13.68% | 25.252 | 21.173 | +4.079 / +16.15% |
| Spectral MSE ↓ | 975.873 | 770.363 | +205.510 / +21.06% | 954.076 | 719.924 | +234.152 / +24.54% |
| MAE ↓ | 0.2147 | 0.2029 | +0.0118 / +5.49% | 0.2141 | 0.2008 | +0.0134 / +6.24% |
| Corr ↑ | 0.2740 | 0.3823 | +0.1083 / +39.52% | 0.2815 | 0.4039 | +0.1224 / +43.47% |

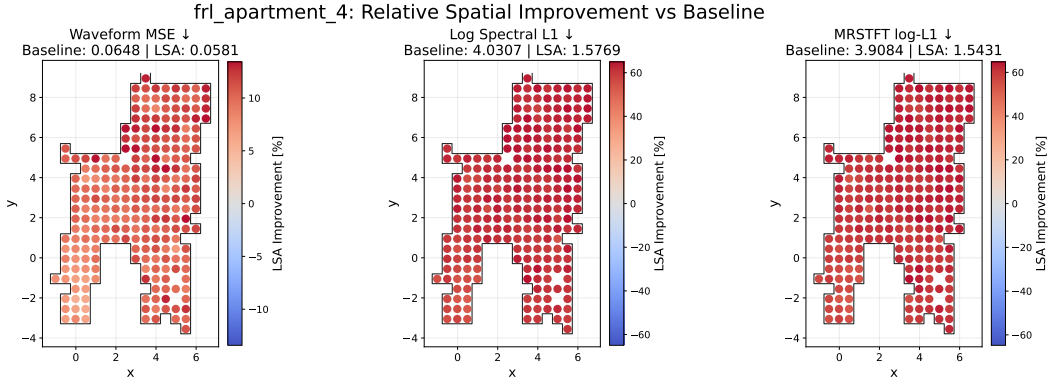



Figure 12: Spatial improvement of LSA over the NAF baseline. frl apartment scene 4

the choice of PE critical. All models share the same architecture and training setup, only the PE differs - we compare between not using any form of PE, the seminal sinusoidal PE from [43], plain coordinate-space learning of PE, and learned PE based on our proposed spectral formulation (see Appendix). To ablate the amount of signal carried by the coordinate identity rather than its RGB values, we also compare to a variant of our learned PE that does not receive the raw pixel data as input, but only coordinate location. For all regimes we report PSNR high-gradient PSNR (*Edge*), computed only on the top 20% of pixels by ground-truth gradient magnitude.

Figure 17 shows that the task is highly PE-sensitive. Removing structure (*No PE*) yields oversmoothed reconstructions, while learned table embeddings perform poorly, indicating that unstructured learned PEs are insufficient. Fixed sinusoidal PE improves results but exhibits structured artifacts and elevated residuals near edges and fine details.

Learned spectral PE consistently produces sharper reconstructions and lower errors, particularly in high-frequency regions. Notably, the *PE-only* variant-where spatial information is provided exclusively through the learned spectral encoding-performs best, showing that the learned representation itself forms an effective coordinate system.

From the harmonic dictionary perspective (Section 3), fixed PEs impose a predefined spectral basis, while learned spectral PE adapts this basis to the task. The improvement observed here indicates that learning the spectral structure of positional encodings yields a more suitable representation of spatial relationships, rather than merely adding features.

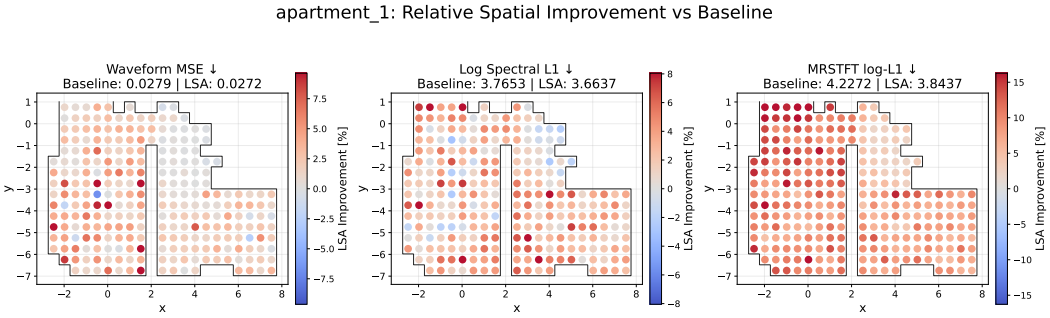



Figure 13: Spatial improvement of LSA over the NAF baseline. Apartment Scene 1

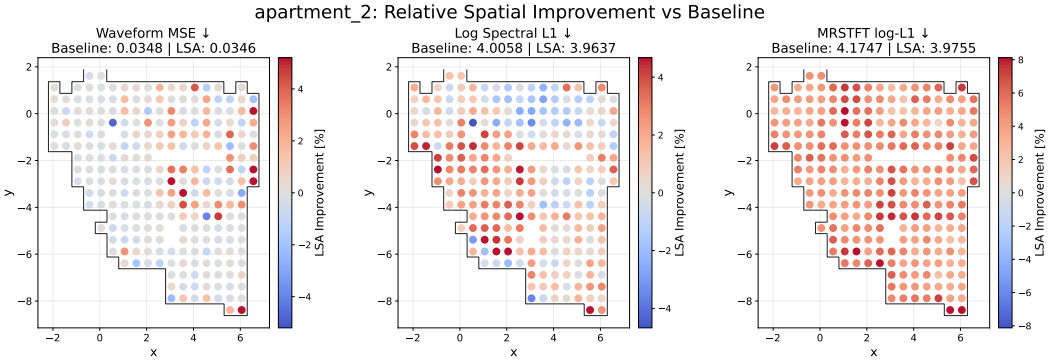



Figure 14: Spatial improvement of LSA over the NAF baseline. Apartment Scene 2.

