FUTON: Fourier Tensor Networks for Implicit Neural Representations

Pooya Ashtari[1,*] **Pourya Behmandpoor**[2] **Nikos Deligiannis**[2] **Aleksandra Pižurica**[1]
[1]Ghent University [2]Vrije Universiteit Brussel & imec

Abstract

Implicit neural representations (INRs) have emerged as powerful tools for encoding signals, yet dominant MLP-based designs often suffer from slow convergence, and how their accuracy depends on their size is known only empirically. We introduce **FUTON (Fourier Tensor Network)**, which models signals as generalized Fourier series whose coefficients are parameterized by a low-rank tensor decomposition, read out by a linear or MLP head. FUTON combines complementary inductive biases: Fourier bases capture smoothness and periodicity, while the low-rank parameterization enforces low-dimensional spectral structure. Beyond universal approximation, we show that with a linear head, the best FUTON's error splits exactly into the truncation of the signal's series and the best low-rank error of its coefficients, which we bracket from the samples before training, with a FUTON that attains the upper bound. On image fitting, occupancy volumes, and novel view synthesis, against twelve INRs of matched size, FUTON achieves the best average quality, reaches the final quality of each task's strongest baseline 4–34$\times$ sooner than that baseline, infers fastest, and trains in the least memory.

Project page: `pashtari.github.io/inro`

1 Introduction

Implicit neural representations (INRs), also known as *neural fields*, have emerged as a powerful paradigm for representing signals, including images, volumes, and radiance fields. Unlike grid-based representations that store signals as discrete arrays (*e.g.*, pixels or voxels), INRs encode signals as continuous functions parameterized by neural network weights, mapping coordinates to signal values. This coordinate-based parameterization offers key advantages: (1) *resolution independence*, enabling super-resolution without retraining; (2) *memory efficiency* through compact parameter storage; (3) natural handling of *irregular sampling*; and (4) *differentiability* with respect to coordinates, facilitating gradient-based optimization for inverse problems.

INRs have demonstrated strong performance across diverse tasks, including 3D reconstruction [1, 2], view synthesis [3], and inverse problems [4, 5]. Typical INRs employ multilayer perceptrons (MLPs) with specialized activations, *e.g.*, periodic sine functions [6], Gaussian nonlinearities [7], or complex Gabor wavelets [8], all of which significantly outperform ReLU-based networks for signal representation [9].

MLP-based INRs suffer from two key limitations: (1) *slow convergence*, leading to long training times; and (2) an *opaque size–accuracy relation*: how width and depth translate into error is known only empirically. These issues arise because MLPs lack inductive biases aligned with the low-dimensional structure of natural signals. Hybrid INRs, which pair a small MLP with feature grids such as hash tables [10] or factorized planes [11, 12], converge faster but spend most of their parameters on the grids.

*Corresponding author: `pooya.ashtari@ugent.be`

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

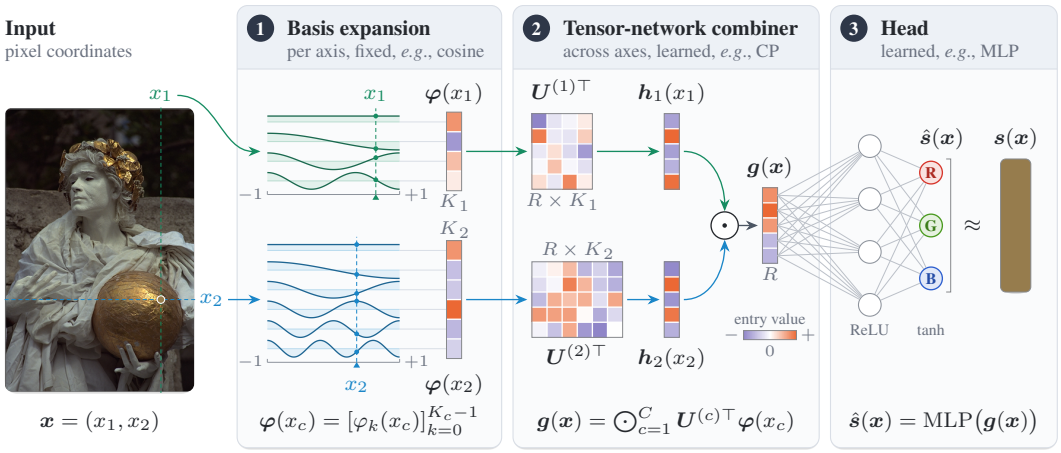


Figure 1: **FUTON's forward pass** for one pixel of an image ($C = 2$, $D = 3$). **(1)** A fixed basis, here cosines, expands each coordinate x_c into K_c functions, at spectral resolution $K = (K_1, K_2)$. **(2)** The combiner's factor matrices $\boldsymbol{U}^{(c)}$ project each expansion onto R components, multiplied across axes into $\boldsymbol{g}(\boldsymbol{x})$: a CP decomposition of the coefficient tensor. **(3)** A learned head, here an MLP, maps $\boldsymbol{g}(\boldsymbol{x})$ to the signal. Arrays hold exact values for random factors, at sizes far below practice.

We introduce *Fourier Tensor Network (FUTON)*, an INR that addresses these challenges by expressing signals as *generalized Fourier series* whose coefficients are parameterized via a *low-rank tensor decomposition*. This design embeds complementary inductive biases: Fourier bases capture smoothness and periodicity, while low-rank factorization captures the intrinsic low-dimensional spectral structure of natural signals. As Fig. 1 shows, a fixed basis expands each coordinate, a tensor-network combiner multiplies the axes' projections, and a small head maps the result to the signal.

FUTON leverages the insight that structured signals exhibit frequency-domain correlations amenable to low-rank approximation. With a Canonical Polyadic (CP) decomposition [13], the construction is classical, a *separated representation* [14] with a fixed basis, used at modest resolution (Sec. 2); FUTON brings it to INRs, trained end to end at a spectral resolution per axis comparable to the signal's own.

Contributions. Our main contributions are:

- **Architecture:** We propose FUTON, combining generalized Fourier series with a low-rank tensor decomposition, providing an efficient alternative to MLP-based INRs, with a spectral resolution per axis, a rank that sets its size, and a linear or MLP head.
- **Theoretical foundations:** Beyond universal approximation, we show that with a linear head, the best FUTON's error splits exactly into a truncation term and the best rank-R error of the signal's coefficients, bracket it from the samples with a FUTON that attains the upper bound, and verify on 29 signals that training lands in the bracket, to within 0.02 dB.
- **Computational efficiency:** A coordinate costs $\mathcal{O}(R \sum_c K_c)$ plus the head, linear in each K_c and in R; on a grid, the parameter-free basis is cached when the model is built and the per-axis projections are shared, so training and inference cost little beyond the head.
- **Empirical performance:** On image fitting, occupancy volumes, and novel view synthesis, against twelve INRs of matched size, FUTON achieves the best average quality, reaches the final quality of the fastest baseline, Instant-NGP [10], 2.7× to over 9× sooner than it does, infers fastest, and trains in the least memory.

Notation. Vectors are denoted by bold lowercase (*e.g.*, $\boldsymbol{x}$), matrices by bold uppercase (*e.g.*, $\boldsymbol{U}$), tensors by calligraphic bold uppercase (*e.g.*, $\boldsymbol{\mathcal{W}}$), and sets by blackboard bold uppercase (*e.g.*, $\mathbb{X}$). Individual elements are indexed by subscripts in non-bold notation: x_i, U_{ij}, $\mathcal{W}_{ijk}$. Nonnegative integers are denoted by $\mathbb{N}_0$.

2 Related Work

Implicit Neural Representations. NeRF [3] popularized representing scenes as continuous radiance fields parameterized by MLPs, enabling high-quality novel view synthesis. Subsequent work extended INRs to dynamic scenes [15, 16], shape representation [1, 2], and generative modeling [17], establishing INRs as a fundamental tool in visual computing.

Most INRs adopt MLP backbones whose inductive bias is strongly shaped by the activation function and input encoding. ReLU MLPs exhibit a low-frequency bias and struggle to fit fine detail [9]. Fourier features mitigate this by mapping coordinates to high-dimensional sinusoidal embeddings [18]. SIREN employs sinusoidal activations with principled initialization for stable optimization [6]. Alternatives include Gaussian activations [7] and complex Gabor wavelets (WIRE) [8], which improve space–frequency localization. FINER [19] introduces variable-periodic activation functions, allowing flexible spectral bias tuning and improved reconstruction of fine details. MFN [20] replaces depth with multiplicative combinations of sinusoidal or Gabor filters for efficient function approximation in coordinate-based tasks. Recent work speeds up sinusoidal networks by matching their frequency scale to the signal's spectrum (FreSh) [21], by scaling their initialization [22], or by adapting their frequencies across space [23]. Hybrid INRs instead store features on explicit grids and decode them with a small MLP: multiresolution hash tables in Instant-NGP [10], CP or vector–matrix factors in TensoRF [11], planes in K-Planes [12], and lines, planes, and a coarse volume in GA-Planes [24].

Frequency-Domain Representations. Frequency-domain ideas are used to counter the spectral bias of coordinate MLPs. Fourier features and periodic activations inject high frequencies directly into the representation, and band-limited networks control the frequencies they represent explicitly [25]. More recently, the Fourier reparameterization method [26] learns a coefficient matrix of fixed Fourier bases to compose the weights of MLPs. Unlike these approaches that modulate inputs, nonlinearities, or weights, FUTON parameterizes the Fourier coefficients of the signal directly, via a low-rank tensor decomposition, and uses at most a small MLP after the combination.

Tensor Networks. Tensor decompositions and tensor networks provide compact models for multi-dimensional data [13]. Canonical Polyadic (CP) [27, 28] expresses a tensor as a sum of rank-1 components, while Tucker [29] introduces a low-rank core with factor matrices, and tensor trains [30] chain three-way cores. In the INR context, TensoRF factorizes radiance fields into low-rank components; K-Planes extends factorization to dynamic scenes; and F-INR [31] factorizes an INR into axis-wise sub-networks combined by a tensor decomposition.

Closest to FUTON are *separated representations* [14, 32], sums of products of univariate functions, and their instances with a fixed basis for supervised regression [33], kernel machines [34], and spectral tensor trains [35]. FUTON is such a model, trained end to end at a high, per-axis spectral resolution, with an MLP head, and an exact error analysis of its linear-head form; unlike TensoRF and F-INR, it factorizes the coefficients of a series in a fixed basis.

3 Preliminaries

3.1 Implicit Neural Representations

INRs are functions $\boldsymbol{f}_{\boldsymbol{\theta}} : \mathbb{R}^C \to \mathbb{R}^D$ parameterized by neural networks with learnable parameters $\boldsymbol{\theta}$ that map continuous coordinates to signal values. For instance, in image representation, $\boldsymbol{x} = (x_1, x_2)$ represents pixel coordinates, and $\boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathbb{R}^3$ outputs RGB values. Given a ground-truth signal $\boldsymbol{s}(\boldsymbol{x})$, the goal is to learn parameters $\boldsymbol{\theta}$ such that $\boldsymbol{s}(\boldsymbol{x}) \approx \boldsymbol{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$. This is typically achieved by minimizing the mean squared error over observed coordinate–signal pairs $\{(\boldsymbol{x}_n, \boldsymbol{s}_n)\}_{n=1}^N$, where $\boldsymbol{s}_n = \boldsymbol{s}(\boldsymbol{x}_n)$.

Typical INRs use MLPs, whose activation choice critically affects capacity: a sine $\sin(\omega_0 x)$ (SIREN [6]), a Gaussian $e^{-(\gamma_0 x)^2}$ [7], or a complex Gabor wavelet $e^{j\omega_0 x} e^{-|\gamma_0 x|^2}$ (WIRE [8]).

3.2 Generalized Fourier Series

Fourier series decomposes signals into trigonometric components, enabling compact representation through frequency selectivity. The *generalized Fourier series (GFS)* [36] extends this idea to arbitrary orthonormal bases and higher-dimensional spaces, offering flexibility for diverse signal classes.

Orthonormal Basis. A set of functions $\{\varphi_k\}_{k\in\mathbb{N}_0} \subseteq L^2(\mathbb{X})$ (defined in App. A) is an **orthonormal basis (ONB)** for $L^2(\mathbb{X})$ if it satisfies **orthonormality**, $\langle\varphi_k, \varphi_\ell\rangle_{L^2} = \delta_{k\ell}$ for all $k, \ell \in \mathbb{N}_0$, with $\delta_{k\ell}$ the Kronecker delta, and **completeness**: for any function $f \in L^2(\mathbb{X})$, there exist coefficients $w_k \in \mathbb{R}$ such that the partial sums $s_K = \sum_{k=0}^{K-1} w_k\varphi_k$ converge to f in $L^2(\mathbb{X})$ as $K \to \infty$ (Definition 8). As a canonical example, the cosine functions

$$\varphi_0(x) = 1, \qquad \varphi_k(x) = \sqrt{2}\cos(k\pi x) \quad (k \geq 1) \tag{1}$$

form an ONB in $L^2([0, 1])$, widely studied in Fourier analysis [37]; coordinates in $[-1, 1]$, as in our experiments, are mapped affinely onto $[0, 1]$.

Separable Multi-dimensional Basis. Let $\{\varphi_k\}_{k\in\mathbb{N}_0}$ be a set of one-dimensional functions in $L^2([0, 1])$. We construct a set of *separable* multi-dimensional functions in $L^2([0, 1]^C)$ by products of the one-dimensional basis functions:

$$\phi_{\boldsymbol{k}}(\boldsymbol{x}) \triangleq \prod_{c=1}^{C} \varphi_{k_c}(x_c), \tag{2}$$

where $\boldsymbol{k} = (k_1, \ldots, k_C) \in \mathbb{N}_0^C$ is a multi-index, and $\boldsymbol{x} = (x_1, \ldots, x_C) \in [0, 1]^C$ are coordinates.

Theorem 1 (Separable Basis Orthonormality; proof in App. C)**.** *If $\{\varphi_k\}_{k\in\mathbb{N}_0}$ is an ONB for $L^2([0, 1])$, then $\{\phi_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{N}_0^C}$ defined in* (2) *forms an ONB for $L^2([0, 1]^C)$.*

Generalized Fourier Series. For a vector-valued function $\boldsymbol{s}(\boldsymbol{x}) = [s_1(\boldsymbol{x}), \ldots, s_D(\boldsymbol{x})]^\top$ with $s_d \in L^2([0, 1]^C)$ for $d \in \{1, \ldots, D\}$, the partial sum of the *generalized Fourier series (GFS)* [36] using the ONB $\{\phi_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{N}_0^C}$ (defined in (2)) is defined by

$$\boldsymbol{s}_K(\boldsymbol{x}) \triangleq \sum_{\boldsymbol{k}\in\mathbb{K}} \boldsymbol{w}^{(\boldsymbol{k})}\phi_{\boldsymbol{k}}(\boldsymbol{x}), \tag{3}$$

where $\mathbb{K} = \{0, \ldots, K_1 - 1\} \times \cdots \times \{0, \ldots, K_C - 1\}$ is a finite truncation of all multi-indices, set by the *spectral resolution* $K = (K_1, \ldots, K_C)$, the number of basis functions per axis, and the coefficient vector $\boldsymbol{w}^{(\boldsymbol{k})} = [w_1^{(\boldsymbol{k})}, \ldots, w_D^{(\boldsymbol{k})}]^\top \in \mathbb{R}^D$ has components obtained via inner products:

$$w_d^{(\boldsymbol{k})} = \langle s_d, \phi_{\boldsymbol{k}}\rangle_{L^2} = \int_{[0,1]^C} s_d(\boldsymbol{x})\phi_{\boldsymbol{k}}(\boldsymbol{x})\, d\boldsymbol{x}. \tag{4}$$

Theorem 2 (GFS Convergence; proof in App. D)**.** *The partial sum $\boldsymbol{s}_K$ converges to $\boldsymbol{s}$ in $L^2([0, 1]^C; \mathbb{R}^D)$,* i.e.*, $\|s_d - s_{d,K}\|_{L^2} \to 0$ as $\min_c K_c \to \infty$ for all $d \in \{1, \ldots, D\}$.*

Tensor Representation. The multi-dimensional basis functions in (2) can be expressed as elements of the tensor product (defined in App. B.2):

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \boldsymbol{\varphi}(x_1) \otimes \boldsymbol{\varphi}(x_2) \otimes \cdots \otimes \boldsymbol{\varphi}(x_C), \tag{5}$$

where $\boldsymbol{\Phi}(\boldsymbol{x}) \in \mathbb{R}^{K_1\times\cdots\times K_C}$ is a *rank-1 Cth-order tensor*, and $\boldsymbol{\varphi}(x_c) = [\varphi_k(x_c)]_{k=0}^{K_c-1} \in \mathbb{R}^{K_c}$ with $\{\varphi_k\}_{k\in\mathbb{N}_0}$ being the one-dimensional basis from (1).

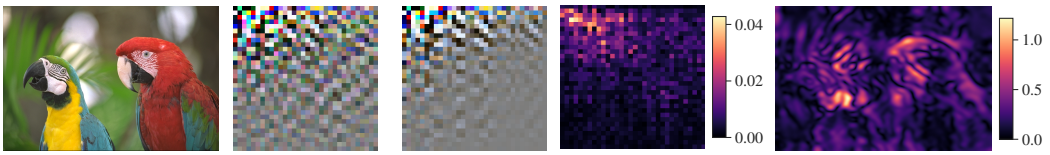
Now let $\boldsymbol{\mathcal{W}} = [w_d^{(\boldsymbol{k})}] \in \mathbb{R}^{K_1\times\cdots\times K_C\times D}$ denote the coefficient tensor of order $C + 1$, where $w_d^{(\boldsymbol{k})}$ is the coefficient for basis function $\phi_{\boldsymbol{k}}$ and output dimension d, as defined in (4). Using the generalized dot product $\bullet$, which contracts all modes of its first argument with the leading modes of its second (Definition 12), the GFS in (3) can be compactly expressed as

$$\boldsymbol{s}_K(\boldsymbol{x}) = \boldsymbol{\Phi}(\boldsymbol{x}) \bullet \boldsymbol{\mathcal{W}}. \tag{6}$$

4 Methods

4.1 Fourier Tensor Network (FUTON)

While the GFS provides a theoretically sound foundation, its computational complexity poses a challenge. The tensor contraction in (6) requires $\mathcal{O}(D\prod_c K_c)$ operations, scaling exponentially with input dimension C.



(a) $\boldsymbol{s}$ (image) (b) $\boldsymbol{\mathcal{W}}^\star$ (full rank) (c) $\boldsymbol{\mathcal{W}}$ ($R = 8$) (d) $\|\boldsymbol{w}^{\star(\boldsymbol{k})} - \boldsymbol{w}^{(\boldsymbol{k})}\|$ (e) $\|\boldsymbol{s}_K(\boldsymbol{x}) - \hat{\boldsymbol{s}}(\boldsymbol{x})\|$

Figure 2: **FUTON's coefficient tensor.** (a) A Kodak image and its coefficient tensor in the cosine basis at $K_c = 32$, visualized as RGB images with the lowest frequencies at the top left: (b) in full, 3,072 coefficients (23.79 dB), and (c) at rank $R = 8$, the certified FUTON of Prop. 1, built without training, 536 parameters (19.89 dB). (d, e) The norm of their difference per frequency, and per pixel between the images they represent.

To address this computational bottleneck, we *reparameterize* the coefficient tensor $\boldsymbol{\mathcal{W}}$ using *low-rank tensor decomposition*, yielding our *Fourier Tensor Network* (FUTON) model. Specifically, we employ the Canonical Polyadic (CP) decomposition format [13], which represents $\boldsymbol{\mathcal{W}}$ as a sum of R rank-1 tensors:

$$\boldsymbol{\mathcal{W}} = \sum_{r=1}^{R} \boldsymbol{u}_r^{(1)} \otimes \boldsymbol{u}_r^{(2)} \otimes \cdots \otimes \boldsymbol{u}_r^{(C)} \otimes \boldsymbol{v}_r, \tag{7}$$

where $\boldsymbol{u}_r^{(c)} \in \mathbb{R}^{K_c}$, $\boldsymbol{v}_r \in \mathbb{R}^D$, and R is the CP *rank*. We define factor matrices $\boldsymbol{U}^{(c)} = [\boldsymbol{u}_1^{(c)} \mid \cdots \mid \boldsymbol{u}_R^{(c)}] \in \mathbb{R}^{K_c \times R}$ for each input dimension c, and $\boldsymbol{V} = [\boldsymbol{v}_1 \mid \cdots \mid \boldsymbol{v}_R] \in \mathbb{R}^{D \times R}$ for the output. For a Kodak image at $K_c = 32$ (Fig. 2), rank $R = 8$ keeps 89% of the energy of $\boldsymbol{\mathcal{W}}^\star \in \mathbb{R}^{32 \times 32 \times 3}$ while reducing parameters from 3,072 to 536. This highlights the inherent low-rank structure of natural signals in the Fourier domain, which FUTON exploits for efficient and compact representation.