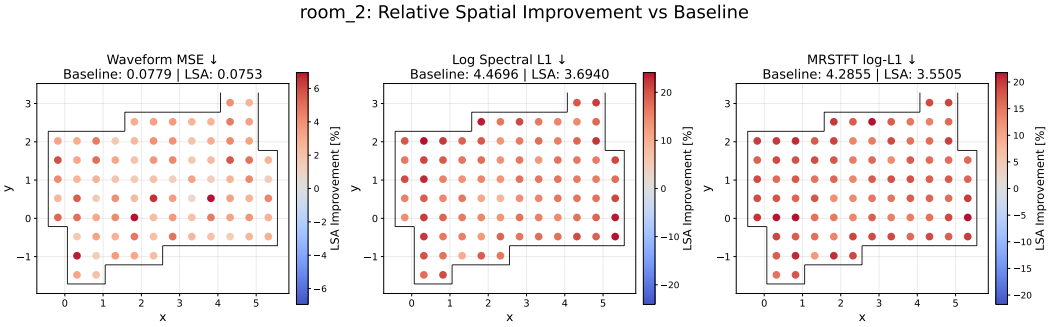



Figure 15: Spatial improvement of LSA over the NAF baseline.Room Scene.

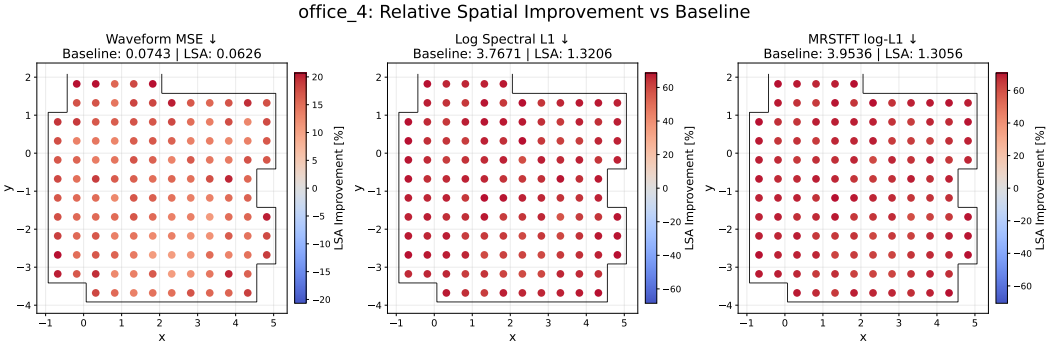



Figure 16: Spatial improvement of LSA over the NAF baseline. Office Scene

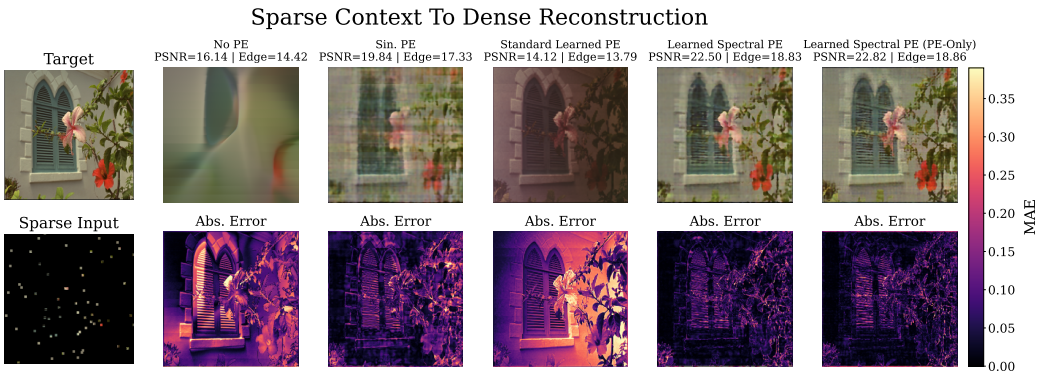



Figure 17: Sparse context $\rightarrow$ dense reconstruction under different positional encodings. Top: reconstructions. Bottom: absolute error maps.

# C Additional Comparisons

These experiments report matched-backbone activation comparisons and broader INR comparisons under their respective protocols. Each comparison specifies its tuning and evaluation protocol. STAF [28] and SL$^2$A [37] use their released implementations. WIRE [30] and MFN GaborNet [35] provide wavelet and multiplicative-filter comparisons. The fractional Jacobi experiments use the single-degree sequential fJNB construction of Aghaei [39]. We use the label fJNB throughout these tables to distinguish that construction from the Fourier KAN of Mehrabian et al. [36], discussed in Section 2.

## C.1 Image Fitting

We evaluate full-resolution Kodak images using raw normalized coordinates and RGB outputs. The controlled MLP comparisons use width 256, four nonlinear layers, minibatches of 65,536 pixels, seed 0, and the image metrics in Section A.2. We report two selection protocols: per-image tuning and a fixed configuration selected on six calibration images, then evaluated on the other 18 images. The calibration images are kodim02, kodim05, kodim07, kodim10, kodim13, and kodim23. Under the latter protocol, every image is fitted separately. The holdout concerns hyperparameter selection across images.

LSA calibration tests the $(\text{learning rate}, \text{coefficient multiplier})$ pairs $(5 \times 10^{-4}, 2)$, $(10^{-3}, 2)$, and $(10^{-3}, 1)$. All use $K = 32$, 5,000 updates, and exponential decay to 0.1 of the initial rate. The selected global configuration, $(5 \times 10^{-4}, 2)$, is frozen for Figures 4, 6 and 10. Per-image LSA tuning selects separately from these same three candidates.

For the paired PSNR intervals in Tables 5 to 7, we resample paired image-level differences 20,000 times and report the 2.5th and 97.5th percentiles of the bootstrap means. Means and paired gains are rounded independently.

**STAF.** We use the released skip-connected STAF implementation with first-layer $\omega_0 = 60$, hidden-layer $\omega_0 = 30$, and its other default settings. The per-image sweep comprises 48 configurations with $\tau \in \{3, 5, 8, 16\}$ sinusoidal terms, learning rates $\{2.5 \times 10^{-4}, 5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}\}$, and constant or decayed schedules with final multipliers 0.1 and 0.05. An additional eight configurations use $\tau = 3$, learning rates $\{3 \times 10^{-3}, 5 \times 10^{-3}, 10^{-2}, 2 \times 10^{-2}\}$, and the two decayed schedules. Thus STAF receives 56 candidates per image, with 5,000 updates per fit. LSA uses three candidates per image. The boundary extension improves STAF on 22 of 24 images, by 0.545 dB on average. Selection uses each image's best evaluated fitting PSNR. Table 5 summarizes the selected results, and Table 9 gives the paired image values.

Table 5: Per-image Kodak tuning. Values are mean and standard deviation across 24 images. Both methods select their configurations separately for each image.

| Method | Candidates/image | PSNR (dB) | SSIM | PSNR wins |
|---|---|---|---|---|
| LSA | 3 | $38.308 \pm 2.849$ | $0.9568 \pm 0.0111$ | 24/24 |
| STAF | 56 | $35.425 \pm 2.190$ | $0.9209 \pm 0.0119$ | 0/24 |

The mean paired PSNR difference is 2.883 dB, with a 95% bootstrap interval of $[2.439, 3.295]$ dB. LSA also has higher SSIM on every image. These results establish performance under the specified per-image search budgets.

The matched-budget global STAF configuration in Figure 4 is selected from the completed $56 \times 24$ sweep using mean PSNR on the same six calibration images, then frozen across all 24 images. The selected configuration has $\tau = 3$, learning rate $3 \times 10^{-3}$, and decay to 0.05 of the initial rate over 5,000 updates. Its all-24 mean is 35.369 dB. The extended-budget global STAF result in Table 8 uses a separate historical search over term count, learning rate, and schedule, with up to 10,000 updates and plateau-based early stopping. Its all-24 mean is 35.146 dB. Both full-set means include calibration images and are descriptive.

**SL$^2$A.** We evaluate the official architecture under a shared 5,000-update image-fitting protocol. The full model uses Chebyshev degree 256 and rank 128, with 330,243 parameters. The parameter-

matched model uses degree 192 and rank 64, with 199,171 parameters. LSA has 199,043 trainable parameters. Each SL$^2$A variant is calibrated over learning rates $\{5 \times 10^{-4}, 10^{-3}, 2 \times 10^{-3}, 4 \times 10^{-3}, 8 \times 10^{-3}\}$ on the same six images. The selected rate is $4 \times 10^{-3}$ for both full and parameter-matched models, with the same $0.1^{\min(s/5000,1)}$ decay schedule. These settings are frozen before the 18-image evaluation. We also report an author-style SL$^2$A run with degree 256, rank 128, 501 epochs, batch size 16,384, and seed 42. This optimization protocol is reported separately in Table 6.

Table 6: SL$^2$A comparison on Kodak. The 18-image holdout is the primary comparison after calibration. The 24-image means also include the six calibration images. Wins refer to LSA relative to each row on the holdout.

| Method | Parameters | Holdout PSNR | All-24 PSNR | LSA wins |
|---|---|---|---|---|
| LSA | 199,043 | 38.045 | 38.110 | - |
| SL$^2$A, full | 330,243 | 37.857 | 37.298 | 10/18 |
| SL$^2$A, parameter matched | 199,171 | 35.910 | 35.995 | 18/18 |
| SL$^2$A, author protocol | 330,243 | 37.463 | 37.541 | 15/18 |

The mean difference for LSA minus full SL$^2$A is $0.188$ dB, with ten wins and eight losses for LSA. The exact sign test and Wilcoxon test give $p = 0.815$ and $p = 0.229$, respectively. Full SL$^2$A has a slightly lower descriptive mean and uses 65.9% more parameters. With nearly matched parameter counts, LSA gains 2.135 dB and wins on all 18 images. Against the author-style protocol, LSA gains 0.582 dB with 15/18 wins. These comparisons quantify reconstruction quality and parameter cost under the specified protocols.

**WIRE, MFN, and fJNB.** These comparisons use the same six-image calibration and 18-image holdout. WIRE is selected from 32 candidates covering frequency, spatial scale, learning rate, initialization, and schedule. MFN GaborNet is selected from 36 candidates covering input scale, Gabor scale, and learning rate. Both receive up to 10,000 updates, and LSA receives three calibration candidates and 5,000 updates. The fJNB sweep uses degrees $q \in \{2, 3, 4, 5, 6\}$, shared Adam rates $\{10^{-3}, 3 \times 10^{-3}, 10^{-2}, 3 \times 10^{-2}\}$, and constant or decayed schedules, giving 240 image-configuration attempts. The selected stable configuration uses $q = 5$, learning rate $3 \times 10^{-3}$, and a constant schedule. This single-degree sequential construction is the scope of the fJNB comparison.

Table 7: Additional Kodak holdout results after six-image calibration. Confidence intervals are paired bootstrap intervals for LSA's mean PSNR gain. Means and paired gains are rounded independently.

| Method | Holdout PSNR | LSA gain (dB) | 95% interval | LSA wins |
|---|---|---|---|---|
| LSA | 38.045 | - | - | - |
| fJNB | 22.985 | 15.060 | [14.595, 15.559] | 18/18 |
| WIRE | 36.477 | 1.568 | [1.143, 1.970] | 17/18 |
| MFN GaborNet | 35.086 | 2.959 | [2.764, 3.157] | 18/18 |

On the holdout, LSA and fJNB obtain SSIM $0.9545 \pm 0.0115$ and $0.5449 \pm 0.1274$, respectively. Their PSNR standard deviations are 2.631 and 2.325 dB. Table 8 collects the descriptive 24-image means for globally selected configurations. These full-set results provide context alongside the holdout and per-image evaluations.

### C.2 Audio Fitting

We use the fixed 10-clip NSynth and 40-clip LibriSpeech manifests, one-second mono signals resampled to 48 kHz, waveform MSE, and PSNR with unit dynamic range, as in Section A.1. The coordinate MLP comparisons use width 256, four hidden layers, Adam, and 5,000 updates. PSNR is selected from evaluated checkpoints on the fitting grid. FF16 denotes 16 Gaussian Fourier-feature bands with the raw coordinate included. Settings are selected separately for each method and evaluated on the same clips. These are signal-fitting comparisons.