Substituting (5) and (7) into (6), and subsequently rearranging the summations, yields the following compact form, which Fig. 10 derives in tensor diagram notation:

$$\hat{\boldsymbol{s}}(\boldsymbol{x}) = \boldsymbol{V}\boldsymbol{g}(\boldsymbol{x}), \tag{8}$$

$$\boldsymbol{g}(\boldsymbol{x}) \triangleq \boldsymbol{h}_1(x_1) \odot \boldsymbol{h}_2(x_2) \odot \cdots \odot \boldsymbol{h}_C(x_C), \tag{9}$$

$$\boldsymbol{h}_c(x_c) \triangleq \boldsymbol{U}^{(c)\top} \boldsymbol{\varphi}(x_c), \quad c \in \{1, \ldots, C\} \tag{10}$$

where $\boldsymbol{h}_c(x_c) \in \mathbb{R}^R$ projects basis functions at x_c onto $\boldsymbol{U}^{(c)}$, and the *feature vector* $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^R$ is the Hadamard product of projected features.

Basis, Combiner, and Head. Equations (8)–(10) are the three stages of Fig. 1: a fixed *basis* expands each coordinate, a learned *combiner*, here CP, projects each expansion onto R components and multiplies them across axes, and a learned *head*, here $\boldsymbol{V}$, maps the feature vector to the signal. With this linear head, FUTON is a GFS whose coefficient tensor has CP rank at most R, which makes its error exactly analyzable (Sec. 4.3). A nonlinear head adds capacity: a one-hidden-layer MLP of width M,

$$\hat{\boldsymbol{s}}(\boldsymbol{x}) = \boldsymbol{\psi}\big(\boldsymbol{g}(\boldsymbol{x})\big), \qquad \boldsymbol{\psi}(\boldsymbol{g}) = \boldsymbol{V}_2 \,\mathrm{ReLU}(\boldsymbol{V}_1 \boldsymbol{g} + \boldsymbol{b}_1) + \boldsymbol{b}_2, \tag{11}$$

which contains every linear head when $M \geq 2D$, as $\mathrm{ReLU}(t) - \mathrm{ReLU}(-t) = t$, and adds 7.4 dB on images (App. I). An optional output activation, such as $\tanh$, completes the model.

Spectral Resolution and Rank. We set each axis's number of basis functions relative to its number of samples L_c, $K = (\kappa L_1, \ldots, \kappa L_C)$, with a *relative bandwidth* κ. At $\kappa = 1$, the cosines span every signal on the sampling grid, and a smaller κ keeps fewer, lower frequencies; $\kappa \in \{1/4, 1/2, 1\}$ work best in practice (Fig. 3b), and we use $\kappa = 1/2$. The rank then sets the size, $R \sum_c K_c$ parameters in the combiner plus RD (and D biases) for a linear head or $RM + M + MD + D$ for an MLP head: we take the largest R within a parameter budget.

Given observed coordinate–signal pairs $\{(\boldsymbol{x}_n, \boldsymbol{s}_n)\}_{n=1}^N$, the FUTON parameters $\boldsymbol{\theta} = (\boldsymbol{U}^{(1)}, \ldots, \boldsymbol{U}^{(C)}, \boldsymbol{\psi})$, with $\boldsymbol{\psi}$ the head, are learned by minimizing the reconstruction loss:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}\left(\boldsymbol{s}_n, \hat{\boldsymbol{s}}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\right), \tag{12}$$

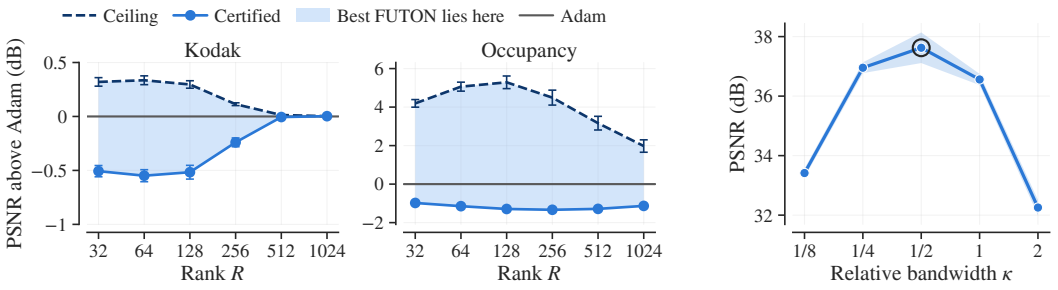


Figure 3: **The error bound and the bandwidth.** (a) PSNR relative to Adam's FUTON per rank at $\kappa = 1/2$, averaged over the 24 Kodak images and the five volumes, $\pm$ one standard error: the ceiling and the certified FUTON bracket the best FUTON. (b) Mean PSNR on Kodak against the relative bandwidth κ at 197k parameters, the rank adjusted to keep the size, the default circled.

where $\mathcal{L}$ is a loss function, *e.g.*, mean squared error. Factor matrices are initialized from a uniform distribution (Kaiming-uniform with fan-in K_c) and optimized via gradient-based methods (*e.g.*, Adam [38]). FUTON is a universal approximator (Thm. 3): it can approximate any function in $L^2([0,1]^C;\mathbb{R}^D)$ to arbitrary precision with a sufficiently large spectral resolution K and rank R.

Theorem 3 (Universal Approximation; proof in App. E)**.** *For any $\varepsilon > 0$ and vector-valued function $\boldsymbol{s} \in L^2([0,1]^C;\mathbb{R}^D)$, there exist $K \in \mathbb{N}$ (the same on every axis), rank $R \le K^{C-1}\min\{K, D\}$, and factor matrices $(\boldsymbol{U}^{(c)})_{c=1}^{C}$, $\boldsymbol{V}$ such that $\|s_d - \hat{s}_d\|_{L^2} < \varepsilon$ for all $d \in \{1,\dots,D\}$.*

Universality alone does not distinguish FUTON from MLPs [39, 40]; it certifies that the truncated low-rank parameterization loses no expressivity, also with an MLP head ($M \ge 2D$) or, for targets $\tanh(\tilde{\boldsymbol{s}})$, a $\tanh$ output (Rem. 1).

4.2 Computational Complexity

A key advantage of FUTON is its low time complexity with respect to the input and output dimensions. In the optimal contraction order (Fig. 10 and App. B.3), the forward pass projects each basis vector $\boldsymbol{\varphi}(x_c)$ onto the factor matrix $\boldsymbol{U}^{(c)}$ (10), in $\mathcal{O}(R\sum_c K_c)$ operations, multiplies the projections (9), in $\mathcal{O}(CR)$, and maps the feature vector $\boldsymbol{g}(\boldsymbol{x})$ to the output (8), in $\mathcal{O}(DR)$, or $\mathcal{O}(MR + MD)$ for an MLP head of width M. The total per coordinate, $\mathcal{O}(R\sum_c K_c + CR + DR)$ with a linear head, is linear in each K_c, in R, and in C and D separately.

Cached Basis and Shared Projections. The basis has no parameters, so on a sampling grid FUTON evaluates it once, when the model is built, and caches an $L_c \times K_c$ table per axis: training on grid samples never evaluates the basis again, and inference on the cached grid is particularly fast. Each $\boldsymbol{h}_c$ depends on one coordinate, so the projections of the table's rows are shared by all points with that coordinate: on a grid of $N = \prod_c L_c$ points, the combiner costs $\mathcal{O}(R\sum_c L_c K_c + NCR)$ instead of $\mathcal{O}(NR\sum_c K_c)$. For a 768×512 image at $\kappa = 1/2$ and $R = 280$, that is under 10^3 operations per pixel instead of 1.8×10^5, so the head dominates. Off the grid, as along a radiance field's rays, the basis is evaluated per point.

4.3 Error Analysis

In practice, the question is what a FUTON of resolution K and rank R loses on a given signal, before training; for the linear head, the answer is exact. Let $\boldsymbol{\mathcal{W}}^\star \in \mathbb{R}^{K_1 \times \cdots \times K_C \times D}$ hold the signal's coefficients (4), $\|\boldsymbol{f}\|^2 = \sum_d \|f_d\|_{L^2}^2$, $\varepsilon_R(\boldsymbol{\mathcal{A}}) = \inf\{\|\boldsymbol{\mathcal{A}} - \boldsymbol{\mathcal{B}}\|_F : \operatorname{rank}\boldsymbol{\mathcal{B}} \le R\}$ be the best rank-R error of a tensor $\boldsymbol{\mathcal{A}}$, and $\boldsymbol{\mathcal{A}}_{\langle\mathbb{S}\rangle}$ be its matricization with the modes in $\mathbb{S}$ as rows.

Proposition 1 (Error of FUTON with a linear head)**.** *Let $\boldsymbol{s} \in L^2([0,1]^C;\mathbb{R}^D)$, $K = (K_1,\dots,K_C)$, $R \ge 1$, and $\hat{\boldsymbol{s}}_{\boldsymbol{\theta}}$ the model* (8) *with parameters $\boldsymbol{\theta} = (\boldsymbol{U}^{(1)},\dots,\boldsymbol{U}^{(C)},\boldsymbol{V})$.*

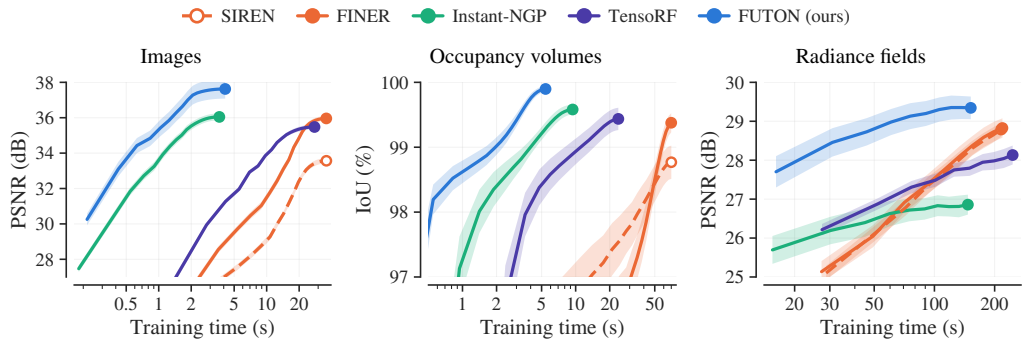


Figure 4: **Training speed.** Mean quality during training against median training time, with one standard error, for the featured models; radiance fields on validation views. Figures 13 and 15 plot every model's final quality against its costs.

Table 1: **Images**: quality averaged over the 24 Kodak images, and median training time, inference throughput, and peak GPU memory, at about 197k parameters. Best **bold**, second underlined, among the models shown; Tab. 11 adds standard errors and the other three baselines.

	Quality			Training		Inference	
Model	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	Time (s) ↓	Memory (GB) ↓	Throughput (MPix/s) ↑	Memory (MB) ↓
RFF [18]	31.28	0.846	0.271	29.8	2.08	30.7	1208
MFN [20]	36.72	0.938	0.129	57.3	7.67	26.8	3224
SIREN [6]	33.57	0.906	0.192	35.7	3.25	34.4	805
FreSh (SIREN) [21]	34.88	0.922	0.150	39.3	3.25	34.5	805
FINER [19]	35.96	0.938	0.120	35.7	3.25	34.3	805
Instant-NGP [10]	36.05	0.932	0.142	**3.6**	0.72	198.3	503
TensoRF [11]	35.48	0.920	0.161	27.8	1.79	55.2	805
GA-Planes [24]	32.23	0.878	0.227	19.0	1.32	68.1	805
F-INR (SIREN) [31]	29.35	0.786	0.343	7.1	2.70	136.3	1338
FUTON (ours)	**37.63**	**0.955**	**0.098**	4.1	**0.36**	**228.7**	**201**

1. Every FUTON. *For every $\boldsymbol{\theta}$, the error is a truncation plus a CP misfit:* $\|\boldsymbol{s} - \hat{\boldsymbol{s}}_{\boldsymbol{\theta}}\|^2 = \|\boldsymbol{s} - \boldsymbol{s}_K\|^2 + \left\|\boldsymbol{\mathcal{W}}^\star - [\![\boldsymbol{U}^{(1)}, \ldots, \boldsymbol{U}^{(C)}, \boldsymbol{V}]\!]\right\|_F^2$.
2. The best FUTON. $\inf_{\boldsymbol{\theta}} \|\boldsymbol{s} - \hat{\boldsymbol{s}}_{\boldsymbol{\theta}}\|^2 = \|\boldsymbol{s} - \boldsymbol{s}_K\|^2 + \varepsilon_R(\boldsymbol{\mathcal{W}}^\star)^2$.
3. Computable bounds. $\underline{\varepsilon}_R \le \varepsilon_R(\boldsymbol{\mathcal{W}}^\star) \le \overline{\varepsilon}_R$, *where* $\underline{\varepsilon}_R^2 = \max_{\mathbb{S}} \sum_{i>R} \sigma_i^2\left(\boldsymbol{\mathcal{W}}^\star_{\langle \mathbb{S} \rangle}\right)$ *over the nonempty proper subsets* $\mathbb{S}$ *of the* $C+1$ *modes, and* $\overline{\varepsilon}_R$ *is the misfit of an explicit rank-*R *tensor built from* $\boldsymbol{\mathcal{W}}^\star$ *by nested SVDs; the FUTON with its factors, the* certified FUTON*, has squared error exactly* $\|\boldsymbol{s} - \boldsymbol{s}_K\|^2 + \overline{\varepsilon}_R^2$.

Appendix F proves it for any basis and on sampling grids. For a signal in $[-1, 1]$, the proposition gives three PSNRs at each (K, R): the *truncation* PSNR, which no FUTON of resolution K exceeds; the *ceiling*, from $\underline{\varepsilon}_R$, which no FUTON of that size exceeds; and the *certified* PSNR, from $\overline{\varepsilon}_R$, attained without training, with the best FUTON between the last two. They take one pass over the samples and SVDs (App. F.2), on average 0.02 to 3.0 s per signal for six ranks at once, 16 to 860 times less than training. On the Kodak images and the volumes, in 522 runs over κ and R (App. F.3), Adam's FUTON always lands between the certified FUTON, to within 0.02 dB, and the ceiling (Fig. 3): on images, the bracket is a median 0.05 dB wide, so the bounds predict what training reaches, and on volumes, where the ceiling is loose, the certified FUTON still guarantees a PSNR before training.

The bracket also sizes FUTON before training: for a target PSNR, the smallest FUTON whose ceiling reaches it and the smallest whose certified PSNR reaches it enclose the smallest one Adam trains to it, on every signal and target; on Kodak, the two coincide for 56% of images and targets, and the certified choice matches Adam's for 76% (App. F.3). Appendix F.1 bounds the truncation by the

Table 2: **Occupancy volumes**: IoU on each shape and the mean, and median costs, marked as in Tab. 1, at about 131k parameters. Table 12 adds standard errors and the other baselines.

	IoU (%) ↑						Training		Inference	
Model	Armadillo	Dragon	Happy Buddha	Lucy	Thai Statue	Mean	Time (s) ↓	Memory (GB) ↓	Throughput (MVox/s) ↑	Memory (GB) ↓
RFF	99.71	99.65	99.59	99.78	99.28	99.60	46.5	4.15	48.9	3.33
MFN	99.23	97.47	97.50	99.10	98.74	98.41	108.5	15.07	38.9	5.49
SIREN	98.78	99.18	99.41	98.53	97.94	98.77	70.0	6.48	51.0	2.26
FreSh (SIREN)	99.34	99.67	99.65	99.15	98.83	99.33	263.6	6.48	51.3	2.26
FINER	99.45	99.59	99.70	99.29	98.85	99.38	69.9	6.48	51.3	2.26
Instant-NGP	99.50	99.59	99.69	99.78	99.35	99.58	9.4	2.11	162.6	2.93
TensoRF	99.53	99.71	99.78	99.34	98.82	99.44	23.8	1.75	202.3	1.19
GA-Planes	99.58	99.79	99.93	99.79	99.34	99.69	62.2	2.31	111.9	1.37
F-INR (SIREN)	97.75	98.25	98.08	96.27	96.42	97.35	20.9	6.59	179.2	3.39
FUTON (ours)	**99.86**	**99.94**	**99.99**	**99.96**	**99.74**	**99.90**	**5.4**	**0.28**	**537.4**	**0.61**

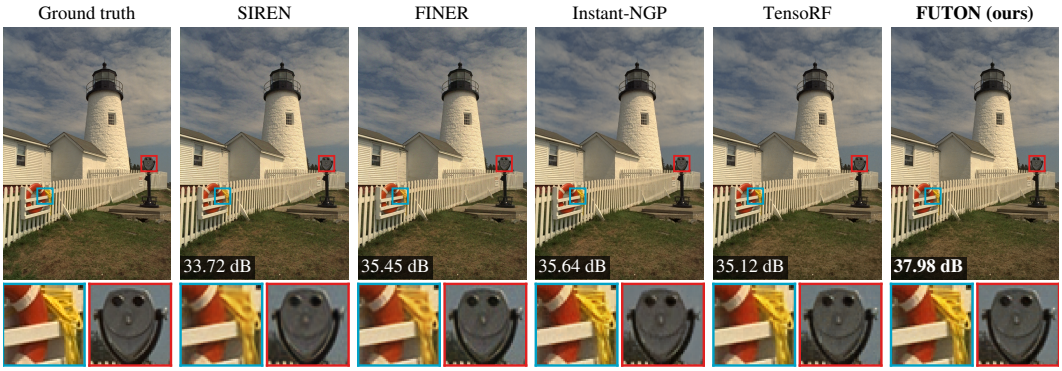


Figure 5: **Image fitting** on kodim19, with PSNR and two regions magnified; Fig. 14 shows more.

signal's smoothness, $\|\boldsymbol{s} - \boldsymbol{s}_K\|^2 \leq \sum_c \|\partial_c \boldsymbol{s}\|^2 / (\pi K_c)^2$ for the cosine basis (Cor. 1), and proves a low-rank sufficiency theorem.

5 Experiments

We compare FUTON with twelve INRs on image fitting, occupancy volumes, and novel view synthesis: MLPs on Fourier features (RFF) or positional encodings (PE-MLP), MFN, MLPs with specialized activations (SIREN, FreSh, Gauss, WIRE, and FINER), the hybrid grid models Instant-NGP, TensoRF, and GA-Planes, and F-INR. The main tables omit PE-MLP, Gauss, and WIRE, which never rank first or second in a task's mean quality or costs (App. H reports all twelve).

On each task, every baseline is the smallest of its kind above a common parameter budget, and FUTON's rank keeps it below all of them. Every model trains with Adam [38] at its best learning rate from one grid, under a cosine schedule, on one NVIDIA A100 GPU (App. G). Quality, PSNR, SSIM [41], LPIPS [42], or IoU, is a mean over the signals; training time, inference speed on whole signals, and peak GPU memory are medians. FUTON uses the cosine basis (1) throughout.

5.1 Image Fitting

On the 24 Kodak images [43], fitted whole at every step for 1,000 steps at about 197k parameters, FUTON ($\kappa = 1/2$, $R = 280$, MLP head of width $M = 64$) reaches 37.63 dB PSNR on average (Tab. 1), 0.9 dB above MFN, the strongest baseline, with the best SSIM and LPIPS, and is the most accurate model on 21 of the 24 images, kodim19 among them (Fig. 5).

FUTON converges fastest (Fig. 4), helped by being shallow, with one layer of factor matrices and a one-hidden-layer head: it passes 35 dB in 0.9 s, against 1.6 s for Instant-NGP and 14.6 s for MFN, and

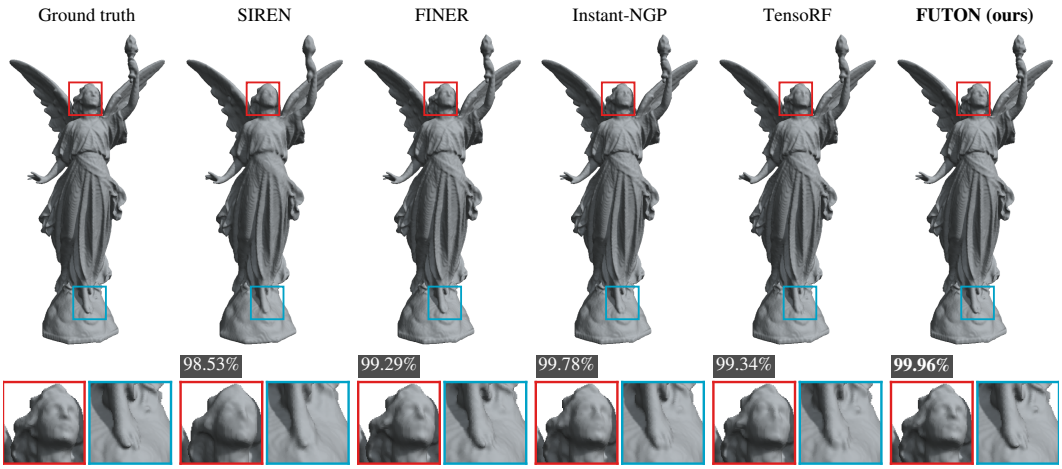


Figure 6: **Occupancy volumes** on Lucy, with IoU and two regions magnified; Fig. 16 shows more.