**STAF.** We evaluate the official audio-style raw-coordinate configuration with $\tau = 5$ and first-layer $\omega_0 = 3000$, as well as matched Fourier-feature settings at $\sigma = 20$ and 100. A $\tau = 16$ setting matches

Table 8: Descriptive Kodak results over all 24 images with globally selected configurations. Calibration images are included. STAF, WIRE, and MFN receive up to 10,000 updates. LSA, SIREN, and FINER use 5,000.

| Method | Mean PSNR (dB) |
|---|---|
| SIREN | 34.990 |
| FINER | 37.317 |
| STAF | 35.146 |
| WIRE | 36.596 |
| MFN GaborNet | 35.151 |
| fJNB | 22.948 |
| LSA | 38.110 |

Table 9: Per-image STAF comparison after selecting from 56 STAF and three LSA configurations per image.

| Image | LSA PSNR | STAF PSNR | LSA SSIM | STAF SSIM |
|---|---|---|---|---|
| kodim01 | 35.671 | 32.973 | 0.9562 | 0.9240 |
| kodim02 | 39.563 | 35.636 | 0.9529 | 0.8902 |
| kodim03 | 42.714 | 38.166 | 0.9737 | 0.9321 |
| kodim04 | 39.194 | 35.999 | 0.9563 | 0.9086 |
| kodim05 | 35.157 | 31.669 | 0.9605 | 0.9157 |
| kodim06 | 38.192 | 34.668 | 0.9614 | 0.9256 |
| kodim07 | 41.622 | 37.352 | 0.9780 | 0.9388 |
| kodim08 | 32.677 | 31.542 | 0.9368 | 0.9212 |
| kodim09 | 41.458 | 37.500 | 0.9617 | 0.9221 |
| kodim10 | 40.621 | 36.995 | 0.9631 | 0.9218 |
| kodim11 | 38.268 | 35.267 | 0.9570 | 0.9142 |
| kodim12 | 40.604 | 37.835 | 0.9620 | 0.9306 |
| kodim13 | 31.910 | 31.334 | 0.9364 | 0.9323 |
| kodim14 | 36.461 | 33.264 | 0.9568 | 0.9122 |
| kodim15 | 37.810 | 36.565 | 0.9417 | 0.9149 |
| kodim16 | 40.987 | 36.891 | 0.9731 | 0.9268 |
| kodim17 | 40.085 | 37.235 | 0.9664 | 0.9359 |
| kodim18 | 35.968 | 33.370 | 0.9442 | 0.9068 |
| kodim19 | 37.853 | 35.374 | 0.9508 | 0.9190 |
| kodim20 | 39.199 | 37.962 | 0.9572 | 0.9405 |
| kodim21 | 38.674 | 35.635 | 0.9569 | 0.9150 |
| kodim22 | 37.464 | 34.624 | 0.9476 | 0.9024 |
| kodim23 | 41.965 | 38.376 | 0.9675 | 0.9295 |
| kodim24 | 35.286 | 33.968 | 0.9440 | 0.9203 |

LSA's harmonic count. Table 10 gives all four settings. The raw-coordinate setting is strongest among those tested, reaching 68.79 dB on NSynth and 68.40 dB on LibriSpeech. FINER reaches 67.19 and 66.32 dB, respectively. LSA at FF16, $\sigma = 100$, and $K = 16$ reaches $86.72 \pm 21.18$ and $90.78 \pm 18.57$ dB, with 8/10 and 37/40 wins over the strongest STAF setting.

Table 10: STAF audio configurations. Entries are mean PSNR and standard deviation across clips. The raw-coordinate setting uses first-layer $\omega_0 = 3000$.

| Input | Terms $\tau$ | NSynth | LibriSpeech |
|---|---|---|---|
| Raw coordinates | 5 | $68.79 \pm 3.04$ | $68.40 \pm 6.29$ |
| FF16, $\sigma = 20$ | 5 | $55.11 \pm 11.63$ | $50.24 \pm 1.38$ |
| FF16, $\sigma = 100$ | 5 | $51.83 \pm 4.47$ | $51.85 \pm 1.28$ |
| FF16, $\sigma = 100$ | 16 | $50.60 \pm 2.44$ | $51.21 \pm 1.43$ |

**SL$^2$A.** The sweep covers Chebyshev degrees 256 and 512 and ranks 64 and 128, using the official learnable-activation and fusion architecture. Table 11 reports the configurations at learning rate $10^{-3}$. Degree 512 and rank 128 give the strongest mean result on both datasets, with 68.03 dB on NSynth and 47.42 dB on LibriSpeech. LSA wins on 7/10 and 40/40 clips, respectively.

Table 11: SL$^2$A audio results for the tested degree and rank settings. Values are mean PSNR and standard deviation across clips. All rows use learning rate $10^{-3}$.

| Degree | Rank | Parameters | NSynth | LibriSpeech |
|---|---|---|---|---|
| 512 | 128 | 329,473 | $68.03 \pm 10.20$ | $47.42 \pm 6.22$ |
| 512 | 64 | 231,169 | $67.16 \pm 10.57$ | $46.35 \pm 6.62$ |
| 256 | 128 | 263,937 | $59.16 \pm 16.05$ | $38.84 \pm 7.18$ |
| 256 | 64 | 165,633 | $59.47 \pm 15.35$ | $38.37 \pm 7.28$ |

**WIRE and fJNB.** The selected WIRE settings are $(\omega_0, s_0) = (10, 10)$ for NSynth and $(10, 5)$ for LibriSpeech, both at learning rate $10^{-3}$. For fJNB, the sweep covers degrees $q \in \{2, 3, 4, 5, 6\}$, learning rates $\{10^{-3}, 5 \times 10^{-4}, 2 \times 10^{-4}, 10^{-4}\}$, and Fourier scales $\sigma \in \{20, 100\}$, together with raw-coordinate runs. We report the strongest completed configurations. Table 12 includes both input regimes for fJNB and the corresponding LSA results.

Table 12: Additional audio fitting comparisons. Gains and wins refer to LSA relative to the specified baseline on the same clips. fJNB uses the fractional Jacobi construction.

| Dataset | Baseline | Input | Baseline PSNR | LSA PSNR | Gain | Wins |
|---|---|---|---|---|---|---|
| NSynth | fJNB | Raw | 16.45 | 47.41 | 30.96 | 9/10 |
| LibriSpeech | fJNB | Raw | 25.23 | 31.96 | 6.73 | 31/40 |
| NSynth | fJNB | FF16, $\sigma = 100$ | 55.09 | 86.72 | 31.63 | 10/10 |
| LibriSpeech | fJNB | FF16, $\sigma = 100$ | 43.97 | 90.78 | 46.81 | 40/40 |
| NSynth | WIRE | Selected | 47.11 | 86.72 | 39.61 | 10/10 |
| LibriSpeech | WIRE | Selected | 51.79 | 90.78 | 38.99 | 40/40 |

### C.3 Novel View Synthesis

We adapt the official fractional Jacobi block as an activation in the NeRF backbone. Degree and learning rate are tuned on Chair, Ficus, and Ship using 50,000 steps and mean final validation PSNR, followed by one 150,000-step run on each of the eight scenes. The fJNB and LSA comparison uses the same Blender split, reduced-view protocol, backbone, ray budget, sample count, width, depth, optimizer family, and final-step evaluation rule. This protocol concerns the adapted fJNB block.

Table 13 also reproduces WIRE PSNR from the reduced-view Blender table in FINER [24], which uses 25 training views. WIRE is a published reference with its source protocol, while the fJNB and LSA columns are evaluated in the shared backbone.

Table 13: Additional Blender PSNR results. WIRE is reproduced from Liu et al. [24]. fJNB and LSA use the shared-backbone evaluation described above. Means are across eight scenes.

| Scene | WIRE (published) | fJNB (adapted) | LSA |
|---|---|---|---|
| Chair | 29.31 | 30.15 | 34.18 |
| Drums | 22.22 | 18.08 | 24.79 |
| Ficus | 25.91 | 22.24 | 28.85 |
| Hotdog | 30.11 | 31.81 | 33.36 |
| Lego | 25.76 | 22.67 | 29.39 |
| Materials | 25.05 | 24.73 | 27.22 |
| Mic | 32.35 | 31.58 | 33.91 |
| Ship | 21.15 | 19.62 | 22.60 |
| Mean | 26.48 | 25.11 | 29.29 |

### C.4 Ablations

The truncation ablation in Table 14 evaluates six Kodak images. Quality increases through $K = 32$ and saturates at $K = 64$ under this protocol. Together with the runtime and memory measurements in Table 17, the results identify $K$ as a control on quality and activation cost. The quality and resource tables concern their respective image and audio settings.

Table 14: Harmonic-count ablation on six Kodak images. Standard deviations and medians are computed across images.

| $K$ | Mean PSNR | Std. PSNR | Median PSNR | Mean SSIM |
|---|---|---|---|---|
| 4 | 30.44899 | 5.28322 | 29.17142 | 0.81556 |
| 8 | 35.35890 | 4.63919 | 34.52907 | 0.92290 |
| 16 | 38.67377 | 4.34150 | 40.33058 | 0.96070 |
| 32 | 39.50432 | 4.32379 | 41.38003 | 0.96614 |
| 64 | 39.34945 | 4.14439 | 40.49803 | 0.96494 |

For the frozen-coefficient ablation, both variants use raw coordinates, $K = 32$, the same initial state, data order, random seed, optimizer settings, and 5,000-step budget. The frozen variant keeps $a$ at its small-random initialization while training all ordinary weights and biases. The coefficients were verified to remain bitwise unchanged. Hyperparameters are selected on six calibration images, with the other 18 used for the primary evaluation.

Table 15: Learning versus freezing the near-identity coefficients. PSNR is mean and standard deviation across images. Confidence intervals refer to the paired mean PSNR gain.

| Evaluation | Coefficients | PSNR | SSIM | LSA wins | Gain 95% interval |
|---|---|---|---|---|---|
| 18 holdout | Learned | $38.05 \pm 2.63$ | 0.9545 | 18/18 | $[17.45, 18.87]$ |
| 18 holdout | Frozen | $19.91 \pm 1.78$ | 0.4861 | | |
| 24 pooled | Learned | $38.11 \pm 2.90$ | 0.9557 | 24/24 | $[17.56, 19.17]$ |
| 24 pooled | Frozen | $19.76 \pm 1.95$ | 0.4834 | | |

The paired mean PSNR gains are 18.13 dB on the holdout and 18.35 dB on all 24 images. The frozen activation remains close to the identity, so this comparison measures the contribution of learning spectral coefficients from that initialization. It establishes that the learned nonlinear response is important for the reported fits. The result is specific to this near-identity frozen control.

## D Additional Analysis

In this section we provide additional analysis of our proposed method and its shaping of the INR.

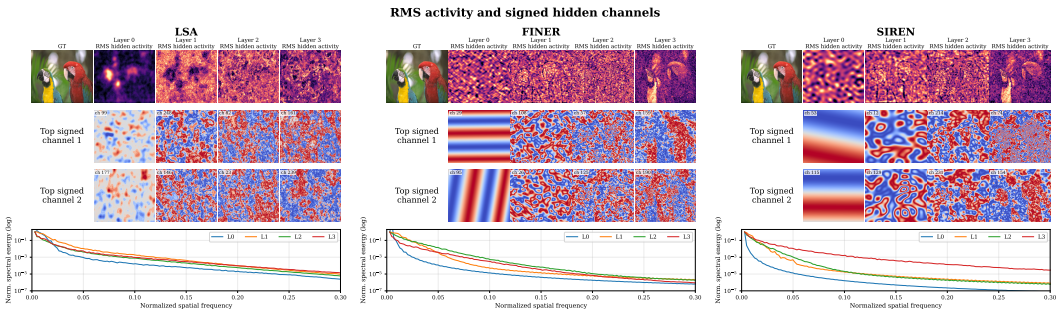



Figure 18: Hidden-representation diagnostics on image fitting. For each method, we visualize layer-wise RMS hidden activity, the two highest-variance signed hidden channels, and the radial spectra of those selected channels. LSA produces more spatially localized and less purely periodic hidden features than SIREN and FINER, supporting the view that learned spectral activations modify the internal basis rather than only the final reconstruction.

### D.1 Hidden-Representation Diagnostics

To further inspect how learned spectral activations modify the internal representation, we visualize hidden features from trained image INRs in Figure 18. For each method and layer, we evaluate the network on a dense coordinate grid and record the hidden activations $h^{(\ell)}(x) \in \mathbb{R}^C$. We summarize the layer-wise activity by the RMS map

$$A^{(\ell)}(x) = \left( \frac{1}{C} \sum_{c=1}^{C} \left( h_c^{(\ell)}(x) \right)^2 \right)^{1/2},$$

which indicates where hidden units are active after aggregating over channels. We also show the two channels with largest spatial variance in each layer, preserving their sign, to visualize representative high-energy hidden basis functions.

The diagnostic reveals a qualitative difference in how the methods organize their hidden representations. SIREN and FINER exhibit more globally periodic early-layer channels, consistent with fixed sinusoidal or frequency-warped nonlinear operators. Their later layers gradually become more image-aligned, suggesting that localized structure is assembled through depth. In contrast, LSA produces less purely periodic and more spatially localized hidden features already in early layers. This supports the mechanism proposed in Section 3: learned spectral activations change the neuron-level basis used to compose heterogeneous structure, rather than only improving the final reconstruction through the output layer.