Table 3: **Radiance fields**: PSNR on each Blender scene's test views and the mean, and median costs, marked as in Tab. 1, speed in frames per second, at about 74k parameters. Table 13 adds SSIM, LPIPS, standard errors, and the other baselines.

	PSNR (dB) ↑									Training		Inference	
Model	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean	Time (s) ↓	Mem. (GB) ↓	Speed (FPS) ↑	Mem. (GB) ↓
RFF	33.10	22.91	27.47	32.76	28.48	25.62	33.37	20.88	28.07	179.3	2.91	38.7	2.12
MFN	32.94	23.77	27.28	32.56	28.52	25.60	33.04	22.50	28.28	275.6	9.97	35.5	3.61
SIREN	32.54	24.58	27.61	32.40	29.03	26.53	33.25	22.29	28.53	215.5	5.08	38.4	2.14
FreSh (SIREN)	32.38	24.19	27.60	32.81	29.62	25.18	32.58	22.32	28.34	217.2	5.03	39.4	2.13
FINER	32.85	**24.61**	**27.74**	32.30	29.46	26.11	**33.58**	22.27	28.62	218.8	5.02	39.1	2.15
Instant-NGP	31.35	23.03	22.64	30.87	27.29	24.57	31.92	21.16	26.60	**147.0**	2.80	44.0	3.26
TensoRF	31.60	23.70	25.74	32.73	28.60	26.31	32.43	22.78	27.99	246.2	2.14	47.2	**2.04**
GA-Planes	32.58	23.75	25.77	33.06	29.63	26.09	32.29	22.79	28.24	271.6	2.10	45.7	**2.04**
F-INR (SIREN)	32.81	21.42	24.23	32.39	29.22	27.37	31.78	20.93	27.52	292.8	16.18	33.7	6.69
FUTON (ours)	**33.18**	23.86	27.70	**33.20**	**31.28**	**27.65**	33.40	**23.48**	**29.22**	152.0	**1.88**	**49.6**	**2.04**

reaches MFN's final PSNR, which MFN needs 57 s to reach, in 1.7 s (Tab. 10). Its 4.1 s of training are second only to Instant-NGP's 3.6 s, whose final PSNR FUTON reaches in 1.3 s. FUTON also infers fastest and uses the least memory, half of Instant-NGP's in training and 40% in inference: on the pixel grid, its cached basis and shared projections leave only the products across axes and the head to compute per pixel (Sec. 4.2), and, being shallow, it keeps few activations in training.

5.2 Occupancy Volumes

Five Stanford scans [44], normalized as in BACON [25], are labeled ± 1 at each voxel center at 512 voxels per unit length, and fitted from 2^{20} random voxels per step for 1,000 steps at about 131k parameters. With just a linear head ($\kappa = 1/2$, $R = 215$ to 310), FUTON has the best IoU on every shape (Tab. 2 and Fig. 6), 99.90% on average, against 99.69% for GA-Planes, the strongest baseline, and at most 99.60% for the MLP-based INRs. Its margin is largest on Thai Statue, 0.39 points, the shape on which most models score lowest. FUTON trains fastest, reaching 99.5% IoU in 3.2 s against 7.2 s for Instant-NGP and 44.5 s for GA-Planes, infers 2.7 times faster than TensoRF, the next fastest, and uses the least memory: a sixth of TensoRF's in training and half in inference.

5.3 Novel View Synthesis

On the eight Blender scenes [3], under FINER's protocol [19], the compared INR is a radiance field's density network, with a shared color network, at about 74k parameters, trained for 25,000 steps;

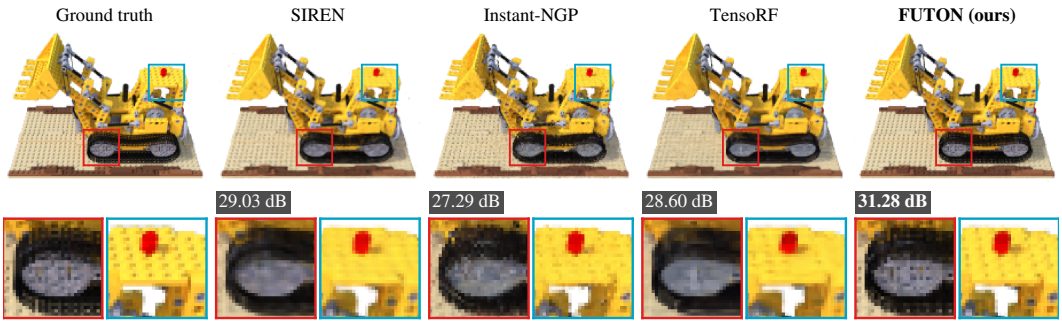


Figure 7: **Novel view synthesis** on Lego, with mean test-view PSNR and two regions magnified; Fig. 17 shows more.

FUTON has $K_c = 100$, $R = 180$, and an MLP head of width $M = 64$. It achieves the best mean PSNR (Tab. 3), 29.22 dB against 28.62 dB for FINER, with the best mean SSIM and LPIPS, 0.943 and 0.065 against 0.936 and 0.071 for GA-Planes (Tab. 13), and the best PSNR on five of the eight scenes, Lego by 1.65 dB (Fig. 7); FINER leads on Drums by 0.75 dB and on Ficus and Mic by under 0.2 dB.

FUTON trains in 152 s, close to Instant-NGP's 147 s, reaches FINER's final validation PSNR 4.2 times sooner, and trains in the least memory. It renders fastest too, but by little (49.6 against 47.2 FPS): along rays, samples lie off the grid, so the basis is evaluated per point, and the shared renderer keeps every model between 33.7 and 49.6 FPS.

5.4 Ablations

Five ablations on Kodak (App. I) motivate FUTON's defaults. The relative bandwidth trades truncation against rank at a fixed size (Fig. 3b): $\kappa = 1/8$ truncates the images and $\kappa = 2$ leaves too small a rank, 4.2 and 5.4 dB below $\kappa = 1/2$, while $\kappa = 1/4$ and 1 stay within 1.1 dB; training time falls as κ grows, with the rank.

The MLP head adds 7.4 dB over a linear head, and wider heads at most 0.5 dB, at 2–3 times the training time; at about equal size, CP leads a tensor ring and Tucker by 0.4 and 2.5 dB, though Tucker trains in half the time, and cosines lead five other bases by 0.2 to 1.6 dB. PSNR grows by 3.4 to 5.0 dB per doubling of R, with no plateau up to $R = 1{,}120$, while training time grows from 2.2 to 10.9 s: a smooth trade of size for quality.

6 Conclusion

We presented FUTON, a structured INR that unifies generalized Fourier series with a low-rank CP decomposition of the coefficient tensor, read out by a linear or MLP head. This design addresses common limitations of MLP-based INRs (slow convergence, an opaque size–accuracy relation): its cost is linear in each axis's resolution and in the rank, and with a linear head its best error splits exactly into terms bounded before training, bounds that also size it for a target quality. Against twelve INRs of matched size on three tasks, FUTON delivered the best average quality, reached every quality target first, inferred fastest, and trained in the least memory. These findings position FUTON as a practical alternative to MLPs for signal representation.

Limitations. FUTON's separable basis favors axis-aligned structure: an oblique edge needs a rank that grows with its sharpness (Tab. 4). Signals whose coefficients lack low-rank structure, such as noise, need a large rank, which Prop. 1 reveals before training. The exact error analysis covers the linear head on fully sampled signals: only its upper bound transfers to the MLP head, radiance fields lie outside it, and on volumes its bracket is wide (App. J).

Acknowledgments and Disclosure of Funding

This work is funded by the Flanders AI Research Program, under grant no. 174B09119. The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO, and the Flemish Government – department EWI. The authors declare no competing interests. Pourya Behmandpoor and Nikos Deligiannis acknowledge the support of the Innoviris WatchBOT Project (Project 2025-JRDIC-3b) and the Francqui Research Professorship (N. Deligiannis). They also acknowledge the support of the ERC Consolidator Grant IONIAN (No. 101171240, DOI: 10.3030/101171240). Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, pages 165–174, 2019.

[2] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 3789–3799, 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. In *ECCV*, pages 405–421, 2020.

[4] Liyue Shen, John Pauly, and Lei Xing. NeRP: Implicit neural representation learning with prior embedding for sparsely sampled image reconstruction. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):770–782, 2024.

[5] Ruyi Zha, Yanhao Zhang, and Hongdong Li. NAF: Neural attenuation fields for sparse-view CBCT reconstruction. In *MICCAI*, pages 442–452, 2022.

[6] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, volume 33, pages 7462–7473, 2020.

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

[8] Vishwanath Saragadam, Daniel LeJeune, Jasper Tan, Guha Balakrishnan, Ashok Veeraraghavan, and Richard G. Baraniuk. WIRE: Wavelet implicit neural representations. In *CVPR*, pages 18507–18516, 2023.

[9] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 5301–5310, 2019.

[10] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics*, 41(4):102:1–102:15, 2022.

[11] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. TensoRF: Tensorial radiance fields. In *ECCV*, pages 333–350, 2022.

[12] Sara Fridovich-Keil, Giacomo Meanti, Frederik Rahbæk Warburg, Benjamin Recht, and Angjoo Kanazawa. K-Planes: Explicit radiance fields in space, time, and appearance. In *CVPR*, pages 12479–12488, 2023.

[13] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3):455–500, 2009.

[14] Gregory Beylkin and Martin J. Mohlenkamp. Numerical operator calculus in higher dimensions. *Proceedings of the National Academy of Sciences*, 99(16):10246–10251, 2002.

[15] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. In *CVPR*, pages 10318–10327, 2021.

[16] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *ICCV*, pages 5865–5874, 2021.

[17] Eric R. Chan, Connor Z. Lin, Matthew A. Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J. Guibas, Jonathan Tremblay, Sameh Khamis, Tero Karras, and Gordon Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, pages 16123–16133, 2022.

[18] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, volume 33, pages 7537–7547, 2020.

[19] 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 *CVPR*, pages 2713–2722, 2024.

[20] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. Multiplicative filter networks. In *ICLR*, 2021.

[21] Adam Kania, Marko Mihajlovic, Sergey Prokudin, Jacek Tabor, and Przemysław Spurek. FreSh: Frequency shifting for accelerated neural representation learning. In *ICLR*, 2025.

[22] Taesun Yeom, Sangyoon Lee, and Jaeho Lee. Fast training of sinusoidal neural fields via scaling initialization. In *ICLR*, 2025.

[23] Haoan Feng, Diana Aldana, Tiago Novello, and Leila De Floriani. SASNet: Spatially-adaptive sinusoidal networks for INRs. In *CVPR*, pages 41964–41973, 2026.

[24] Irmak Sivgin, Sara Fridovich-Keil, Gordon Wetzstein, and Mert Pilanci. Geometric algebra planes: Convex implicit neural volumes. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pages 55828–55853, 2025.

[25] David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. BACON: Band-limited coordinate networks for multiscale scene representation. In *CVPR*, pages 16252–16262, 2022.

[26] Kexuan Shi, Xingyu Zhou, and Shuhang Gu. Improved implicit neural representation with Fourier reparameterized training. In *CVPR*, pages 25985–25994, 2024.

[27] J. Douglas Carroll and Jih-Jie Chang. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[28] Richard A. Harshman. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

[29] Ledyard R. Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[30] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[31] Sai Karthikeya Vemuri, Tim Büchner, and Joachim Denzler. F-INR: Functional tensor decomposition for implicit neural representations. In *WACV*, pages 6557–6568, 2026.

[32] Gregory Beylkin and Martin J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. *SIAM Journal on Scientific Computing*, 26(6):2133–2159, 2005.

[33] Nikos Kargas and Nicholas D. Sidiropoulos. Supervised learning and canonical decomposition of multivariate functions. *IEEE Transactions on Signal Processing*, 69:1097–1107, 2021.

[34] Frederiek Wesel and Kim Batselier. Large-scale learning with Fourier features and tensor decompositions. In *NeurIPS*, volume 34, pages 17543–17554, 2021.

[35] Daniele Bigoni, Allan P. Engsig-Karup, and Youssef M. Marzouk. Spectral tensor-train decomposition. *SIAM Journal on Scientific Computing*, 38(4):A2405–A2439, 2016.

[36] Gerald B. Folland. *Fourier Analysis and Its Applications*, volume 4 of *Pure and Applied Undergraduate Texts*. American Mathematical Society, Providence, RI, 2009. Originally published by Wadsworth & Brooks/Cole, 1992.

[37] Elias M. Stein and Rami Shakarchi. *Fourier Analysis: An Introduction*, volume 1 of *Princeton Lectures in Analysis*. Princeton University Press, Princeton, NJ, 2003.

[38] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[39] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

[40] Sho Sonoda and Noboru Murata. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*, 43(2):233–268, 2017.

[41] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[42] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.

[43] Eastman Kodak Company. Kodak lossless true color image suite. `https://r0k.us/graphics/kodak/`, 1993.

[44] Stanford Computer Graphics Laboratory. The Stanford 3D scanning repository. `https://graphics.stanford.edu/data/3Dscanrep/`.

[45] Christopher J. Hillar and Lek-Heng Lim. Most tensor problems are NP-hard. *Journal of the ACM*, 60(6):45:1–45:39, 2013.

[46] Vin de Silva and Lek-Heng Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.

[47] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[48] J. M. Landsberg. *Tensors: Geometry and Applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2012.

[49] Thomas Lickteig. Typical tensorial rank. *Linear Algebra and its Applications*, 69:95–120, 1985.

[50] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

[51] L. Mirsky. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1):50–59, 1960.

[52] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[53] Charles F. Van Loan and Nikos Pitsianis. Approximation with Kronecker products. In Marc S. Moonen, Gene H. Golub, and Bart L. R. De Moor, editors, *Linear Algebra for Large Scale and Real-Time Applications*, volume 232 of *NATO ASI Series E: Applied Sciences*, pages 293–314. Kluwer Academic Publishers, Dordrecht, 1993.

[54] Gizem Yüce, Guillermo Ortiz-Jiménez, Beril Besbinar, and Pascal Frossard. A structured dictionary perspective on implicit neural representations. In *CVPR*, pages 19228–19238, 2022.

[55] Jiaxiang Tang. torch-ngp: A PyTorch implementation of instant-ngp. `https://github.com/ashawkey/torch-ngp`, 2022.

[56] Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. NerfAcc: Efficient sampling accelerates NeRFs. In *ICCV*, pages 18537–18546, 2023.

[57] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, C. K. Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Shunting Zhang, Michael Suo, Phil Tillet, Xu Zhao, Eikan Wang, Keren Zhou, Richard Zou, Xiaodong Wang, Ajit Mathews, William Wen, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster machine learning through dynamic Python bytecode transformation and graph compilation. In *ASPLOS*, pages 929–947, 2024.

[58] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, volume 32, pages 8024–8035, 2019.

Supplementary Material

This supplementary material is organized as follows. Appendices A and B provide background on L^2 spaces and tensor networks, respectively, which underpin the theoretical framework of FUTON. Appendices C to E contain complete proofs of the theorems stated in the main paper. Appendix F proves the error analysis of Sec. 4.3 for any basis and on sampling grids, draws its consequences (a truncation rate, a low-rank sufficiency theorem, and an edge test), gives its algorithm, and reports its experiments. Appendix G provides detailed experimental configurations for reproducibility, App. H the full results of every task, and App. I the ablation studies. Finally, App. J discusses limitations.

A L^2 Space

FUTON's theoretical foundation relies on generalized Fourier series in L^2 spaces (Sec. 3.2). We provide formal definitions of the L^2 space and related concepts.

Definition 1 (L^2 Space)**.** *Let $\mathbb{X} \subseteq \mathbb{R}^C$ for some $C \in \mathbb{N}$. The Hilbert space $L^2(\mathbb{X})$ consists of **square-integrable** functions on $\mathbb{X}$, where $f \in L^2(\mathbb{X})$ if and only if*

$$\int_{\mathbb{X}} |f(\boldsymbol{x})|^2 \, d\boldsymbol{x} < \infty. \tag{13}$$

For instance, $L^2([0,1]^C)$ denotes the space of square-integrable functions on the C-dimensional unit hypercube.

Definition 2 (L^2 Inner Product)**.** *For $f, g \in L^2(\mathbb{X})$, the inner product is defined as*

$$\langle f, g \rangle_{L^2} \triangleq \int_{\mathbb{X}} f(\boldsymbol{x}) g(\boldsymbol{x}) \, d\boldsymbol{x}. \tag{14}$$

Definition 3 (L^2 Norm)**.** *For $f \in L^2(\mathbb{X})$, the norm is defined as*

$$\|f\|_{L^2} \triangleq \sqrt{\langle f, f \rangle_{L^2}}. \tag{15}$$

Definition 4 (L^2 Convergence)**.** *A sequence $(f_k)_{k \in \mathbb{N}_0}$ in $L^2(\mathbb{X})$ is said to converge in $L^2(\mathbb{X})$ to a function $f \in L^2(\mathbb{X})$ if $\lim_{k \to \infty} \|f_k - f\|_{L^2} = 0$, denoted as $f_k \xrightarrow{L^2} f$.*

Definition 5 (L^2 Orthonormality)**.** *A set of functions $\{\varphi_k\}_{k \in \mathbb{I}} \subseteq L^2(\mathbb{X})$ is **orthonormal** in $L^2(\mathbb{X})$ if $\langle \varphi_k, \varphi_\ell \rangle_{L^2} = \delta_{k\ell}$ for all $k, \ell \in \mathbb{I}$, where $\delta_{k\ell}$ is the Kronecker delta (1 if $k = \ell$, 0 otherwise).*

Definition 6 (L^2 Density)**.** *A subset $\mathbb{D} \subseteq L^2(\mathbb{X})$ is said to be **dense** in $L^2(\mathbb{X})$ if for every function $f \in L^2(\mathbb{X})$ there exists a sequence $(f_k)_{k \in \mathbb{N}_0}$ in $\mathbb{D}$ such that $f_k \xrightarrow{L^2} f$. In other words, any function in $L^2(\mathbb{X})$ can be approximated arbitrarily closely by functions from $\mathbb{D}$.*

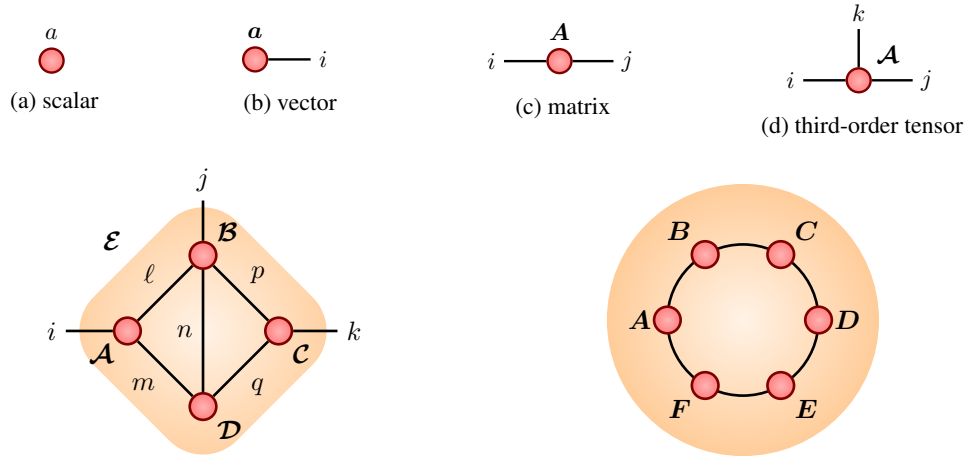
Definition 7 (L^2 Completeness)**.** *A set of functions is said to be **complete** in $L^2(\mathbb{X})$ if its **span** is **dense** in $L^2(\mathbb{X})$. Formally, $\{\varphi_k\}_{k \in \mathbb{I}} \subseteq L^2(\mathbb{X})$ is complete if for every function $f \in L^2(\mathbb{X})$ and every $\varepsilon > 0$, some finite linear combination g of the φ_k satisfies $\|f - g\|_{L^2} < \varepsilon$. In other words, any function in $L^2(\mathbb{X})$ can be approximated arbitrarily closely by a finite linear combination of functions from $\{\varphi_k\}_{k \in \mathbb{I}}$.*

Definition 8 (L^2 Orthonormal Basis)**.** *A set of functions $\{\varphi_k\}_{k \in \mathbb{N}_0} \subseteq L^2(\mathbb{X})$ is said to be an **orthonormal basis (ONB)** for $L^2(\mathbb{X})$ if it is both **orthonormal** and **complete** in $L^2(\mathbb{X})$.*

B Tensor Networks

B.1 Tensor Diagram Notation

A tensor is a multi-dimensional array of numbers. A Tth-order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_T}$ is a multi-dimensional array where the mth *mode* (dimension or index) has size I_m. Special cases include scalars ($T = 0$), vectors ($T = 1$), and matrices ($T = 2$).