The bottom row of Figure 18 shows radial spectra of the selected high-variance channels. These spectra should be interpreted as a diagnostic of dominant hidden components rather than a complete spectrum of the full representation, since they are computed only from the selected channels. Nevertheless, they are consistent with the spatial visualizations: LSA exhibits similar spectral behavior across layers, while SIREN and FINER show stronger layer-wise changes. This suggests that LSA expresses part of the required spectral complexity directly inside individual hidden responses, whereas fixed activation families rely more heavily on progressive composition across layers.

### D.2 Learned Activation Coefficient Spectra

In addition to visualizing hidden activations, we inspect the learned spectral operators themselves. For each trained image INR, LSA contains layer-wise activation coefficients $\{a_k^{(\ell)}\}_{k=1}^{K}$ in Equation (2). These coefficients are model parameters, not data-dependent activations. We therefore visualize them directly to determine whether the learned activation uses a nontrivial harmonic profile.

For each layer $\ell$, we plot

$$M_{\ell,k} = \log_{10} \left( |a_k^{(\ell)}| + 10^{-8} \right),$$

where $k$ indexes the harmonic component. Each heatmap in Figure 19 corresponds to a separately trained INR on one Kodak image. Rows denote layers and columns denote harmonic indices.

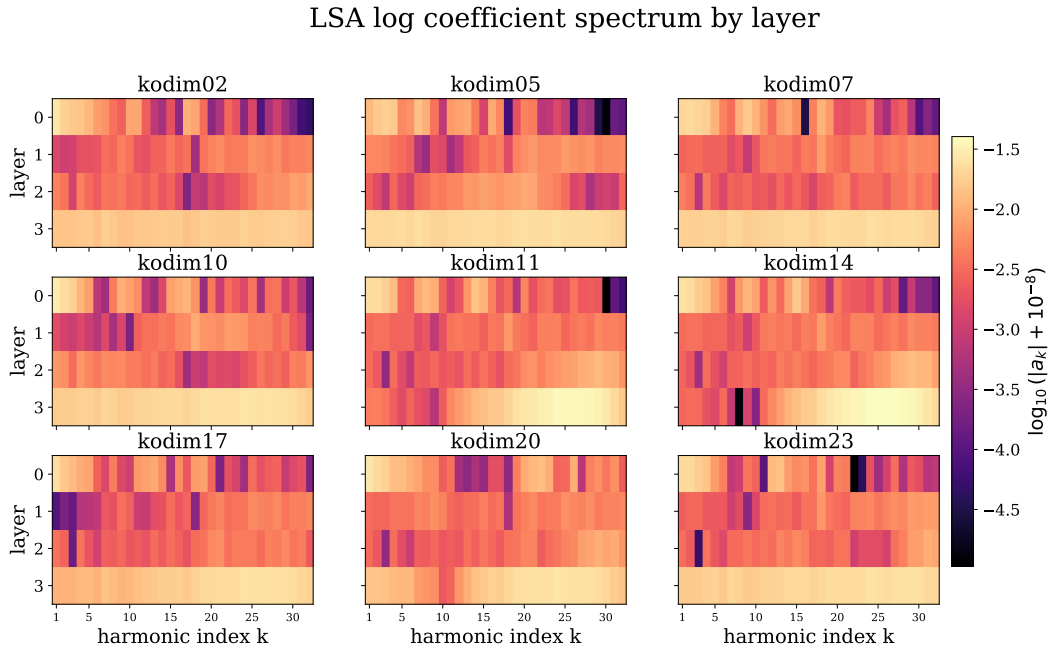



Figure 19: Learned LSA coefficient spectra across Kodak images. Each panel shows $\log_{10}(|a_k^{(\ell)}| + 10^{-8})$ for harmonic index $k$ and layer $\ell$. The learned operators exhibit layer-dependent harmonic profiles and image-dependent variation, indicating that LSA adapts its internal spectral basis rather than using a fixed spectrum.

The learned coefficients exhibit clear layer-dependent structure. Early layers often show selective harmonic suppression or enhancement, visible as dark or bright bands across specific harmonic indices. Later layers, especially the final hidden layer, tend to retain broader and more uniform harmonic support. The overall pattern is consistent across images, while the exact emphasized or suppressed harmonics vary, indicating that the learned spectral operator adapts to the fitted signal rather than collapsing to a fixed handcrafted profile.

This diagnostic is complementary to the hidden-activation visualization in Figure 18. The coefficient spectra show that LSA learns nontrivial harmonic operators in parameter space, while the hidden-channel diagnostics show how these operators manifest as spatial features over the coordinate domain.

### D.3 Learned Activation Profiles

We further inspect the activation functions learned by LSA on Kodak image fitting. For each trained image INR, the activation at layer $\ell$ is parameterized by the coefficients $\{a_k^{(\ell)}\}_{k=1}^K$ in Equation (2). We visualize these parameters in two complementary ways: first, by plotting the coefficient magnitudes $|a_k^{(\ell)}|$ as a function of harmonic index $k$, and second, by plotting the resulting learned activation waveform $\phi^{(\ell)}(u)$ over the pre-activation range $u \in [-1, 1]$.

The coefficient spectra in Figure 19 show that the learned harmonic profiles are structured rather than uniform. Early layers often display selective attenuation or enhancement of specific harmonics, while later layers tend to retain broader harmonic support. The precise profile varies across images, suggesting that the learned operator adapts to the fitted signal.

The waveform plots in Figure 21 provide the corresponding activation-space view. The learned activations remain close to the residual identity mapping, which is consistent with the small initialization used in Section 3. At the same time, the curves contain localized deviations and small oscillatory corrections, especially around frequently used pre-activation regions. This supports the intended role of LSA: the method does not require large departures from a stable base activation, but learns targeted harmonic corrections that reshape the hidden response.