(a) scalar (b) vector (c) matrix (d) third-order tensor

(e) $e_{ijk} = \sum_{\ell,m,n,p,q=1}^{I} a_{i\ell m} b_{j\ell np} c_{kpq} d_{mnq}$ (f) $\mathrm{tr}(\boldsymbol{ABCDEF})$

Figure 8: **Tensor diagram notation.** Graphical representation of tensors with different orders: (a) scalar (0th-order), (b) vector (1st-order), (c) matrix (2nd-order), and (d) third-order tensor. In tensor diagrams, tensors are represented as nodes (shapes) with edges (lines) denoting indices. Dangling edges represent open (uncontracted) indices. (e) Complex tensor network from (17) with multiple contracted indices. Edges connecting different nodes represent shared indices that are summed over. (f) Trace of matrix product $\mathrm{tr}(\boldsymbol{ABCDEF})$ represented as a ring diagram, where the cyclic property becomes visually apparent, *i.e.*, any rotation of the ring yields the same value.

Index Contraction. An *index contraction* sums over all values of shared indices between tensors. For example, matrix multiplication

$$c_{ij} = \sum_{k} a_{ik} b_{kj} \tag{16}$$

contracts the shared index k. More complex contractions involve multiple tensors and indices:

$$e_{ijk} = \sum_{\ell,m,n,p,q=1}^{I} a_{i\ell m} b_{j\ell np} c_{kpq} d_{mnq}. \tag{17}$$

Indices that remain after contraction (i, j, k) are called *open indices*. A *tensor network* is a set of tensors with indices contracted according to a specified pattern.

Tensor Diagrams. *Tensor diagrams* provide powerful graphical notation for tensor operations. As shown in Fig. 8, tensors are represented as nodes with edges corresponding to indices. Edges connecting tensors represent contracted indices, while dangling edges represent open indices. This notation simplifies understanding of complex tensor operations and is used for FUTON in Fig. 10.

Figure 8e illustrates tensor diagrams for the complex contraction in (17). As another example of complex calculations, the trace of the product of six matrices can be represented by a simple ring-like diagram as seen in Fig. 8f. From this tensor diagram, the cyclic property of the trace becomes clear. This is a simple example of why tensor diagrams are advantageous. Tensor diagrams make structural properties visually apparent and are particularly useful for understanding operations like those in FUTON (see Fig. 10).

Many matrix and tensor operations perform some form of tensor network contraction, and hence, can be represented graphically using tensor diagrams. Figure 9 illustrates the tensor diagrams of some basic operations.

B.2 Basic Tensor Operations

Definition 9 (Fiber)**.** *A mode-m fiber of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_T}$ is a vector obtained by fixing every index except the mth, i.e., $\boldsymbol{\mathcal{A}}_{i_1 \cdots i_{m-1} : i_{m+1} \cdots i_T}$ for fixed indices $i_j \in \{1, \ldots, I_j\}$ where $j \neq m$.*

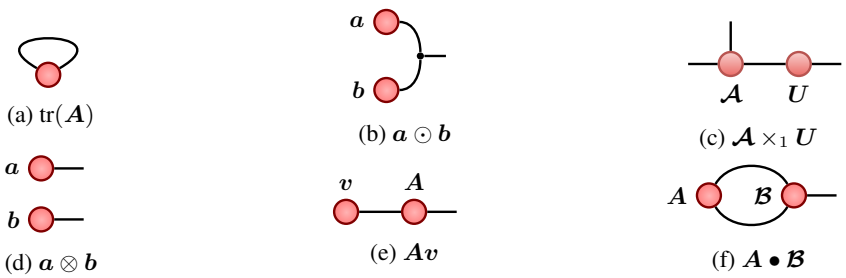


Figure 9: **Tensor diagrams of basic operations.** (a) matrix trace; (b) Hadamard product, used in (9); (c) mode-1 product; (d) tensor product, used in (5) and (7); (e) matrix-vector multiplication, used in (10) and (8); and (f) generalized dot product, used in (6).

Definition 10 (Matricization)**.** *The mode-m matricization (or unfolding) of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_T}$ is defined as the matrix $\boldsymbol{A}_{(m)} \in \mathbb{R}^{I_m \times (I_1 \cdots I_{m-1} I_{m+1} \cdots I_T)}$ obtained by stacking all mode-m fibers of $\boldsymbol{\mathcal{A}}$ along the columns, arranged in lexicographical order.*

Definition 11 (Hadamard Product)**.** *The Hadamard product $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \odot \boldsymbol{\mathcal{B}}$ of equally sized tensors $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times \cdots \times I_T}$ performs elementwise multiplication:*

$$c_{i_1 \cdots i_T} = a_{i_1 \cdots i_T} b_{i_1 \cdots i_T}. \tag{18}$$

Figure 9b illustrates the tensor diagram for the Hadamard product of two vectors.

Definition 12 (Generalized Dot Product)**.** *For tensors $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_T}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times \cdots \times I_T \times J_1 \times \cdots \times J_P}$, the* generalized dot product *$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \bullet \boldsymbol{\mathcal{B}} \in \mathbb{R}^{J_1 \times \cdots \times J_P}$ contracts over all T dimensions of $\boldsymbol{\mathcal{A}}$:*

$$c_{j_1, \ldots, j_P} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_T=1}^{I_T} a_{i_1, \ldots, i_T} \, b_{i_1, \ldots, i_T, j_1, \ldots, j_P}. \tag{19}$$

Definition 13 (Inner Product)**.** *The inner product (or dot product) of equally sized tensors $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{I_1 \times \cdots \times I_T}$ is*

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle \triangleq \sum_{i_1, \ldots, i_T} a_{i_1 \cdots i_T} b_{i_1 \cdots i_T}. \tag{20}$$

The inner product is a special case of the generalized dot product (Definition 12) where both tensors have identical shapes.

Definition 14 (Tensor Product)**.** *The tensor product $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \otimes \boldsymbol{\mathcal{B}}$ of tensors $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_T}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{J_1 \times \cdots \times J_P}$ yields $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{I_1 \times \cdots \times I_T \times J_1 \times \cdots \times J_P}$ with elements*

$$c_{i_1 \cdots i_T j_1 \cdots j_P} = a_{i_1 \cdots i_T} b_{j_1 \cdots j_P}. \tag{21}$$

Figure 9d illustrates the tensor diagram for the tensor product of two vectors.

Definition 15 (Tensor-Matrix Product)**.** *The mode-m product of a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_T}$ and a matrix $\boldsymbol{U} \in \mathbb{R}^{J \times I_m}$ is the tensor*

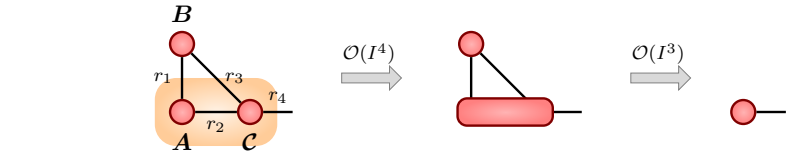
$$\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \times_m \boldsymbol{U} \in \mathbb{R}^{I_1 \times \cdots \times I_{m-1} \times J \times I_{m+1} \times \cdots \times I_T} \tag{22}$$

with elements

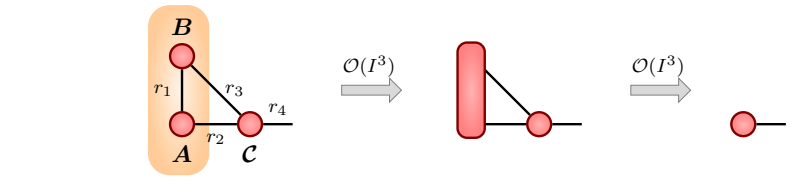
$$c_{i_1 \cdots i_{m-1} j i_{m+1} \cdots i_T} = \sum_{i_m=1}^{I_m} a_{i_1 \cdots i_T} U_{j i_m}. \tag{23}$$

Figure 9c illustrates the tensor diagram of the mode-1 product of a third-order tensor and a matrix.

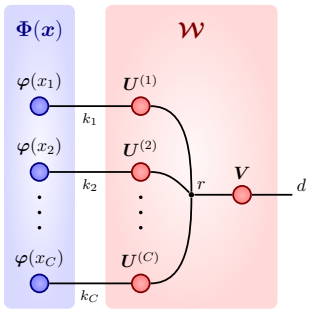
An equivalent matrix representation of the mode-m product is $\boldsymbol{C}_{(m)} = \boldsymbol{U}\boldsymbol{A}_{(m)}$, which allows us to employ established fast matrix multiplications when dealing with very large-scale tensors.



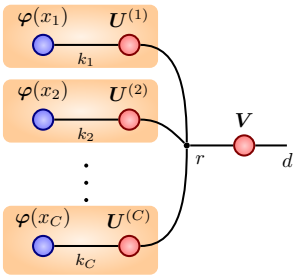
(a)



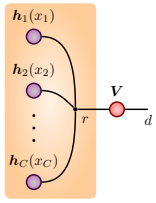
(b)



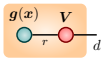
(c) FUTON's forward pass



(d) Matrix-vector products



(e) Hadamard products



(f) Matrix-vector product

Figure 10: **Contraction order, and FUTON's tensor network** (linear head). (a, b) Sequential contraction of a tensor network whose modes are all of size I. In (a), the matrix $\boldsymbol{A}$ and the tensor $\boldsymbol{\mathcal{C}}$ are first contracted (mode-m product), then the resulting tensor is contracted with the matrix $\boldsymbol{B}$, requiring $\mathcal{O}(I^4)$ operations in total. In (b), the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are first contracted (multiplied), then the resulting matrix is contracted with the tensor $\boldsymbol{\mathcal{C}}$, requiring $\mathcal{O}(I^3)$ operations in total. (c) Tensor diagram of FUTON's forward pass, where nodes represent tensors, connected edges represent contracted indices (summed over), and dangling edges correspond to open indices. See App. B.1 for tensor diagram notation. (d–f) Optimal contraction order achieving linear complexity $\mathcal{O}(R\sum_c K_c + CR + DR)$. Orange boxes indicate tensors being contracted at each step: (d) project each basis vector $\boldsymbol{\varphi}(x_c)$ onto its factor matrix $\boldsymbol{U}^{(c)}$ via matrix-vector multiplication (10), incurring $\mathcal{O}(R\sum_c K_c)$ operations; (e) combine the projected features $\boldsymbol{h}_c(x_c)$ via Hadamard product (9), incurring $\mathcal{O}(CR)$ operations; and (f) map the resulting vector $\boldsymbol{g}(\boldsymbol{x})$ to the output via matrix-vector multiplication with $\boldsymbol{V}$ (8), incurring $\mathcal{O}(DR)$ operations. This contrasts with the naive approach of directly computing $\boldsymbol{\Phi}(\boldsymbol{x}) \bullet \boldsymbol{\mathcal{W}}$, which requires $\mathcal{O}(D\prod_c K_c)$ operations and scales exponentially with the input dimension C.

B.3 Contraction Order

The computational cost of tensor network contractions depends critically on the order in which indices are contracted. Figure 10 illustrates this: contracting $\boldsymbol{A}$ and $\boldsymbol{\mathcal{C}}$ first (Fig. 10a) requires $\mathcal{O}(I^4)$ operations, while contracting $\boldsymbol{A}$ and $\boldsymbol{B}$ first (Fig. 10b) requires only $\mathcal{O}(I^3)$ operations. This principle is crucial for FUTON's linear complexity (see Sec. 4.2). Its panels (c)–(f) apply it to FUTON with a linear head.

B.4 Canonical Polyadic Decomposition

Definition 16 (Rank-1 Tensor)**.** *A rank-1 tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_T}$ is a tensor that can be expressed as the tensor product of nonzero vectors $\{\boldsymbol{u}^{(m)} \in \mathbb{R}^{I_m}\}_{m=1}^{T}$:*

$$\boldsymbol{\mathcal{A}} = \boldsymbol{u}^{(1)} \otimes \boldsymbol{u}^{(2)} \otimes \cdots \otimes \boldsymbol{u}^{(T)}. \tag{24}$$

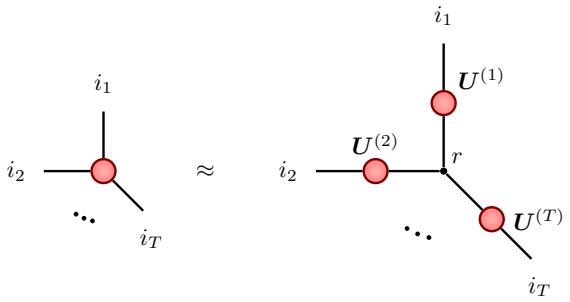


Figure 11: **Tensor diagram of CP decomposition.**

Equivalently, the elements of $\boldsymbol{\mathcal{A}}$ satisfy

$$a_{i_1 \cdots i_T} = u^{(1)}_{i_1} u^{(2)}_{i_2} \cdots u^{(T)}_{i_T}. \tag{25}$$

Definition 17 (Rank)**.** *The rank of a tensor $\boldsymbol{\mathcal{A}}$, denoted by $\operatorname{rank}(\boldsymbol{\mathcal{A}})$, is the minimal number of rank-1 tensors that sum to $\boldsymbol{\mathcal{A}}$.*

Definition 18 (CP Decomposition)**.** *Canonical polyadic (CP) decomposition [27, 28] represents a tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_T}$ as a sum of R rank-1 tensors:*

$$\boldsymbol{\mathcal{A}} = \sum_{r=1}^{R} \boldsymbol{u}^{(1)}_r \otimes \boldsymbol{u}^{(2)}_r \otimes \cdots \otimes \boldsymbol{u}^{(T)}_r, \tag{26}$$

where $\boldsymbol{u}^{(m)}_r \in \mathbb{R}^{I_m}$ are factor vectors, and R is the CP rank. Factor matrices $\boldsymbol{U}^{(m)} = [\boldsymbol{u}^{(m)}_1 \mid \cdots \mid \boldsymbol{u}^{(m)}_R] \in \mathbb{R}^{I_m \times R}$ collect the vectors for mode m.

Figure 11 shows the tensor diagram for CP decomposition, which can be understood by expressing (26) elementwise:

$$a_{i_1 \cdots i_T} = \sum_{r=1}^{R} \prod_{m=1}^{T} u^{(m)}_{i_m r}. \tag{27}$$

The CP decomposition (26) can be written in terms of matricization as:

$$\boldsymbol{A}_{(m)} = \boldsymbol{U}^{(m)} \boldsymbol{Y}^{(m)^\top}, \tag{28}$$

where $\boldsymbol{Y}^{(m)} = [\boldsymbol{y}^{(m)}_1 \mid \cdots \mid \boldsymbol{y}^{(m)}_R] \in \mathbb{R}^{(\prod_{j \neq m} I_j) \times R}$, and $\boldsymbol{y}^{(m)}_r = \operatorname{vec}\big(\bigotimes_{j \neq m} \boldsymbol{u}^{(j)}_r\big)$ for $r \in \{1, \ldots, R\}$.

It is important to distinguish between the *CP rank* R in (26) and the *(exact) rank* $\operatorname{rank}(\boldsymbol{\mathcal{A}})$ from Definition 17. The CP rank specifies the number of rank-1 components in a decomposition (26), exact or approximate, while the (exact) rank is the minimal number required for exact representation. When $R = \operatorname{rank}(\boldsymbol{\mathcal{A}})$ in (26), the result is called a *tensor rank decomposition* or *minimal CP decomposition*. However, determining the (exact) rank is NP-hard in general [45]. In practice, approximate low-rank CP decompositions are computed using iterative methods such as gradient-based optimization [13], where R is chosen based on computational constraints and approximation accuracy.

B.5 Tensor Networks as a Computational Tool

Low-rank tensor representations enable efficient computation with high-dimensional data. Storing a Tth-order tensor with each mode of size I requires I^T elements, which grows exponentially in T. In contrast, CP decomposition with rank R requires only $\mathcal{O}(TIR)$ parameters, reducing storage and computational costs dramatically.

Example 1 (Rank-1 Inner Product)**.** *Consider computing the inner product of two Tth-order tensors $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{I \times I \times \cdots \times I}$. Direct computation using* (20) *requires $\mathcal{O}(I^T)$ operations, which becomes*

intractable for large T. However, when $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$ are rank-1 tensors, i.e.*, $\boldsymbol{\mathcal{A}} = \boldsymbol{a}^{(1)} \otimes \boldsymbol{a}^{(2)} \otimes \cdots \otimes \boldsymbol{a}^{(T)}$ and $\boldsymbol{\mathcal{B}} = \boldsymbol{b}^{(1)} \otimes \boldsymbol{b}^{(2)} \otimes \cdots \otimes \boldsymbol{b}^{(T)}$, the inner product decomposes as*

$$\begin{aligned}
\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle &= \sum_{i_1,\ldots,i_T=1}^{I} a_{i_1 \cdots i_T} b_{i_1 \cdots i_T} \\
&= \sum_{i_1,\ldots,i_T=1}^{I} \left[\prod_{m=1}^{T} a_{i_m}^{(m)} \right] \left[\prod_{m=1}^{T} b_{i_m}^{(m)} \right] \\
&= \prod_{m=1}^{T} \left[\sum_{i_m=1}^{I} a_{i_m}^{(m)} b_{i_m}^{(m)} \right] \\
&= \prod_{m=1}^{T} \langle \boldsymbol{a}^{(m)}, \boldsymbol{b}^{(m)} \rangle.
\end{aligned}$$

*This requires only $\mathcal{O}(TI)$ operations by computing the mode-wise inner products $\langle \boldsymbol{a}^{(m)}, \boldsymbol{b}^{(m)} \rangle$ first, then multiplying the results—an exponential speedup. More generally, representing high-order tensors with low-rank tensor networks (*e.g.*, CP decomposition) and executing contractions in optimal order (as detailed in App. B.3) dramatically reduces computational complexity in numerous machine learning applications.*

C Proof of Thm. 1

Orthonormality. Let $\boldsymbol{k}, \boldsymbol{\ell} \in \mathbb{N}_0^C$. By Fubini's theorem and the separable definition of $\phi_{\boldsymbol{k}}$ in (2), the inner product decomposes over dimensions:

$$\begin{aligned}
\langle \phi_{\boldsymbol{k}}, \phi_{\boldsymbol{\ell}} \rangle_{L^2} &= \int_{[0,1]^C} \phi_{\boldsymbol{k}}(\boldsymbol{x}) \phi_{\boldsymbol{\ell}}(\boldsymbol{x}) \, d\boldsymbol{x} \\
&= \int_{[0,1]^C} \left[\prod_{c=1}^{C} \varphi_{k_c}(x_c) \right] \left[\prod_{c=1}^{C} \varphi_{\ell_c}(x_c) \right] d\boldsymbol{x} \\
&= \prod_{c=1}^{C} \int_0^1 \varphi_{k_c}(x_c) \varphi_{\ell_c}(x_c) \, dx_c \\
&= \prod_{c=1}^{C} \delta_{k_c \ell_c} = \delta_{\boldsymbol{k}\boldsymbol{\ell}}.
\end{aligned} \tag{29}$$

The penultimate equality follows from the orthonormality of the one-dimensional basis $\{\varphi_k\}_{k \in \mathbb{N}_0}$.

Completeness. We prove the completeness by the *orthogonal complement* criterion. Let $f \in L^2([0,1]^C)$ satisfy $\langle f, \phi_{\boldsymbol{k}} \rangle_{L^2} = 0$ for all $\boldsymbol{k} \in \mathbb{N}_0^C$. We show $f = 0$ almost everywhere (*i.e.*, $\|f\|_{L^2} = 0$).

We argue by induction on C. The case $C = 1$ holds because $\{\varphi_k\}_{k \in \mathbb{N}_0}$ is an ONB in $L^2([0,1])$ by assumption. Assume the claim holds for dimension $C - 1$. For $k_1 \in \mathbb{N}_0$, define

$$h_{k_1}(\boldsymbol{x}_{-1}) \triangleq \int_0^1 f(x_1, \boldsymbol{x}_{-1}) \, \varphi_{k_1}(x_1) \, dx_1, \tag{30}$$

where $\boldsymbol{x}_{-1} = (x_2, \ldots, x_C)$ is the vector of all coordinates except x_1. By the Cauchy–Schwarz inequality applied in the x_1-coordinate, for almost every $\boldsymbol{x}_{-1} \in [0,1]^{C-1}$,

$$\begin{aligned}
|h_{k_1}(\boldsymbol{x}_{-1})|^2 &= \left| \int_0^1 f(x_1, \boldsymbol{x}_{-1}) \, \varphi_{k_1}(x_1) \, dx_1 \right|^2 \\
&\leq \left(\int_0^1 |f(\boldsymbol{x})|^2 dx_1 \right) \left(\int_0^1 |\varphi_{k_1}(x_1)|^2 dx_1 \right) \\
&= \int_0^1 |f(\boldsymbol{x})|^2 dx_1,
\end{aligned} \tag{31}$$

since $\{\varphi_k\}_{k\in\mathbb{N}_0}$ is orthonormal in $L^2([0,1])$. Integrating over $\boldsymbol{x}_{-1}$ and applying Fubini's theorem yields

$$\begin{aligned}\|h_{k_1}\|_{L^2}^2 &= \int_{[0,1]^{C-1}} |h_{k_1}(\boldsymbol{x}_{-1})|^2 d\boldsymbol{x}_{-1} \\ &\leq \int_{[0,1]^C} |f(\boldsymbol{x})|^2 d\boldsymbol{x} < \infty, \end{aligned} \tag{32}$$

hence $h_{k_1} \in L^2([0,1]^{C-1})$. For any $\boldsymbol{k}_{-1} = (k_2, \ldots, k_C) \in \mathbb{N}_0^{C-1}$,

$$\begin{aligned} 0 &= \langle f, \phi_{(k_1,\boldsymbol{k}_{-1})} \rangle_{L^2} \\ &= \int_{[0,1]^{C-1}} h_{k_1}(\boldsymbol{x}_{-1}) \prod_{c=2}^{C} \varphi_{k_c}(x_c)\, d\boldsymbol{x}_{-1}. \end{aligned} \tag{33}$$

Hence h_{k_1} is orthogonal in $L^2([0,1]^{C-1})$ to the set $\big\{ \prod_{c=2}^{C} \varphi_{k_c} \big\}_{\boldsymbol{k}_{-1}\in\mathbb{N}_0^{C-1}}$, which is an ONB by the induction hypothesis; therefore $h_{k_1} = 0$ almost everywhere for every $k_1 \in \mathbb{N}_0$.

As the exceptional null sets, one per k_1, are countably many, their union is null, so for almost every $\boldsymbol{x}_{-1}$, the slice $x_1 \mapsto f(x_1, \boldsymbol{x}_{-1})$ has all one-dimensional Fourier coefficients zero:

$$\int_0^1 f(x_1, \boldsymbol{x}_{-1})\, \varphi_{k_1}(x_1)\, dx_1 = 0, \quad \forall\, k_1 \in \mathbb{N}_0. \tag{34}$$

By completeness of $\{\varphi_k\}_{k\in\mathbb{N}_0}$ in $L^2([0,1])$, it follows that $f(\cdot, \boldsymbol{x}_{-1}) = 0$ in $L^2([0,1])$ for almost every $\boldsymbol{x}_{-1}$, and thus $f = 0$ almost everywhere on $[0,1]^C$. □

D Proof of Thm. 2

It suffices to prove convergence for each component s_d independently. Fix $d \in \{1, \ldots, D\}$ and let $s_d \in L^2([0,1]^C)$ be arbitrary. By Thm. 1, the separable family $\{\phi_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{N}_0^C}$ from (2) forms an ONB for $L^2([0,1]^C)$.

For $K = (K_1, \ldots, K_C) \in \mathbb{N}^C$, recall $\mathbb{K} = \{0, \ldots, K_1 - 1\} \times \cdots \times \{0, \ldots, K_C - 1\}$ and define the finite-dimensional subspace

$$\mathbb{V}_K \triangleq \operatorname{span}\{\phi_{\boldsymbol{k}} : \boldsymbol{k} \in \mathbb{K}\}. \tag{35}$$

Clearly $\mathbb{V}_K \subseteq \mathbb{V}_{K'}$ whenever $K_c \leq K'_c$ for every c, and by completeness of the ONB, $\bigcup_K \mathbb{V}_K$ is dense in $L^2([0,1]^C)$.

Let $P_K \colon L^2([0,1]^C) \to \mathbb{V}_K$ denote the orthogonal projection onto $\mathbb{V}_K$. Since $\{\phi_{\boldsymbol{k}}\}_{\boldsymbol{k}\in\mathbb{K}}$ is an orthonormal set, the projection has the Fourier expansion

$$P_K(s_d) = \sum_{\boldsymbol{k}\in\mathbb{K}} \langle s_d, \phi_{\boldsymbol{k}} \rangle_{L^2} \phi_{\boldsymbol{k}}. \tag{36}$$

Comparing with (3) and (4), we identify $P_K(s_d) = s_{d,K}$.

To establish L^2 convergence, fix $\varepsilon > 0$. By density, there exists a finite linear combination $f = \sum_{\boldsymbol{k}\in\mathbb{J}} a_{\boldsymbol{k}} \phi_{\boldsymbol{k}}$ for some finite index set $\mathbb{J} \subset \mathbb{N}_0^C$ such that $\|s_d - f\|_{L^2} < \varepsilon/2$. Since $\mathbb{J}$ is finite, there exists $K_0 \in \mathbb{N}$ sufficiently large that $\mathbb{J} \subseteq \mathbb{K}$ whenever $\min_c K_c \geq K_0$; hence $f \in \mathbb{V}_K$ for all such K.

For $\min_c K_c \geq K_0$, the best approximation property of orthogonal projections (*i.e.*, $P_K(s_d)$ minimizes $\|s_d - v\|_{L^2}$ over all $v \in \mathbb{V}_K$) yields

$$\|s_d - s_{d,K}\|_{L^2} = \|s_d - P_K(s_d)\|_{L^2} \leq \|s_d - f\|_{L^2} < \frac{\varepsilon}{2} < \varepsilon,$$

where the inequality holds since $f \in \mathbb{V}_K$. Since $\varepsilon > 0$ was arbitrary, we conclude $\|s_d - s_{d,K}\|_{L^2} \to 0$ as $\min_c K_c \to \infty$.

Since the choice of $d \in \{1, \ldots, D\}$ was arbitrary, the convergence holds for all components, yielding $\boldsymbol{s}_K \xrightarrow{L^2} \boldsymbol{s}$ as $\min_c K_c \to \infty$. □

E Proof of Thm. 3

Lemma 1 (Rank Upper Bound)**.** *Let $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_T}$ be a Tth-order tensor. Then*

$$\operatorname{rank}(\boldsymbol{\mathcal{A}}) \leq \min_{1 \leq m \leq T} \prod_{\substack{j=1 \\ j \neq m}}^{T} I_j. \tag{37}$$

Proof. Fix $m \in \{1, \ldots, T\}$. We construct an explicit rank-1 decomposition with at most $\prod_{j \neq m} I_j$ terms by expanding in canonical bases.

Let $\boldsymbol{e}_{i_j}^{(j)} \in \mathbb{R}^{I_j}$ denote the i_jth standard basis vector, *i.e.*, $[\boldsymbol{e}_{i_j}^{(j)}]_k = \delta_{i_j k}$. Any tensor $\boldsymbol{\mathcal{A}}$ admits the expansion

$$\boldsymbol{\mathcal{A}} = \sum_{i_1=1}^{I_1} \cdots \sum_{i_T=1}^{I_T} a_{i_1 \cdots i_T}\, \boldsymbol{e}_{i_1}^{(1)} \otimes \cdots \otimes \boldsymbol{e}_{i_T}^{(T)}. \tag{38}$$

Define the multi-index $\boldsymbol{i}_{-m} \triangleq (i_1, \ldots, i_{m-1}, i_{m+1}, \ldots, i_T) \in \prod_{j \neq m} \{1, \ldots, I_j\}$. Reordering the summations in (38) and factoring out the mode-m sum yields

$$\begin{aligned} \boldsymbol{\mathcal{A}} &= \sum_{\boldsymbol{i}_{-m}} \sum_{i_m=1}^{I_m} a_{i_1 \cdots i_T}\, \boldsymbol{e}_{i_1}^{(1)} \otimes \cdots \otimes \boldsymbol{e}_{i_m}^{(m)} \otimes \cdots \otimes \boldsymbol{e}_{i_T}^{(T)} \\ &= \sum_{\boldsymbol{i}_{-m}} \boldsymbol{e}_{i_1}^{(1)} \otimes \cdots \otimes \underbrace{\left(\sum_{i_m=1}^{I_m} a_{i_1 \cdots i_T}\, \boldsymbol{e}_{i_m}^{(m)} \right)}_{\boldsymbol{a}_{\boldsymbol{i}_{-m}}^{(m)}} \otimes \cdots \otimes \boldsymbol{e}_{i_T}^{(T)}. \end{aligned} \tag{39}$$

Here $\boldsymbol{a}_{\boldsymbol{i}_{-m}}^{(m)} \in \mathbb{R}^{I_m}$ is the mode-m fiber of $\boldsymbol{\mathcal{A}}$ at index $\boldsymbol{i}_{-m}$ (Definition 9), with components $[\boldsymbol{a}_{\boldsymbol{i}_{-m}}^{(m)}]_{i_m} = a_{i_1 \cdots i_T}$. Each summand in (39) with a nonzero fiber is rank-1 (Definition 16), as the tensor product of T nonzero vectors; the others vanish. The sum runs over $\prod_{j \neq m} I_j$ distinct values of $\boldsymbol{i}_{-m}$, so (39) expresses $\boldsymbol{\mathcal{A}}$ as a sum of at most $\prod_{j \neq m} I_j$ rank-1 tensors. By Definition 17, $\operatorname{rank}(\boldsymbol{\mathcal{A}}) \leq \prod_{j \neq m} I_j$. Since $m \in \{1, \ldots, T\}$ was arbitrary, taking the minimum over all modes yields (37). □

Proof of Thm. 3. Fix $\varepsilon > 0$. By Thm. 2, taking the same number K of functions on every axis, for each component $d \in \{1, \ldots, D\}$ there exists $\tilde{K}_d \in \mathbb{N}$ such that for all $K \geq \tilde{K}_d$, the truncated generalized Fourier series $s_{d,K}$ in (3) satisfies $\|s_d - s_{d,K}\|_{L^2} < \varepsilon$. Let $K_0 \triangleq \max_d \tilde{K}_d$ and fix any $K \geq K_0$. Then $\|s_d - s_{d,K}\|_{L^2} < \varepsilon$ holds simultaneously for all d.

For this K, collect the Fourier coefficients $w_d^{(\boldsymbol{k})}$ from (4) into the $(C{+}1)$st-order coefficient tensor $\boldsymbol{\mathcal{W}} \in \mathbb{R}^{K^{\times C} \times D}$ so that $\boldsymbol{s}_K(\boldsymbol{x}) = \boldsymbol{\Phi}(\boldsymbol{x}) \bullet \boldsymbol{\mathcal{W}}$ by (6), with $\boldsymbol{\Phi}$ defined in (5). Applying Lem. 1 to $\boldsymbol{\mathcal{W}}$ (of sizes $I_1 = \cdots = I_C = K$ and $I_{C+1} = D$) yields

$$\operatorname{rank}(\boldsymbol{\mathcal{W}}) \leq \min\{K^C,\, K^{C-1} D\} = K^{C-1} \min\{K, D\}. \tag{40}$$

Hence $\boldsymbol{\mathcal{W}}$ admits a CP decomposition of the form (7) with some rank $R \leq K^{C-1} \min\{K, D\}$ and factor matrices $\{\boldsymbol{U}^{(c)}\}_{c=1}^{C}$, $\boldsymbol{V}$. Substituting this decomposition together with (5) into (6) and regrouping terms gives exactly the FUTON computation (8)–(10). Therefore $\hat{\boldsymbol{s}}(\boldsymbol{x}) \equiv \boldsymbol{s}_K(\boldsymbol{x})$, and for the chosen K and corresponding R and factors

$$\|s_d - \hat{s}_d\|_{L^2} = \|s_d - s_{d,K}\|_{L^2} < \varepsilon, \quad \forall d \in \{1, \ldots, D\}. \tag{41}$$

This establishes the theorem. □

F Error Analysis of FUTON with a Linear Head

Objective and takeaway. How accurate can a FUTON of spectral resolution K and rank R be on a given signal, before it is trained? With a linear head, the answer is exact (Thm. 4). The squared error is the *truncation*, the signal's energy beyond the first K_c basis functions per axis, which only K controls, plus the *CP misfit*, the distance from the signal's coefficient tensor to FUTON's rank-R tensor, which R limits and training reduces. The truncation follows from the samples, and SVDs bracket the best misfit: below by the *ceiling*, a PSNR no FUTON of that size passes, and above by a *certified FUTON*, built without training. Trained FUTONs land in the bracket, to within 0.02 dB; on images it is a median 0.05 dB wide, so it predicts training, at 16 to 860 times less cost (App. F.3). We prove this for any basis, on the continuous domain and on sampling grids, then ask what sets each part (App. F.1) and how the bounds are computed (App. F.2).

Setting. Let $\mathbb{X} = \mathbb{X}_1 \times \cdots \times \mathbb{X}_C$ carry a product measure $\mu = \mu_1 \otimes \cdots \otimes \mu_C$, with $\langle \boldsymbol{f}, \boldsymbol{f}' \rangle = \sum_d \int_{\mathbb{X}} f_d f'_d \, d\mu$ and $\|\boldsymbol{f}\|^2 = \langle \boldsymbol{f}, \boldsymbol{f} \rangle$ for signals $\boldsymbol{f}, \boldsymbol{f}' : \mathbb{X} \to \mathbb{R}^D$. Either $\mathbb{X}_c = [0,1]$ with the Lebesgue measure, as in the main text, or $\mathbb{X}_c$ holds the L_c sample positions of axis c with the uniform measure; then $\|\boldsymbol{f}\|^2 / D$ is the mean squared error over the $N = \prod_c L_c$ grid points, on which FUTON is trained and evaluated. Sampled cosines are not orthonormal on FUTON's grid, which includes both ends, so axis c may use any K_c square-integrable functions $\boldsymbol{\varphi}(x_c) \in \mathbb{R}^{K_c}$.

One change of basis does the work. The functions of axis c span a space with an orthonormal basis $\boldsymbol{\xi}(x_c) \in \mathbb{R}^{K'_c}$, $K'_c \le K_c$, and $\boldsymbol{\varphi} = \boldsymbol{B}_c^\top \boldsymbol{\xi}$ for a matrix $\boldsymbol{B}_c$ of full row rank (Lem. 2); for an orthonormal basis, $\boldsymbol{\xi} = \boldsymbol{\varphi}$ and $\boldsymbol{B}_c = \boldsymbol{I}$. The products $\prod_c \xi_{j_c}(x_c)$ are orthonormal, by Fubini as in Thm. 1, and span the space $\mathbb{V}_K$ of the products $\phi_{\boldsymbol{k}}$, $\boldsymbol{k} \in \mathbb{K}$; with P_K the orthogonal projection onto $\mathbb{V}_K$, per output dimension, $\boldsymbol{s}_K = P_K \boldsymbol{s}$ is the best approximation of $\boldsymbol{s}$ in $\mathbb{V}_K$, the partial sum (3) for an orthonormal basis. The signal's coefficients in the product basis form

$$\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{K'_1 \times \cdots \times K'_C \times D}, \qquad z_d^{(\boldsymbol{j})} = \langle s_d, \textstyle\prod_c \xi_{j_c} \rangle, \tag{42}$$

which is $\boldsymbol{\mathcal{W}}^\star$ for an orthonormal basis. With $\boldsymbol{\Xi}(\boldsymbol{x}) = \boldsymbol{\xi}(x_1) \otimes \cdots \otimes \boldsymbol{\xi}(x_C)$, $\boldsymbol{s}_K = \boldsymbol{\Xi} \bullet \boldsymbol{\mathcal{Z}}$, and Parseval reads $\|\boldsymbol{\Xi} \bullet \boldsymbol{\mathcal{A}}\| = \|\boldsymbol{\mathcal{A}}\|_F$. FUTON is $\hat{\boldsymbol{s}}_{\boldsymbol{\theta}} = \boldsymbol{\Phi} \bullet \boldsymbol{\mathcal{W}}$ with $\boldsymbol{\mathcal{W}} = [\![\boldsymbol{U}^{(1)}, \ldots, \boldsymbol{U}^{(C)}, \boldsymbol{V}]\!]$, by (6) and (7), with no bias or output activation; ε_R and $\boldsymbol{\mathcal{A}}_{\langle \mathbb{S} \rangle}$ are as in Sec. 4.3.

Theorem 4 (Prop. 1 for any basis)**.** *Let $\boldsymbol{s}$ be square-integrable, $K = (K_1, \ldots, K_C)$ and $R \ge 1$.*

1. Every FUTON. *For every $\boldsymbol{\theta}$, $\|\boldsymbol{s} - \hat{\boldsymbol{s}}_{\boldsymbol{\theta}}\|^2 = \|\boldsymbol{s} - \boldsymbol{s}_K\|^2 + \big\|\boldsymbol{\mathcal{Z}} - [\![\boldsymbol{B}_1 \boldsymbol{U}^{(1)}, \ldots, \boldsymbol{B}_C \boldsymbol{U}^{(C)}, \boldsymbol{V}]\!]\big\|_F^2$, and the truncation is $\|\boldsymbol{s} - \boldsymbol{s}_K\|^2 = \|\boldsymbol{s}\|^2 - \|\boldsymbol{\mathcal{Z}}\|_F^2$.*
2. The best FUTON. $\inf_{\boldsymbol{\theta}} \|\boldsymbol{s} - \hat{\boldsymbol{s}}_{\boldsymbol{\theta}}\|^2 = \|\boldsymbol{s} - \boldsymbol{s}_K\|^2 + \varepsilon_R(\boldsymbol{\mathcal{Z}})^2$.
3. Computable bounds. *$\underline{\varepsilon}_R \le \varepsilon_R(\boldsymbol{\mathcal{Z}}) \le \overline{\varepsilon}_R$, where $\underline{\varepsilon}_R^2 = \max_{\mathbb{S}} \sum_{i>R} \sigma_i^2(\boldsymbol{\mathcal{Z}}_{\langle \mathbb{S} \rangle})$ over the nonempty proper subsets $\mathbb{S}$ of the modes, and $\overline{\varepsilon}_R^2$ is the squared misfit of the R heaviest terms of an orthonormal rank-1 expansion of $\boldsymbol{\mathcal{Z}}$ by nested SVDs (Lem. 3). The FUTON built from these terms, the* certified FUTON*, has squared error exactly $\|\boldsymbol{s} - \boldsymbol{s}_K\|^2 + \overline{\varepsilon}_R^2$.*
4. When the bounds meet. *If at most two modes of $\boldsymbol{\mathcal{Z}}$ have more than one entry, then $\underline{\varepsilon}_R = \varepsilon_R(\boldsymbol{\mathcal{Z}}) = \overline{\varepsilon}_R$, and the infimum in (2) is attained; with three or more such modes, it need not be.*

All four quantities depend on the basis only through the span of $\boldsymbol{\varphi}$ on each axis.

Item (1) separates what K decides from what R and training decide; (2) reduces the best FUTON to the best rank-R approximation of $\boldsymbol{\mathcal{Z}}$, which (3) brackets and (4) pins down for two-axis signals with one output. Two lemmas supply the change of basis and the expansion.