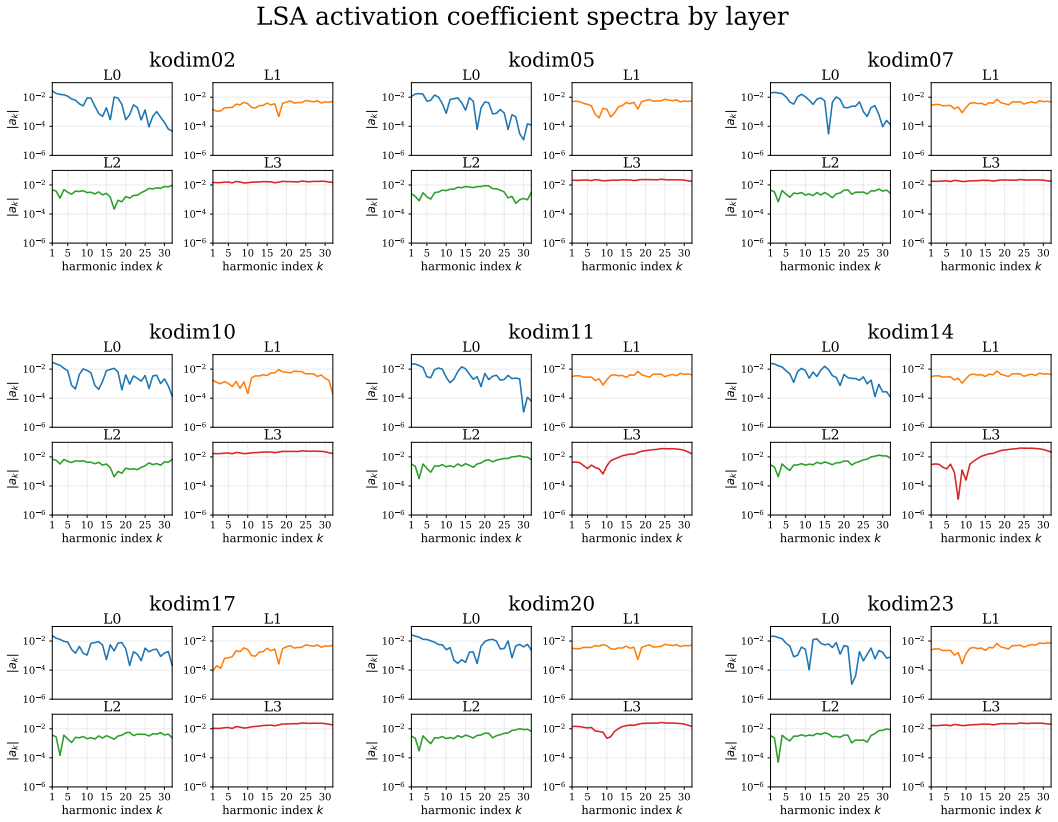



Figure 20: Learned activation coefficient spectra across Kodak images. Each panel shows the magnitude $|a_k^{(\ell)}|$ of the learned harmonic coefficients for one image and layer. The coefficients exhibit layer-dependent profiles and image-dependent variation, indicating that LSA learns nontrivial spectral operators rather than using a fixed harmonic profile.

Together, these diagnostics complement the hidden-representation analysis in Section D.1. The coefficient plots show that the spectral operator itself is learned in parameter space, while the waveform plots show how these coefficients modify the pointwise nonlinear response applied inside the INR.

### D.4 Resource Analysis

LSA is compact in trainable parameters, but its current implementation introduces additional activation compute. When coefficients are shared within each activation layer, LSA adds only $KL_{\text{act}}$ trainable scalar parameters, where $K$ is the number of harmonic terms and $L_{\text{act}}$ is the number of activation layers. For example, in our reported width-256, depth-4 audio INR, $K = 16$ adds only $16 \times 4 = 64$ trainable parameters.

The main overhead comes from evaluating the harmonic expansion pointwise at every hidden activation. For a batch of $B$ coordinates, hidden width $W$, $L_{\text{act}}$ activation layers, and $K$ harmonics, the activation evaluates $BWL_{\text{act}}K$ harmonic responses. Thus, the practical cost is dominated by intermediate activation tensors and autograd storage rather than by the number of trainable parameters.

To quantify this cost, we benchmarked SIREN, FINER, and LSA under matched input regimes. In the primary comparison, all methods use the same FF16 input encoding. Runtime is measured on an NVIDIA RTX A6000 using full-grid 48k audio fitting. Training time includes forward pass, backward pass, and optimizer update. Evaluation time is forward-only. Memory is peak CUDA allocation. LSA is measured using the batched PyTorch implementation used in our experiments.

The results show that LSA has negligible parameter overhead but higher activation cost in the current implementation. At depth 4 and $K = 16$, LSA adds only 64 parameters over the FF16 baselines, while

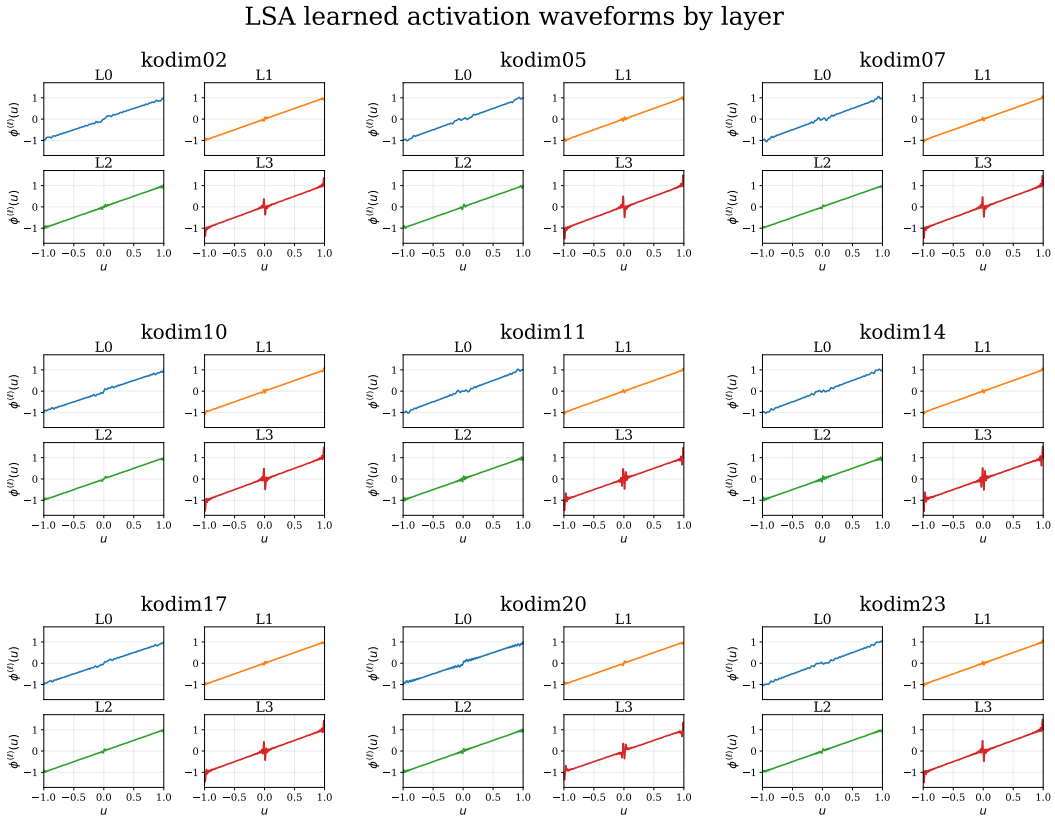



Figure 21: Learned activation waveforms across Kodak images. Each panel plots the resulting activation $\phi^{(\ell)}(u)$ for one image and layer. Although the learned functions remain close to the residual identity shape, they contain localized harmonic deviations whose form varies across layers and fitted images.