Lemma 2 (Orthonormal basis of the span)**.** *Factor the Gram matrix $\boldsymbol{G}_c = \int_{\mathbb{X}_c} \boldsymbol{\varphi}\boldsymbol{\varphi}^\top d\mu_c = \boldsymbol{B}_c^\top \boldsymbol{B}_c$ with $\boldsymbol{B}_c \in \mathbb{R}^{K'_c \times K_c}$ of full row rank $K'_c = \operatorname{rank} \boldsymbol{G}_c$. The entries of $\boldsymbol{\xi} = (\boldsymbol{B}_c^+)^\top \boldsymbol{\varphi}$ are orthonormal in $L^2(\mathbb{X}_c, \mu_c)$, and $\boldsymbol{\varphi} = \boldsymbol{B}_c^\top \boldsymbol{\xi}$ almost everywhere, so $\boldsymbol{\xi}$ is an orthonormal basis of the span of $\boldsymbol{\varphi}$.*

Proof. Full row rank gives $\boldsymbol{B}_c \boldsymbol{B}_c^+ = \boldsymbol{I}$, hence $\int \boldsymbol{\xi}\boldsymbol{\xi}^\top d\mu_c = (\boldsymbol{B}_c^+)^\top \boldsymbol{B}_c^\top \boldsymbol{B}_c \boldsymbol{B}_c^+ = \boldsymbol{I}$. Next, $\boldsymbol{B}_c^\top \boldsymbol{\xi} = \boldsymbol{B}_c^+ \boldsymbol{B}_c \boldsymbol{\varphi}$, as $\boldsymbol{B}_c^+ \boldsymbol{B}_c$ is symmetric, and the rest of $\boldsymbol{\varphi}$ vanishes almost everywhere:

$\int \|(\boldsymbol{I} - \boldsymbol{B}_c^+ \boldsymbol{B}_c)\boldsymbol{\varphi}\|_2^2 \, d\mu_c = \|\boldsymbol{B}_c - \boldsymbol{B}_c \boldsymbol{B}_c^+ \boldsymbol{B}_c\|_F^2 = 0$. So each of $\boldsymbol{\varphi}$ and $\boldsymbol{\xi}$ is a linear map of the other, and they span the same space. □

Lemma 3 (Orthonormal rank-1 expansion)**.** *For any ordering of its modes, nested SVDs write a tensor $\boldsymbol{\mathcal{A}}$ with two or more modes as $\boldsymbol{\mathcal{A}} = \sum_\ell \lambda_\ell \boldsymbol{\mathcal{T}}_\ell$, with orthonormal rank-1 tensors $\boldsymbol{\mathcal{T}}_\ell$ and weights $\lambda_\ell \geq 0$. A sum of R of the terms has rank at most R, and its squared distance to $\boldsymbol{\mathcal{A}}$ is $\|\boldsymbol{\mathcal{A}}\|_F^2$ minus their squared weights, smallest for the R heaviest.*

Proof. By induction on the number of modes; with two, the SVD is such an expansion. Otherwise, an SVD of the unfolding along the first mode of the ordering gives $\boldsymbol{\mathcal{A}} = \sum_i \sigma_i \, \boldsymbol{a}_i \otimes \boldsymbol{\mathcal{G}}_i$, modes in place, with orthonormal vectors $\boldsymbol{a}_i$ and orthonormal tensors $\boldsymbol{\mathcal{G}}_i$ of the other modes. Expanding each $\boldsymbol{\mathcal{G}}_i$ by induction gives terms $\boldsymbol{a}_i \otimes \boldsymbol{\mathcal{T}}'_{i\ell}$, weighted by σ_i times the inner weights and orthonormal, since $\langle \boldsymbol{a}_i \otimes \boldsymbol{\mathcal{T}}'_{i\ell}, \boldsymbol{a}_{i'} \otimes \boldsymbol{\mathcal{T}}'_{i'\ell'} \rangle = \langle \boldsymbol{a}_i, \boldsymbol{a}_{i'} \rangle \langle \boldsymbol{\mathcal{T}}'_{i\ell}, \boldsymbol{\mathcal{T}}'_{i'\ell'} \rangle$. The residual of a sum of terms is the sum of the others, whose squared norm is the sum of their squared weights. □

Proof of Thm. 4. (1) By Lem. 2, $\boldsymbol{\Phi}(\boldsymbol{x}) = \bigotimes_c \boldsymbol{B}_c^\top \boldsymbol{\xi}(x_c)$ almost everywhere, so $\hat{\boldsymbol{s}}_{\boldsymbol{\theta}} = \boldsymbol{\Xi} \bullet \boldsymbol{\mathcal{Y}}_{\boldsymbol{\theta}}$ with $\boldsymbol{\mathcal{Y}}_{\boldsymbol{\theta}} = \boldsymbol{\mathcal{W}} \times_1 \boldsymbol{B}_1 \cdots \times_C \boldsymbol{B}_C$. A mode product with $\boldsymbol{B}_c$ replaces $\boldsymbol{u}_r^{(c)}$ by $\boldsymbol{B}_c \boldsymbol{u}_r^{(c)}$ in every rank-1 term (Definition 15), so $\boldsymbol{\mathcal{Y}}_{\boldsymbol{\theta}} = [\![\boldsymbol{B}_1 \boldsymbol{U}^{(1)}, \ldots, \boldsymbol{B}_C \boldsymbol{U}^{(C)}, \boldsymbol{V}]\!]$. As $\boldsymbol{s} - \hat{\boldsymbol{s}}_{\boldsymbol{\theta}} = (\boldsymbol{s} - \boldsymbol{s}_K) + \boldsymbol{\Xi} \bullet (\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{Y}}_{\boldsymbol{\theta}})$, the first term orthogonal to $\mathbb{V}_K$ and the second in it, Pythagoras and Parseval give the identity, and $\boldsymbol{\mathcal{Y}}_{\boldsymbol{\theta}} = \boldsymbol{0}$ the truncation.

(2) The tensors $\boldsymbol{\mathcal{Y}}_{\boldsymbol{\theta}}$ are exactly those of rank at most R: each is a sum of R rank-1 terms, and any such sum $[\![\tilde{\boldsymbol{U}}^{(1)}, \ldots, \tilde{\boldsymbol{U}}^{(C)}, \tilde{\boldsymbol{V}}]\!]$ is $\boldsymbol{\mathcal{Y}}_{\boldsymbol{\theta}}$ for $\boldsymbol{\theta} = (\boldsymbol{B}_1^+ \tilde{\boldsymbol{U}}^{(1)}, \ldots, \boldsymbol{B}_C^+ \tilde{\boldsymbol{U}}^{(C)}, \tilde{\boldsymbol{V}})$, since $\boldsymbol{B}_c \boldsymbol{B}_c^+ = \boldsymbol{I}$. So (1) is minimized over that set.

(3) *Lower bound.* Every rank-1 term of a tensor $\boldsymbol{\mathcal{Y}}$ of rank at most R unfolds to a rank-1 matrix, as (28) shows for one mode, so $\operatorname{rank} \boldsymbol{\mathcal{Y}}_{\langle \mathbb{S} \rangle} \leq R$; unfolding keeps the Frobenius norm, and the Eckart–Young–Mirsky theorem gives $\|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{Y}}\|_F^2 \geq \sum_{i>R} \sigma_i^2(\boldsymbol{\mathcal{Z}}_{\langle \mathbb{S} \rangle})$ for every $\mathbb{S}$. *Upper bound.* By Lem. 3, the R heaviest terms sum to a tensor of rank at most R at squared distance $\overline{\varepsilon}_R^2$ from $\boldsymbol{\mathcal{Z}}$, which is $\boldsymbol{\mathcal{Y}}_{\boldsymbol{\theta}}$ for the $\boldsymbol{\theta}$ of (2); (1) gives that FUTON's error. Any tensor of rank at most R gives an upper bound in the same way.

(4) With two modes of more than one entry, rank means matrix rank in those modes, and the expansion of Lem. 3 is the SVD, whose R heaviest terms attain the Eckart–Young–Mirsky optimum, so $\overline{\varepsilon}_R = \underline{\varepsilon}_R$. With three or more, the tensors of rank at most R need not form a closed set, and a best approximation need not exist [46].

Last, any orthonormal basis of the same spans is $\boldsymbol{O}_c \boldsymbol{\xi}$ with $\boldsymbol{O}_c$ orthogonal, which rotates $\boldsymbol{\mathcal{Z}}$ mode by mode and changes none of the four quantities. □

F.1 What Sets the Two Terms

The truncation falls with smoothness (Cor. 1); the misfit falls with structure (Prop. 2 and Thm. 5), but no bound on the rank holds for all signals (Prop. 3), so Prop. 1 measures the signal at hand. The edge test shows both effects, and Rem. 1 what carries over to the benchmarks' heads. Corollary 1 and Thm. 5 use the cosine basis on $[0,1]^C$ of Sec. 3.2.

Corollary 1 (Truncation rate)**.** *For the cosine basis* (1) *and $\boldsymbol{s} \in L^2([0,1]^C; \mathbb{R}^D)$ whose weak partial derivatives $\partial_c \boldsymbol{s}$ are square-integrable,*

$$\|\boldsymbol{s} - \boldsymbol{s}_K\|^2 \leq \sum_{c=1}^{C} \frac{\|\partial_c \boldsymbol{s}\|^2}{(\pi K_c)^2}, \tag{43}$$

to which the best FUTON adds $\varepsilon_R(\boldsymbol{\mathcal{W}}^\star)^2$. If, along every axis, the odd derivatives of $\boldsymbol{s}$ of order below m vanish at 0 and 1, the bound improves to $\sum_c \|\partial_c^m \boldsymbol{s}\|^2 / (\pi K_c)^{2m}$.

Proof. Take one axis and $f \in L^2(0,1)$ with $f' \in L^2(0,1)$, and let $w_k = \langle f, \varphi_k \rangle$. The functions $\chi_k(x) = \sqrt{2}\sin(k\pi x)$, $k \geq 1$, are orthonormal, and integrating by parts, $\langle f', \chi_k \rangle = [f\chi_k]_0^1 - k\pi \langle f, \varphi_k \rangle = -k\pi w_k$, since $\chi_k(0) = \chi_k(1) = 0$. By Bessel's inequality, $\sum_{k \geq 1} (k\pi)^2 w_k^2 \leq \|f'\|^2$,

hence $\sum_{k\geq K} w_k^2 \leq \|f'\|^2/(\pi K)^2$. In C dimensions, a multi-index outside $\mathbb{K}$ has $k_c \geq K_c$ for some c, so by Parseval $\|\boldsymbol{s}-\boldsymbol{s}_K\|^2 \leq \sum_c \sum_{\boldsymbol{k}:k_c\geq K_c} \left\|\boldsymbol{w}^{(\boldsymbol{k})}\right\|^2$. For fixed indices of the other axes, $\boldsymbol{w}^{(\boldsymbol{k})}$ is the k_c-th cosine coefficient of the integral of $\boldsymbol{s}$ against the other axes' basis functions, whose derivative is the same integral of $\partial_c \boldsymbol{s}$; the one-axis bound, summed with Bessel's inequality, bounds the inner sum by $\|\partial_c \boldsymbol{s}\|^2/(\pi K_c)^2$. Under the boundary conditions, m integrations by parts leave no boundary terms and give the rate $(\pi K_c)^{-2m}$. □

Proposition 2 (Separation rank)**.** *If $\boldsymbol{s}$ is within ε of a sum of R separable terms, $\left\|\boldsymbol{s}-\sum_{r=1}^{R} q_r^{(1)}(x_1)\cdots q_r^{(C)}(x_C)\,\boldsymbol{v}_r\right\| \leq \varepsilon$ with $q_r^{(c)} \in L^2(\mathbb{X}_c,\mu_c)$ and $\boldsymbol{v}_r \in \mathbb{R}^D$, then, for every basis, the best FUTON of rank R at resolution K has squared error at most $\|\boldsymbol{s}-\boldsymbol{s}_K\|^2+\varepsilon^2$: a fixed basis costs only the truncation.*

Proof. The projection onto $\mathbb{V}_K$ factorizes over the axes, $P_K = P_{K_1}\otimes\cdots\otimes P_{K_C}$, so it maps each separable term to a separable term, and $P_K \sum_r q_r^{(1)}\cdots q_r^{(C)}\boldsymbol{v}_r = \boldsymbol{\Xi}\bullet\boldsymbol{\mathcal{Y}}$ with $\boldsymbol{\mathcal{Y}}$ of rank at most R. By Thm. 4 (1) and (2), the best FUTON's squared error is at most $\|\boldsymbol{s}-\boldsymbol{s}_K\|^2+\|\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{Y}}\|_F^2$, and $\|\boldsymbol{\mathcal{Z}}-\boldsymbol{\mathcal{Y}}\|_F = \left\|P_K(\boldsymbol{s}-\sum_r q_r^{(1)}\cdots q_r^{(C)}\boldsymbol{v}_r)\right\| \leq \varepsilon$, as projections are nonexpansive. □

Lemma 4 (Box covers bound the rank)**.** *Let $\mathcal{F}$ be a finite family of boxes $\mathbb{I}=\mathbb{I}_1\times\cdots\times\mathbb{I}_C \subseteq \mathbb{K}$, $\mathbb{I}_c \subseteq \{0,\ldots,K_c-1\}$, and*

$$\operatorname{cost}(\mathcal{F}) = \sum_{\mathbb{I}\in\mathcal{F}} \frac{D\prod_c |\mathbb{I}_c|}{\max\{|\mathbb{I}_1|,\ldots,|\mathbb{I}_C|,D\}}. \tag{44}$$

Every tensor in $\mathbb{R}^{K_1\times\cdots\times K_C\times D}$ that vanishes outside $\bigcup\mathcal{F}$ has rank at most $\operatorname{cost}(\mathcal{F})$.

Proof. Split the tensor into its parts on each box minus the earlier boxes. Each is a tensor of size $|\mathbb{I}_1|\times\cdots\times|\mathbb{I}_C|\times D$ padded with zeros, of rank at most the product of its sizes but the largest (Lem. 1), its term in (44); rank is subadditive. □

Theorem 5 (Low-rank sufficiency)**.** *Let $\boldsymbol{s}\in L^2([0,1]^C;\mathbb{R}^D)$ have cosine coefficients $\boldsymbol{w}^{(\boldsymbol{k})}$, and let a family of boxes $\mathcal{F}$ cover a set $\mathbb{A}\subseteq\mathbb{K}$ of multi-indices. Some FUTON of resolution K and rank at most $\operatorname{cost}(\mathcal{F})$, with a linear head, has squared error at most $\sum_{\boldsymbol{k}\notin\mathbb{A}}\left\|\boldsymbol{w}^{(\boldsymbol{k})}\right\|^2$, the energy of $\boldsymbol{s}$ outside $\mathbb{A}$.*

Proof. Keep $\boldsymbol{\mathcal{W}}^\star$ on $\bigcup\mathcal{F}\supseteq\mathbb{A}$ and set it to zero elsewhere. By Lem. 4 the result has rank at most $\operatorname{cost}(\mathcal{F})$, so it is $[\![\boldsymbol{U}^{(1)},\ldots,\boldsymbol{U}^{(C)},\boldsymbol{V}]\!]$ for factors of that rank; by Prop. 1 (1) and Parseval, that FUTON's squared error is the energy outside $\bigcup\mathcal{F}$, at most the energy outside $\mathbb{A}$. □

Theorem 3 is the case $\mathbb{A}=\mathbb{K}$. A *sub-box* $\{0,\ldots,q-1\}^C$ costs $q^{C-1}\min\{q,D\}$ and, by Cor. 1, leaves at most $\|\nabla\boldsymbol{s}\|^2/(\pi q)^2$ outside, so rank $q^{C-1}\min\{q,D\}$ with $q=\lceil\|\nabla\boldsymbol{s}\|/(\pi\varepsilon)\rceil$ suffices for squared error ε^2, set by the accuracy, not by K. A *hyperbolic cross* $\{\boldsymbol{k}:(k_1+1)(k_2+1)\leq\nu\}$, where signals of dominating mixed smoothness concentrate [47], lies in two strips of width $\lceil\sqrt{\nu}\rceil$, at cost $2D\lceil\sqrt{\nu}\rceil$: 138 at $\nu=K_c=512$ and $D=3$, against 1,536 for the full grid. Both bound what FUTONs can represent, not what training finds.

Proposition 3 (No rank bound holds for all signals)**.** *If $R(\sum_c K_c'+D-C) < D\prod_c K_c'$, the tensors of rank at most R in $\mathbb{R}^{K_1'\times\cdots\times K_C'\times D}$, with their limits, form a set of Lebesgue measure zero. So $\varepsilon_R(\boldsymbol{\mathcal{Z}})=0$ only for exceptional coefficient tensors; almost every $K\times K\times K$ tensor has rank at least $\lceil K^3/(3K-2)\rceil$, 21,903 at $K=256$.*

Proof. The map from factors to $[\![\boldsymbol{U}^{(1)},\ldots,\boldsymbol{U}^{(C)},\boldsymbol{V}]\!]$ is polynomial, and rescaling a term's $C+1$ factors by numbers of product one leaves it unchanged, so the tensors of rank at most R form a semialgebraic set of dimension at most $R(\sum_c K_c'+D-C)$, as does its closure, which lies in the R-th secant variety of a Segre variety [48]; such a set has measure zero when its dimension is below the ambient one. For $K\times K\times K$ tensors the condition reads $R(3K-2)<K^3$, and the bound is the typical rank over the complex numbers [49]. □

Table 4: **The edge test.** Smallest rank R at which an edge's cosine coefficients ($K_c = 256$) have relative squared error at most 10^{-3}, 10^{-4}, and 10^{-5}, per width δ and direction, exactly from the singular values (4096×4096 midpoint quadrature).

	Axis-aligned			Oblique, 30°			Diagonal, 45°		
Width δ	10^{-3}	10^{-4}	10^{-5}	10^{-3}	10^{-4}	10^{-5}	10^{-3}	10^{-4}	10^{-5}
0.2	1	1	1	4	4	5	4	4	6
0.1	1	1	1	5	6	8	6	8	10
0.05	1	1	1	7	10	12	9	13	16
0.02	1	1	1	13	19	25	18	26	35
0.01	1	1	1	21	33	45	30	47	63
0.005	1	1	1	34	57	80	49	82	115

So typical tensors attain the order of Thm. 3's bound, and only structure in the signal brings the rank down.

The edge test. A soft edge of width δ through the center of $[0,1]^2$, $s(\boldsymbol{x}) = \tanh(\boldsymbol{a}^\top(\boldsymbol{x} - \frac{1}{2}\mathbf{1})/\delta)$ with a unit direction $\boldsymbol{a}$, tests these predictions. Along an axis, it is separable, of rank one at any width. Obliquely, a profile within ε of a polynomial of degree p gives a polynomial of total degree p in (x_1, x_2), of rank at most $p+1$, with p growing as $(1/\delta)\log(1/\varepsilon)$ for $\tanh(t/\delta)$. With $C = 2$ and $D = 1$, Thm. 4 (4) gives the exact rank each accuracy needs (Tab. 4): one for the axis-aligned edge at every width, and about $(1/\delta)^{0.6}$ to $(1/\delta)^{0.8}$ for oblique edges, within the predicted $\mathcal{O}(1/\delta)$; orientation costs rank only through sharpness.

Remark 1 (Nonlinear heads and the output activation). The benchmarks' FUTON adds an MLP head of width $M = 64$ on images and radiance fields, or a bias on volumes, and a $\tanh$ output on images and volumes, which the analysis does not cover. The upper bound transfers to both heads, since a bias can be zero and an MLP head of width $M \geq 2D$ contains every linear head, as $\mathrm{ReLU}(t) - \mathrm{ReLU}(-t) = t$: the best such FUTON at (K, R) is at least as good as the certified one, with more parameters. The lower bound does not, as biases and a hidden layer fit what no linear head can. As $\tanh$ is 1-Lipschitz, for a target $\boldsymbol{s} = \tanh(\tilde{\boldsymbol{s}})$ with $\tilde{\boldsymbol{s}}$ square-integrable (on a grid, any target with $|\boldsymbol{s}| < 1$), the error of $\tanh \circ \hat{\boldsymbol{s}}$ against $\boldsymbol{s}$ is at most that of $\hat{\boldsymbol{s}}$ against $\tilde{\boldsymbol{s}}$, so the analysis of $\tilde{\boldsymbol{s}}$ certifies the $\tanh$ model; a target that reaches ± 1, as occupancy and the Kodak images do, has no such $\tilde{\boldsymbol{s}}$.

Related results. The ingredients are classical; new is their combination for this model, where every quantity is computed from the signal and the upper bound is attained. The truncated SVD is optimal for matrices [50, 51], item (4), but with three or more modes a best approximation need not exist [46], hence (2) and (3). The lower bound is the matricization tail that also bounds the HOSVD [52] and TT-SVD [30] from below, and Lem. 3 is a tree of SVDs, one per branch, with its heaviest terms kept, for two modes the Kronecker-product SVD [53]. Separated representations [14, 32] keep their separation rank in a fixed basis up to the truncation (Prop. 2), and CP models of Fourier coefficients [33, 34] come with bounds in expectation or asymptotically, where Prop. 1 is an identity for the signal at hand; for MLP-based INRs, no analogue of item (2) is known [54].

F.2 Algorithm

On the grid, axis c's basis is the matrix $\boldsymbol{\Phi}_c \in \mathbb{R}^{L_c \times K_c}$ of its values at the sample positions, and the signal is the array $\boldsymbol{\mathcal{S}} \in \mathbb{R}^{L_1 \times \cdots \times L_C \times D}$ of its samples. A thin SVD $\boldsymbol{\Phi}_c/\sqrt{L_c} = \boldsymbol{Q}_c \boldsymbol{\Sigma}_c \boldsymbol{E}_c^\top$, truncated at the numerical rank K_c', orthonormalizes the axis: $\boldsymbol{B}_c = \boldsymbol{\Sigma}_c \boldsymbol{E}_c^\top$, and the values of $\boldsymbol{\xi}$ are the rows of $\sqrt{L_c}\,\boldsymbol{Q}_c$. Table 5 lists the steps, all in double precision. Only Steps 2 and 3 read the signal, one matrix product per axis and a sum of squares; the rest works on $\boldsymbol{\mathcal{Z}}$, whose size is set by K, not by the signal, which keeps the bound cheap. Dividing by D turns the outputs into mean squared errors.

F.3 Experiments

The experiments test what Prop. 1 tells before training: whether its guarantees hold, where training lands, how well it sizes a FUTON, and what it costs. In short: the bracket always holds; on images it

Table 5: The bounds from samples. Steps 5 and 6 give the certified FUTON.

Step	Computes	Cost
1	$\boldsymbol{Q}_c, \boldsymbol{\Sigma}_c, \boldsymbol{E}_c$ from the thin SVD of $\boldsymbol{\Phi}_c/\sqrt{L_c}$	$\mathcal{O}(L_c K_c^2)$ per axis
2	$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{S}} \times_1 \boldsymbol{Q}_1^\top \cdots \times_C \boldsymbol{Q}_C^\top/\sqrt{N}$	$\mathcal{O}(ND\sum_c K_c)$
3	$\|\boldsymbol{s} - \boldsymbol{s}_K\|^2 = \|\boldsymbol{\mathcal{S}}\|_F^2/N - \|\boldsymbol{\mathcal{Z}}\|_F^2$, exact	$\mathcal{O}(ND)$
4	$\underline{\varepsilon}_R^2$: the singular values of every matricization of $\boldsymbol{\mathcal{Z}}$, tails beyond each R	one eigendecomposition per split
5	Lem. 3 for every mode ordering, keeping the ordering with the most weight in its R heaviest terms; $\bar{\varepsilon}_R^2 = \|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{Y}}\|_F^2$ for their sum $\boldsymbol{\mathcal{Y}} = [\![\tilde{\boldsymbol{U}}^{(1)}, \ldots, \tilde{\boldsymbol{U}}^{(C)}, \tilde{\boldsymbol{V}}]\!]$	nested SVDs of $\boldsymbol{\mathcal{Z}}$
6	$\hat{\boldsymbol{U}}^{(c)} = \boldsymbol{E}_c \boldsymbol{\Sigma}_c^{-1} \tilde{\boldsymbol{U}}^{(c)}$, $\hat{\boldsymbol{V}} = \tilde{\boldsymbol{V}}$	$\mathcal{O}(R\sum_c K_c K_c')$

is narrow and locates training, on volumes it is wide and guarantees a PSNR; and it costs a fraction of training.

Setup. The signals are the 24 Kodak images ($C = 2$, $D = 3$, values in $[-1, 1]$) and the five occupancy volumes of Sec. 5.2 ($C = 3$, $D = 1$, values ± 1), in the cosine basis at $K_c = \kappa L_c$ with $\kappa \in \{1/8, 1/4, 1/2\}$ and $R \in \{32, 64, \ldots, 1024\}$, around the benchmark's ranks: 18 settings, 522 runs, and $\kappa \in \{1, 2\}$ for the cost alone. For each K, the algorithm of App. F.2 gives the truncation, the ceiling and the certified FUTON at all six ranks in one timed call; the same linear-head model is trained by Adam under the protocol of Sec. 5, at learning rate 10^{-1} on images and $3 \cdot 10^{-2}$ on volumes. Errors are mean squared errors over the whole grid, and $\mathrm{PSNR} = 10 \log_{10}(4/\mathrm{MSE})$. The bound runs mostly on the CPU, in double precision, so its timings depend on the CPU's load; the volumes' were timed in a run of their own. Figure 3 compares the bounds with training at $\kappa = 1/2$, and Tab. 6 lists the means at every setting.

The bracket holds. No trained FUTON passes the ceiling beyond rounding, in any of the 522 runs: in 15 of them, all on Kodak at full rank, where the ceiling is the truncation and the full-rank FUTON reaches it, Adam's error lies below it by a relative $2.3 \cdot 10^{-8}$ at most, the single-precision rounding of its forward pass. The certified side is a construction, not a limit on training, yet training respects it too: Adam's FUTON never ends more than 0.02 dB below the certified one. Training thus lands in the bracket in every run, to within 0.02 dB. Unit tests check that every certified FUTON attains its upper bound, to a relative 10^{-9}, as well as the Eckart–Young case, the invariance within a span, and a DCT cross-check.

Images: training, located. For an image, the bracket is narrow: a median 0.05 dB over images and settings, at most 0.1 dB in 56% of them and at most 0.5 dB in 78%. It widens with K, to medians of 0.00, 0.05 and 0.37 dB at $\kappa = 1/8$, $1/4$ and $1/2$, and closes as the certified FUTON reaches the truncation, which it does on every image from $R = 256$ on at $\kappa = 1/8$, from 512 on at $1/4$, and at 1024 at $1/2$. Adam lands at a steady place inside it, a median 0.66 of the way from the certified FUTON to the ceiling, and 0.60 to 0.70 in half the settings where it is open; at $\kappa = 1/2$, a median 0.24 dB above the certified FUTON and 0.11 dB below the ceiling. Before training, Prop. 1 thus says what Adam will reach to within a tenth of a decibel in over half the settings, and within a few tenths at the largest K.

Volumes: a wider bracket. With three axes and one output, every matricization of $\boldsymbol{\mathcal{Z}}$ has rank at most the largest K_c, so once R passes it the ceiling falls back to the truncation, although the misfit need not vanish; and the certified FUTON trails training further than on images. The bracket is a median 0.98, 3.43 and 5.37 dB wide at $\kappa = 1/8$, $1/4$ and $1/2$, with Adam a median 0.67, 1.07 and 1.19 dB above the certified FUTON and, at $\kappa = 1/4$ and $1/2$, the ceiling the looser side. At $\kappa = 1/8$, the truncation caps the volumes at 20.6 dB on average, which the certified FUTON reaches to within 3.69 dB at $R = 32$, 0.60 dB at 256 and 0.01 dB at 1024. On volumes the bracket is a guarantee more than a forecast: a PSNR that a FUTON of the size reaches without training, and one that none passes.

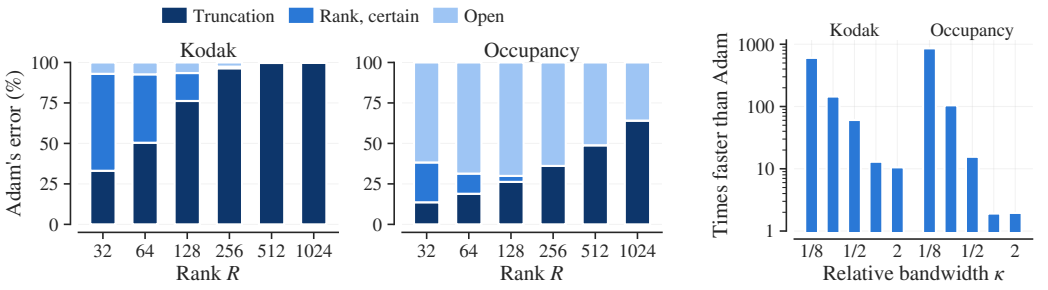
Where Adam's error goes. Proposition 1 splits every FUTON's error into the truncation, which K forces, and the misfit, which the ceiling splits again: up to the ceiling it is forced by R, and above it

Table 6: **Means per setting** over the 24 Kodak images and over the five occupancy volumes. PSNR (dB), ± one within-signal standard error: the truncation, the certified FUTON, the ceiling and Adam's FUTON; seconds on an A100 for the bound at all six ranks of a K and for one Adam run.

						Seconds	
κ	R	Truncation	Certified	Ceiling	Adam	Bound	Adam
Images							
$1/8$	32	23.65±0.06	22.85±0.06	23.27±0.01	23.13±0.02	0.02	1.7
$1/8$	64	23.65±0.01	23.45±0.03	23.62±0.01	23.57±0.00	0.02	1.7
$1/8$	128	23.65±0.00	23.64±0.00	23.65±0.00	23.65±0.00	0.02	1.7
$1/8$	256	23.65±0.00	23.65±0.00	23.65±0.00	23.65±0.00	0.02	1.7
$1/8$	512	23.65±0.00	23.65±0.00	23.65±0.00	23.65±0.00	0.02	1.7
$1/8$	1024	23.65±0.00	23.65±0.00	23.65±0.00	23.65±0.00	0.02	1.7
$1/4$	32	25.88±0.15	23.70±0.10	24.35±0.03	24.10±0.05	0.07	1.7
$1/4$	64	25.88±0.07	24.90±0.07	25.41±0.02	25.23±0.02	0.07	1.7
$1/4$	128	25.88±0.02	25.64±0.03	25.85±0.01	25.79±0.00	0.07	1.7
$1/4$	256	25.88±0.00	25.87±0.00	25.88±0.00	25.88±0.00	0.07	1.7
$1/4$	512	25.88±0.00	25.88±0.00	25.88±0.00	25.88±0.00	0.07	1.7
$1/4$	1024	25.88±0.00	25.88±0.00	25.88±0.00	25.88±0.00	0.07	1.7
$1/2$	32	29.86±0.25	24.34±0.13	25.17±0.06	24.85±0.08	0.17	1.7
$1/2$	64	29.86±0.19	26.22±0.11	27.11±0.04	26.77±0.06	0.17	1.7
$1/2$	128	29.86±0.11	28.13±0.09	28.94±0.03	28.65±0.03	0.17	1.7
$1/2$	256	29.86±0.02	29.46±0.04	29.82±0.02	29.70±0.01	0.17	1.7
$1/2$	512	29.86±0.00	29.84±0.00	29.86±0.00	29.85±0.00	0.17	1.7
$1/2$	1024	29.86±0.00	29.86±0.00	29.86±0.00	29.86±0.00	0.17	1.7
Occupancy volumes							
$1/8$	32	20.62±0.26	16.94±0.25	20.22±0.17	17.92±0.18	0.05	3.5
$1/8$	64	20.62±0.19	18.08±0.24	20.62±0.19	19.08±0.14	0.05	3.8
$1/8$	128	20.62±0.15	19.13±0.21	20.62±0.15	19.99±0.09	0.05	4.7
$1/8$	256	20.62±0.09	20.02±0.18	20.62±0.09	20.47±0.01	0.05	6.4
$1/8$	512	20.62±0.03	20.49±0.09	20.62±0.03	20.60±0.02	0.05	9.8
$1/8$	1024	20.62±0.00	20.61±0.01	20.62±0.00	20.62±0.00	0.05	17.6
$1/4$	32	23.64±0.33	17.10±0.25	21.69±0.10	18.10±0.18	0.44	3.5
$1/4$	64	23.64±0.25	18.34±0.25	23.16±0.17	19.48±0.17	0.44	3.8
$1/4$	128	23.64±0.21	19.57±0.25	23.64±0.21	20.80±0.18	0.44	4.6
$1/4$	256	23.64±0.19	20.81±0.23	23.64±0.19	21.99±0.15	0.44	6.5
$1/4$	512	23.64±0.16	22.00±0.22	23.64±0.16	22.92±0.09	0.44	9.9
$1/4$	1024	23.64±0.10	22.97±0.19	23.64±0.10	23.43±0.02	0.44	17.6
$1/2$	32	26.94±0.39	17.20±0.24	22.37±0.05	18.18±0.18	2.95	3.4
$1/2$	64	26.94±0.32	18.48±0.24	24.68±0.10	19.62±0.18	2.95	3.8
$1/2$	128	26.94±0.27	19.77±0.25	26.35±0.18	21.06±0.20	2.95	4.7
$1/2$	256	26.94±0.24	21.12±0.26	26.94±0.24	22.45±0.21	2.95	6.5
$1/2$	512	26.94±0.22	22.49±0.26	26.94±0.22	23.78±0.19	2.95	9.9
$1/2$	1024	26.94±0.21	23.83±0.25	26.94±0.21	24.96±0.17	2.95	17.6

is open, between the rank and training (Fig. 12a). At $\kappa = 1/2$ on Kodak, the truncation is 33% of Adam's error at $R = 32$ and all but 0.1% at 1024, the part that R forces falls from 60% to nothing, and at most 7% stays open: Adam's error is almost entirely accounted for before training. On the volumes, the truncation is 14% at 32 and 64% at 1024, R forces 25% at 32 and nothing from 256 on, where the ceiling meets the truncation, and the open part, 36 to 70%, is the largest share up to $R = 512$.

How close is training to the best FUTON? The ceiling also grades training: a trained FUTON's distance to it is the most a better optimizer could add. Table 7 (top) trains every signal at $\kappa = 1/2$ and the benchmark's rank with Adam for 1000 and 5000 steps. On Kodak the best FUTON of that size lies between 29.80 dB, Adam's at 5000 steps, and the ceiling's 29.84 dB: the benchmark's 1000 steps end within 0.09 dB of the ceiling on average and within 0.18 dB on every image. On the volumes it lies between 22.94 dB, Adam's at 5000 steps, and the loose ceiling's 26.94 dB. Every trained FUTON passes the certified one on every signal, by up to 0.65 dB on the images and by 1.22 to 2.11 dB on the volumes.



(a) Where Adam's error goes

(b) Cost of the bound

Figure 12: **Where Adam's error goes, and what the bound costs.** (a) Adam's error at $\kappa = 1/2$, in percent, as Prop. 1 splits it, averaged over the 24 Kodak images and the five occupancy volumes, per rank: the truncation, which K forces; the part of the misfit up to the ceiling, which R forces with certainty; and the rest, which the bounds leave open. (b) How many times faster the bound, certified FUTON included, is than the six Adam runs it stands in for, per relative bandwidth κ, from the seconds per signal averaged over each dataset.

Table 7: **Training against the bound.** Top: mean PSNR (dB) over each dataset, $\pm$ one within-signal standard error, at $\kappa = 1/2$ and the benchmark's rank, for the certified FUTON and the ceiling against Adam at the benchmark's 1000 steps and at 5000. Bottom: seconds per signal for the six ranks of a K, from $\kappa = 1/8$ to 2, averaged over each dataset, for the bound, certified FUTON included, and the six Adam runs; and how far the certified FUTON stays below Adam's, in dB, over the signals and ranks.

				Adam	
Dataset	R	Certified	Ceiling	1000 steps	5000 steps
Kodak	280	29.56±0.03	29.84±0.01	29.75±0.00	29.80±0.01
Occupancy	215–310	21.22±0.16	26.94±0.43	22.55±0.10	22.94±0.07

	Kodak images				Occupancy volumes			
	Seconds		Behind Adam (dB)		Seconds		Behind Adam (dB)	
κ	Bound	Adam	Median	Max	Bound	Adam	Median	Max
1/8	0.02	10	0.00	0.93	0.05	46	0.67	1.29
1/4	0.07	10	0.03	1.11	0.44	46	1.07	1.47
1/2	0.17	10	0.24	1.28	2.95	46	1.19	1.56
1	0.79	10	0.85	5.83	24.19	46	1.28	1.57
2	0.97	10	0.81	5.25	23.82	47	1.31	1.66

Sizing a FUTON for a target PSNR. The bracket on PSNR is also one on size: among the candidate settings, no FUTON smaller than the smallest whose ceiling reaches a target reaches it, and the smallest whose certified PSNR reaches it does, with its model. Table 8 sets both against an oracle that trains every setting and takes the smallest that Adam reaches. The oracle's pick lies between the two on every signal and target, and the certified pick is never smaller. On Kodak the bracket on size is tight: the two picks coincide for 56% of the images and targets, so the size a target needs is known before training, and the certified pick is the oracle's for 76% and at most twice it otherwise. On the volumes, the ceiling's pick is a median eight times smaller than the certified one, which is safe but generous, at most twice the oracle's parameters at the median of every target; at 23 dB it reaches four of the five shapes that Adam does. Read along a budget, the certified mean picks the spectral resolution that Adam's results pick at every parameter count on Kodak ($\kappa = 1/8$ at about 5k, $1/4$ at 10k and 21k, $1/2$ from 41k on), and at six of eight on the volumes.

Cost. The bound gives the truncation, the ceiling and the certified FUTON at all six ranks of a K in one call: on an A100, on average 0.02 to 0.17 s per image and 0.05 to 3.0 s per volume at $\kappa \leq 1/2$, against 1.7 and 7.7 s for one Adam run. Sizing a signal over every candidate setting, as above, takes

Table 8: **The smallest setting for a target PSNR.** Reached: the signals of each dataset for which some setting reaches the target by Adam, by the certified FUTON, and by the ceiling (possible). Pick vs. oracle: over the signals both reach, how often the certified pick has the parameter count of the oracle's, which trains every setting and takes the smallest that Adam reaches, or a larger one; it is never smaller. #Params: the median parameter count of the ceiling's pick, below which no candidate reaches the target, of the oracle's, and of the certified pick.

	Reached			Pick vs. oracle		#Params (k)		
Target (dB)	Adam	Certified	Possible	Same	Larger	Ceiling	Adam	Certified
Kodak images								
22	24/24	24/24	24/24	88%	12%	5.2	5.2	5.2
24	23/24	23/24	23/24	70%	30%	5.2	10.3	10.3
26	22/24	22/24	22/24	77%	23%	20.7	20.7	30.9
28	18/24	18/24	18/24	67%	33%	41.2	61.7	61.8
30	13/24	13/24	13/24	77%	23%	82.3	82.3	82.3
Occupancy volumes								
19	5/5	5/5	5/5	20%	80%	3.9	7.7	14.1
20	5/5	5/5	5/5	0%	100%	4.8	15.5	31.0
21	5/5	5/5	5/5	0%	100%	7.8	31.1	62.2
22	5/5	5/5	5/5	0%	100%	14.1	62.2	124.4
23	5/5	4/5	5/5	0%	100%	15.6	124.4	237.6

0.26 s per image against 31 s of training, and 3.4 s per volume against 138 s: 40 to 120 times less. The bound grows with K, as $\boldsymbol{\mathcal{Z}}$ does, and stays below training: at $\kappa = 1$ and 2, 0.8 to 1.0 s per image, under a tenth of the six Adam runs, and 24 s per volume, about half of them (Fig. 12b), while the certified FUTON trails training further on the images there (Tab. 7, bottom).

G Detailed Experimental Setup

Our implementation, the library Inro, with the configuration files and scripts behind every run, is available at `https://github.com/pashtari/inro`.

Data. *Images.* The 24 Kodak images [43], 768×512 or 512×768, with RGB values scaled to $[-1, 1]$ and coordinates on $[-1, 1]^2$; every step uses the whole image. *Occupancy volumes.* Five meshes from the Stanford 3D Scanning Repository [44], normalized as in BACON [25]. Each voxel center of a grid with 512 voxels per unit length is marked inside or outside the scan by a majority of five ray-parity tests, and the grid is cropped to the shape: $386 \times 460 \times 351$ (Armadillo), $461 \times 325 \times 206$ (Dragon), $190 \times 461 \times 189$ (Happy Buddha), $268 \times 154 \times 461$ (Lucy), and $274 \times 461 \times 236$ (Thai Statue), 17 to 62 million voxels. Targets are ± 1; a step draws 2^{20} voxels at random, and an evaluation pass reads as many; IoU thresholds the output at zero over the whole grid. *Radiance fields.* The eight Blender scenes [3] under FINER's protocol [19], which follows torch-ngp [55]: views downsampled by four to 200×200, every fourth of the 100 training views, every 25th validation view for learning curves, and all 200 test views for the final metrics. A radiance field combines the compared INR, the density network, which outputs a density and 15 geometry features, with a shared color network (two hidden layers of 64, view directions in spherical harmonics); rays are marched through an occupancy grid that skips empty space [10, 56], with at most 1,024 samples per ray and 4,096 rays per step.

Models and training. Table 9 lists every model's configuration, size, and learning rate per task. Each baseline is the smallest of its kind with at least 197,123 (images), 131,329 (volumes), or 74,131 (radiance fields, color network included) parameters, and FUTON takes the largest rank, a multiple of 5, that keeps it smaller than every baseline; on volumes, that cap was TensoRF's CP variant (131,585 parameters), which we ran but report, as in 3D throughout, in its vector–matrix form. FUTON normalizes each axis's features to unit length at every point, $\boldsymbol{\varphi}(x_c) / \|\boldsymbol{\varphi}(x_c)\|_2$, which keeps the separable structure and stabilizes training; the error analysis covers any basis (Thm. 4), so it applies to these features too. FUTON's spectral resolution and head were chosen in a sweep of both on the 24 Kodak images, the five shapes (at 256 voxels per unit length), and the validation views of the eight scenes: the best quality at a training time and memory that stay low. Every model minimizes the

mean squared error with PyTorch's default Adam [38] on the schedule of Sec. 5. Learning rates are the best of the grid $\{10^{-4}, 3 \cdot 10^{-4}, 10^{-3}, \ldots, 3 \cdot 10^{-1}\}$ on two Kodak images, on Thai Statue and Dragon (passing over rates that leave a shape empty), or on the validation views of Lego and Drums, the same grid for every model. Other hyperparameters follow the authors' settings for the task, except where they do not apply or break the budget: RFF's σ, MFN's input scale, α, and β, and F-INR's first frequency were tuned for our coordinates and signal sizes; WIRE, in its real-valued form, uses $3 \cdot 10^{-3}$ on images and volumes, because the official $5 \cdot 10^{-3}$ diverges; Instant-NGP uses 20 levels on images and a 32-wide MLP on volumes to meet the budget, and learning rates of $3 \cdot 10^{-2}$ on images and radiance fields and 10^{-2} on volumes, against 10^{-2} and 10^{-4} in the references; F-INR's widths meet the budget; and radiance fields train for 25,000 steps instead of torch-ngp's 37,500. FreSh [21] selects SIREN's frequency scale for each signal with the authors' procedure, which compares the spectrum of the untrained model's output with the signal's; its search counts toward its training time. All models are compiled with `torch.compile` [57] and trained once, with seed 0.

Metrics and costs. PSNR, SSIM [41], and LPIPS [42] are computed on the whole image or test view, and IoU on the whole voxel grid. Training time is the wall-clock time of the training steps, without evaluation; memory is peak GPU memory in training and in inference; throughput is the signal's size over the median, over a run's evaluations, of its evaluation time, in megapixels or megavoxels per second, or frames per second for radiance fields. Costs are medians over the signals, since other work on a machine can only add time; qualities are means, with one standard error over the signals, computed within signals (Cousineau–Morey) for PSNR, SSIM, and LPIPS, pooled over all models, and plainly for IoU.