Table 16: Resource usage under the matched FF16 input regime on full-grid 48k audio fitting. All models use width 256. Training time includes forward, backward, and optimizer update. Evaluation time is forward-only. Memory is peak CUDA allocation.

| Depth | Method | Params | Extra act. params | Train time (ms) | Train mem. (GB) | Eval time (ms) | Eval mem. (GB) |
|---|---|---|---|---|---|---|---|
| 4 | SIREN FF16 | 206,337 | 0 | 8.26 | 0.48 | 3.56 | 0.16 |
| | FINER FF16 | 206,337 | 0 | 14.20 | 0.85 | 5.55 | 0.21 |
| | LSA FF16, $K = 16$ | 206,401 | 64 | 97.93 | 7.53 | 35.08 | 1.58 |
| 6 | SIREN FF16 | 337,921 | 0 | 12.77 | 0.67 | 5.46 | 0.16 |
| | FINER FF16 | 337,921 | 0 | 21.65 | 1.22 | 8.45 | 0.21 |
| | LSA FF16, $K = 16$ | 338,017 | 96 | 147.84 | 10.56 | 52.87 | 1.58 |

requiring additional runtime and peak memory due to evaluating and storing the harmonic responses. This reflects a tradeoff between a richer learned activation operator and per-step computational cost. Since several experiments also show improved optimization trajectories, including the convergence curves in Figure 10, part of this per-step cost may be offset in settings where LSA reaches a target reconstruction quality in fewer optimization steps. We leave a systematic quality-vs-wall-clock analysis to future work.

**Scaling with the number of harmonics.** The activation cost scales with the number of harmonic terms $K$. Table 17 reports LSA resource usage for $K \in \{8, 16, 32, 64\}$ under the same FF16 full-grid setting.

As expected, both runtime and memory increase with $K$, since each additional harmonic introduces another pointwise sinusoidal response. This makes $K$ a direct knob for trading spectral flexibility against computational cost.

Table 17: LSA scaling with the number of harmonic terms $K$ under the matched FF16 full-grid 48k setting. All models use width 256. The implementation is the batched PyTorch harmonic evaluation used in Table 16.

| Depth | $K$ | Train time (ms) | Train mem. (GB) | Eval time (ms) | Eval mem. (GB) |
|---|---|---|---|---|---|
| 4 | 8 | 52.78 | 3.88 | 19.24 | 0.85 |
| | 16 | 98.57 | 7.53 | 35.33 | 1.58 |
| | 32 | 189.87 | 14.87 | 67.26 | 3.05 |
| | 64 | 341.09 | 29.55 | 115.23 | 5.98 |
| 6 | 8 | 79.13 | 5.44 | 28.77 | 0.85 |
| | 16 | 147.84 | 10.56 | 52.87 | 1.58 |
| | 32 | 284.81 | 20.82 | 100.88 | 3.05 |
| | 64 | 511.70 | 41.36 | 172.70 | 5.98 |

Table 18: Implementation diagnostic for LSA in a no-FF, small-batch setting ($B = 4096$, width 256, depth 4, fp32). The accumulate implementation avoids explicitly materializing the full harmonic tensor, reducing memory at the cost of additional runtime.

| $K$ | Implementation | Train time (ms) | Train mem. (GB) | Eval time (ms) | Eval mem. (GB) |
|---|---|---|---|---|---|
| 8 | Batched | 6.63 | 0.35 | 2.26 | 0.09 |
| | Accumulate | 6.98 | 0.30 | 2.35 | 0.04 |
| 16 | Batched | 11.43 | 0.66 | 4.08 | 0.15 |
| | Accumulate | 13.13 | 0.55 | 4.40 | 0.04 |

**Implementation dependence.** The measured overhead also depends on how the harmonic activation is implemented. A batched implementation materializes the harmonic tensor over the $K$ terms, while an accumulate implementation evaluates the terms sequentially:

$$\phi(u) = u + \sum_{k=1}^{K} a_k \sin(2\pi k u).$$

The batched implementation is faster, but uses more memory. The accumulate implementation reduces memory, especially during inference, but is slower.

This diagnostic suggests that the overhead is partly implementation-dependent. In particular, inference memory can be reduced by avoiding materialization of the full harmonic tensor. More optimized implementations, such as fused CUDA kernels, checkpointed harmonic evaluation, or adaptive sparsification over active harmonics, could further reduce this cost without changing the mathematical form of LSA. Thus, the current implementation should be viewed as a straightforward reference implementation rather than a fully optimized activation kernel.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our stated contributions are a formulation for a novel activation function for INRS, brought in Section 3, and empirical analysis of our method, brought in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 5 and Section D.4.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Section 3 states the finite-width descent identity and shared-state fixed-NTK decomposition, with assumptions, derivations, and controlled validations provided in Section A.6. The local scope and residual-alignment conditions are stated explicitly.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We add link to an official codebase that reproduces main results, also Section A includes all required information.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide a link to our code in the abstract.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Included in Section A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Results in Section 4 include standard-deviation values where applicable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We do not directly report specific runtimes for every experiment, since this paper is based on a large amount of relatively light experiments. However, Section A and Section D.4 jointly supply an accurate notion of resource consumption.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Yes. The work is methodological and uses established public benchmark datasets. It involves no human-subject experiments or private data, and we are not aware of ethical concerns specific to the experiments conducted in this paper.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: We cannot identify any immediate societal impact our results or conclusions may have.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper does not release any models or data relevant to this matter.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the original works and tools used in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release an official code repository with the implementation and instructions for reproducing the main experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our research is not relied on human subjects in any way

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our research is not relied on human subjects in any way

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as an important, original, or non-standard component of the core method. Any LLM use is limited to writing, editing, or formatting assistance and does not affect the methodology, experiments, or scientific claims.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.