Compute. Every run uses one NVIDIA A100 (80 GB) with PyTorch [58] 2.14. The 985 distinct runs reported in the benchmarks and ablations train for 11.3 GPU-hours in total, 6.9 of them on radiance fields, and the error-bound study adds under an hour; learning-rate and configuration searches and preliminary experiments used additional compute.

Table 9: **Model configurations per task**: architecture, parameters, and learning rate. "3×256" is three hidden layers of 256 units; ω_0 and γ_0 are the activations' frequency and scale, L_c is the signal's size along axis c, and M the width of FUTON's MLP head. Radiance-field sizes include the shared color network.

Model	Configuration	#Params (k)	LR
Images			
RFF	3×212; 256 Fourier features, $\sigma = 10$	199.7	$3 \cdot 10^{-3}$
PE-MLP	4×256; positional encoding with 10 frequencies	209.2	$3 \cdot 10^{-3}$
MFN	4×256; input scale 1024, $\alpha = 1.25$, $\beta = 0.125$	204.3	$3 \cdot 10^{-2}$
SIREN	4×256; $\omega_0 = 30$	198.9	10^{-3}
FreSh (SIREN)	as SIREN, ω_0 selected per image	198.9	10^{-3}
Gauss	4×256; $\gamma_0 = 30$	198.9	$3 \cdot 10^{-3}$
WIRE	4×256, real Gabor; $\omega_0 = 20$, $\gamma_0 = 10$	199.3	$3 \cdot 10^{-3}$
FINER	4×256; $\omega_0 = 30$	198.9	10^{-3}
Instant-NGP	20 levels $\times$ 2 features, 2^{13} entries, resolutions 16 to $\max_c L_c/2$; MLP 2×64	205.1	$3 \cdot 10^{-2}$
TensoRF (CP)	rank 176, resolution 256; MLP 2×256	202.0	$3 \cdot 10^{-2}$
GA-Planes	26 features, resolutions 256 and 64; MLP 2×256	199.9	10^{-2}
F-INR (SIREN)	CP rank 141; sub-networks 3×140, first $\omega_0 = 200$	198.8	10^{-3}
FUTON (ours)	cosine, $K_c = L_c/2$; CP, $R = 280$; MLP head, $M = 64$	197.4	10^{-1}
Occupancy volumes			
RFF	2×188; 256 Fourier features, $\sigma = 6$	132.2	$3 \cdot 10^{-3}$
PE-MLP	3×241; positional encoding with 10 frequencies	132.3	10^{-3}
MFN	3×256; input scale 256, $\alpha = 6$, $\beta = 1$	138.0	$3 \cdot 10^{-2}$
SIREN	3×256; $\omega_0 = 30$	132.9	10^{-3}
FreSh (SIREN)	as SIREN, ω_0 selected per shape	132.9	$3 \cdot 10^{-3}$
Gauss	3×256; $\gamma_0 = 30$	132.9	$3 \cdot 10^{-3}$
WIRE	3×256, real Gabor; $\omega_0 = 10$, $\gamma_0 = 40$	133.4	$3 \cdot 10^{-3}$
FINER	3×256; $\omega_0 = 30$	132.9	$3 \cdot 10^{-3}$
Instant-NGP	16 levels $\times$ 2 features, 2^{12} entries, resolutions 16 to 512; MLP 2×32	133.2	10^{-2}
TensoRF (VM)	rank 4, resolution 97; MLP 2×128	132.4	10^{-2}
GA-Planes	25 features, resolutions 128, 32, and 8; MLP 2×128	132.0	$3 \cdot 10^{-2}$
F-INR (SIREN)	CP rank 259; sub-networks 3×96	131.8	10^{-3}
FUTON (ours)	cosine, $K_c = L_c/2$; CP, $R = 215$ to 310 by shape; linear head	128.8–131.4	$3 \cdot 10^{-2}$
Radiance fields (density network)			
RFF	2×180; 90 Fourier features, $\sigma = 2$	74.5	$3 \cdot 10^{-3}$
PE-MLP	3×166; positional encoding with 10 frequencies	75.1	10^{-2}
MFN	3×176; input scale 128, $\alpha = 2$, $\beta = 1$	75.8	10^{-2}
SIREN	3×180; $\omega_0 = 30$	75.2	$3 \cdot 10^{-4}$
FreSh (SIREN)	as SIREN, ω_0 selected per scene	75.2	$3 \cdot 10^{-4}$
Gauss	3×180; $\gamma_0 = 30$	75.2	10^{-3}
WIRE	3×180, real Gabor; $\omega_0 = 20$, $\gamma_0 = 10$	74.5	10^{-3}
FINER	3×180; $\omega_0 = 30$	75.2	$3 \cdot 10^{-4}$
Instant-NGP	16 levels $\times$ 2 features, 2^{11} entries, resolutions 16 to 2048; MLP 1×64	75.0	$3 \cdot 10^{-2}$
TensoRF (VM)	rank 14, resolution 32; MLP 2×128	74.8	$3 \cdot 10^{-2}$
GA-Planes	12 features, resolutions 128, 32, and 4; MLP 2×128	75.0	$3 \cdot 10^{-2}$
F-INR (SIREN)	CP rank 39; sub-networks 3×32	74.7	10^{-3}
FUTON (ours)	cosine, $K_c = 100$; CP, $R = 180$; MLP head, $M = 64$	73.0	10^{-2}

Table 10: **Time to target quality**, in seconds, read from each model's mean learning curve, computed as in Fig. 4 and interpolated linearly between evaluations, with the best bold and the second best underlined; "–" marks a target the mean curve never reaches, and "≤" one reached by the first evaluation. Radiance fields use the validation views.

	Images, PSNR (dB)			Occupancy, IoU (%)			Radiance fields, PSNR (dB)		
Model	30	33	35	98	99	99.5	27	28	28.5
RFF	13.7	–	–	11.8	23.1	37.1	49.3	117	–
PE-MLP	–	–	–	–	–	–	63.5	172	–
MFN	≤2.9	6.7	14.6	79.1	–	–	42.5	<u>109</u>	267
SIREN	12.5	22.9	–	35.9	–	–	79.8	132	176
FreSh (SIREN)	9.8	18.5	–	233	251	–	78.8	141	203
Gauss	14.6	–	–	33.8	55.4	–	139	–	–
WIRE	27.0	–	–	29.2	56.1	–	104	169	–
FINER	6.0	14.0	20.5	43.8	57.8	–	74.7	127	<u>167</u>
Instant-NGP	<u>0.3</u>	<u>0.8</u>	<u>1.6</u>	<u>1.4</u>	<u>4.0</u>	<u>7.2</u>	–	–	–
TensoRF	2.8	7.2	15.0	3.7	11.1	–	59.1	205	–
GA-Planes	5.4	–	–	6.0	22.6	44.5	<u>37.3</u>	121	–
F-INR (SIREN)	–	–	–	–	–	–	152	–	–
FUTON (ours)	**≤0.2**	**0.4**	**0.9**	**0.5**	**2.0**	**3.2**	**≤16.1**	**22.0**	**33.3**

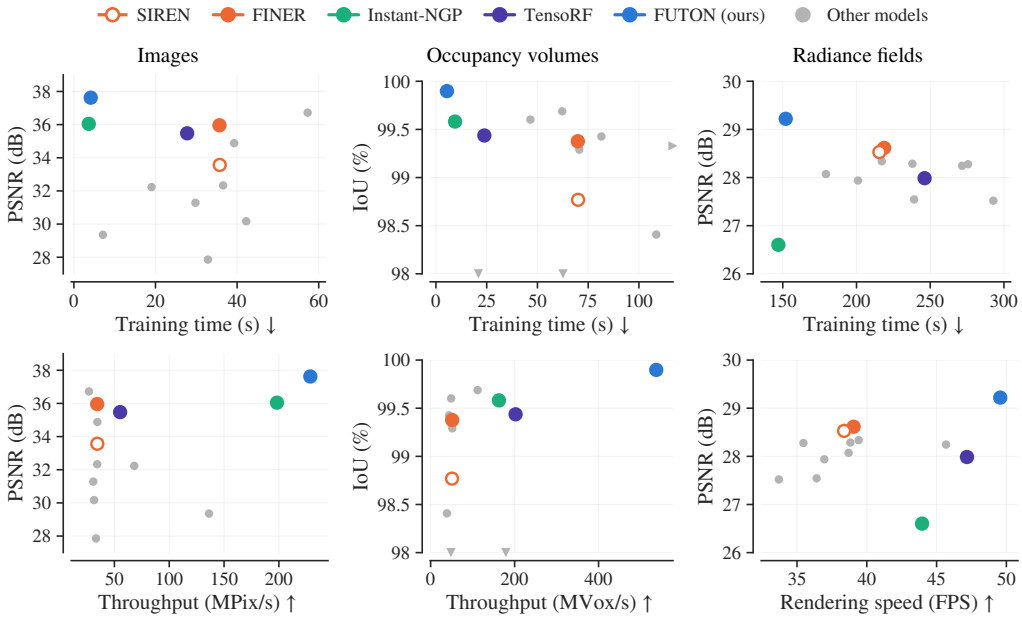


Figure 13: **Quality against training time** (top) **and inference speed** (bottom) for every model; gray dots are the other baselines. On the volumes, the axes start at 98% IoU, and models below sit on the axis line; a model past an axis's range sits at its end as a marker.

H Full Results

Time to target. FUTON reaches every target first (Tab. 10): 35 dB on images in 0.9 s, 99.5% IoU on volumes in 3.2 s, and 28.5 dB on radiance fields in 33 s. FUTON reaches the final quality of Instant-NGP, the fastest baseline, 2.7 times sooner on images (in 1.3 s) and volumes (3.4 s), and at least 9.1 times sooner on radiance fields, by its first evaluation (16 s); and that of each task's most accurate baseline 34 times sooner on images (MFN, 1.7 s against 57 s), 16.5 times on volumes (GA-Planes), and 4.2 times on radiance fields (FINER).

Table 11: **Images**, means over the 24 Kodak images ± one standard error, and median costs, at about 197k parameters (Tab. 9 lists each model's size). The best results are **bold**, and the second-best are underlined.

Model	Training		Inference		PSNR	SSIM	LPIPS
	Time (s) ↓	Memory (GB) ↓	Throughput (MPix/s) ↑	Memory (MB) ↓	(dB) ↑	↑	↓
RFF	29.8	2.08	30.7	1208	31.28±0.17	0.846±0.007	0.271±0.007
PE-MLP	32.8	2.11	33.2	805	27.86±0.16	0.726±0.015	0.431±0.008
MFN	57.3	7.67	26.8	3224	36.72±0.17	0.938±0.003	0.129±0.003
SIREN	35.7	3.25	34.4	805	33.57±0.15	0.906±0.003	0.192±0.004
FreSh (SIREN)	39.3	3.25	34.5	805	34.88±0.15	0.922±0.003	0.150±0.003
Gauss	36.6	3.64	34.3	805	32.33±0.19	0.877±0.005	0.213±0.004
WIRE	42.2	4.00	31.5	856	30.17±0.46	0.807±0.017	0.260±0.010
FINER	35.7	3.25	34.3	805	35.96±0.14	0.938±0.003	0.120±0.002
Instant-NGP	**3.6**	0.72	198.3	503	36.05±0.20	0.932±0.002	0.142±0.004
TensoRF	27.8	1.79	55.2	805	35.48±0.13	0.920±0.002	0.161±0.003
GA-Planes	19.0	1.32	68.1	805	32.23±0.20	0.878±0.004	0.227±0.005
F-INR (SIREN)	7.1	2.70	136.3	1338	29.35±0.13	0.786±0.007	0.343±0.005
FUTON (ours)	4.1	**0.36**	**228.7**	**201**	**37.63**±0.55	**0.955**±0.004	**0.098**±0.004

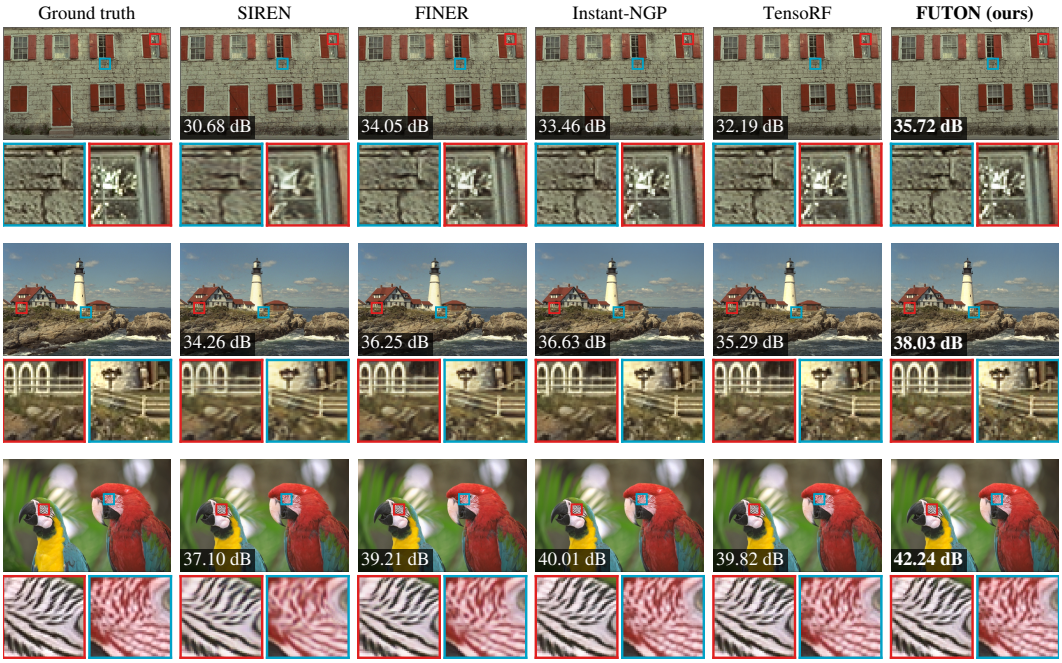


Figure 14: **Image fitting** on kodim01, kodim21, and kodim23, as in Fig. 5.

Table 12: **Occupancy volumes**: IoU on every shape and its mean ± one standard error, and median costs, at about 131k parameters. The best results are **bold**, and the second-best are underlined.

Model	Training		Inference		IoU (%) ↑					
	Time (s) ↓	Memory (GB) ↓	Throughput (MVox/s) ↑	Memory (GB) ↓	Armadillo	Dragon	Happy Buddha	Lucy	Thai Statue	Mean
RFF	46.5	4.15	48.9	3.33	99.71	99.65	99.59	99.78	99.28	99.60±0.09
PE-MLP	62.6	4.09	48.3	2.14	96.96	96.92	97.54	91.96	95.71	95.82±1.01
MFN	108.5	15.07	38.9	5.49	99.23	97.47	97.50	99.10	98.74	98.41±0.39
SIREN	70.0	6.48	51.0	2.26	98.78	99.18	99.41	98.53	97.94	98.77±0.26
FreSh (SIREN)	263.6	6.48	51.3	2.26	99.34	99.67	99.65	99.15	98.83	99.33±0.16
Gauss	70.7	6.48	51.5	2.26	99.52	99.59	99.70	99.07	98.57	99.29±0.21
WIRE	81.5	7.62	44.2	2.39	99.52	99.59	99.74	99.43	98.84	99.43±0.15
FINER	69.9	6.48	51.3	2.26	99.45	99.59	99.70	99.29	98.85	99.38±0.15
Instant-NGP	9.4	2.11	162.6	2.93	99.50	99.59	99.69	99.78	99.35	99.58±0.08
TensoRF	23.8	1.75	202.3	1.19	99.53	99.71	99.78	99.34	98.82	99.44±0.17
GA-Planes	62.2	2.31	111.9	1.37	99.58	99.79	99.93	99.79	99.34	99.69±0.10
F-INR (SIREN)	20.9	6.59	179.2	3.39	97.75	98.25	98.08	96.27	96.42	97.35±0.42
FUTON (ours)	**5.4**	**0.28**	**537.4**	**0.61**	**99.86**	**99.94**	**99.99**	**99.96**	**99.74**	**99.90**±0.04

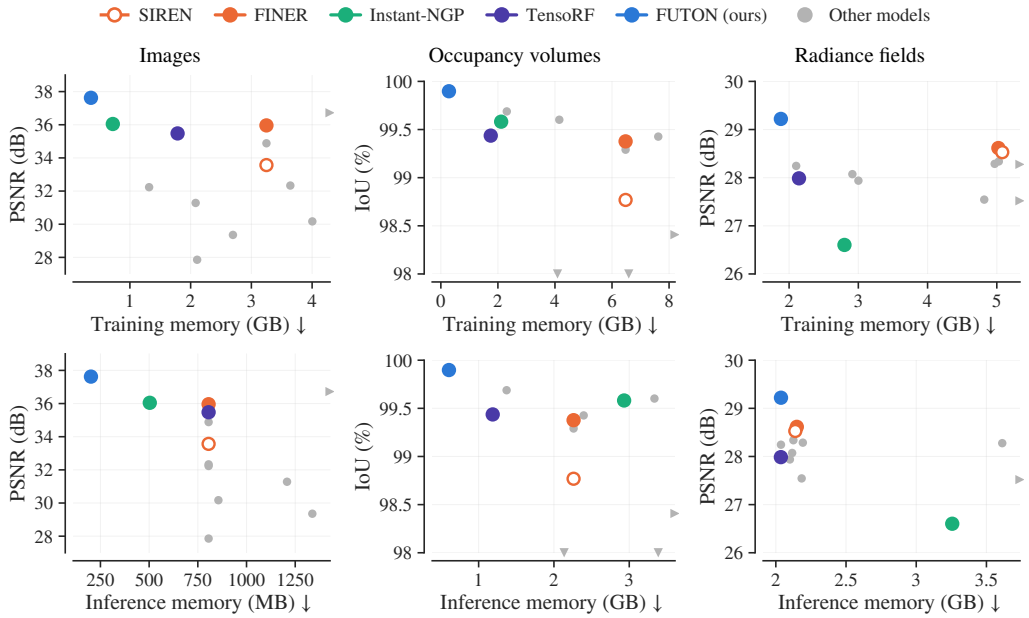


Figure 15: **Quality against training memory** (top) **and inference memory** (bottom), as in Fig. 13.



Figure 16: **Occupancy volumes** on Armadillo, Dragon, Happy Buddha, and Thai Statue, as in Fig. 6.

Table 13: **Radiance fields** at about 74k parameters: median costs and means over the eight Blender scenes ± one standard error (top); per scene, the PSNR of the baselines Tab. 3 omits and every model's SSIM and LPIPS (bottom), marked among all models; 200 test views per scene.

	Training		Inference		Mean over the scenes		
Model	Time (s) ↓	Memory (GB) ↓	Speed (FPS) ↑	Memory (GB) ↓	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
RFF	179.3	2.91	38.7	2.12	28.07±0.30	0.928±0.004	0.082±0.008
PE-MLP	201.0	3.00	37.0	2.10	27.94±0.22	0.931±0.003	0.079±0.006
MFN	275.6	9.97	35.5	3.61	28.28±0.16	0.932±0.002	0.078±0.006
SIREN	215.5	5.08	38.4	2.14	28.53±0.16	0.935±0.002	0.106±0.007
FreSh (SIREN)	217.2	5.03	39.4	2.13	28.34±0.23	0.928±0.004	0.146±0.016
Gauss	239.1	4.82	36.4	2.18	27.54±0.21	0.922±0.003	0.105±0.008
WIRE	237.9	4.97	38.8	2.19	28.29±0.19	0.928±0.003	0.126±0.007
FINER	218.8	5.02	39.1	2.15	28.62±0.19	0.934±0.002	0.120±0.009
Instant-NGP	**147.0**	2.80	44.0	3.26	26.60±0.37	0.908±0.009	0.119±0.011
TensoRF	246.2	2.14	47.2	**2.04**	27.99±0.18	0.931±0.003	0.076±0.007
GA-Planes	271.6	2.10	45.7	**2.04**	28.24±0.19	0.936±0.003	0.071±0.005
F-INR (SIREN)	292.8	16.18	33.7	6.69	27.52±0.47	0.911±0.012	0.127±0.020
FUTON (ours)	152.0	**1.88**	**49.6**	**2.04**	**29.22**±0.24	**0.943**±0.004	**0.065**±0.007

Model	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship
PSNR (dB) ↑, the baselines Tab. 3 omits								
PE-MLP	32.09	22.84	25.75	32.06	28.55	26.64	32.44	23.12
Gauss	30.92	23.69	26.41	32.07	27.66	25.21	32.46	21.94
WIRE	32.63	24.18	27.54	32.00	28.44	26.14	33.23	22.13
SSIM ↑								
RFF	0.975	0.901	0.959	**0.967**	0.935	0.922	0.982	0.783
PE-MLP	0.970	0.896	0.946	0.960	0.943	0.934	0.980	0.816
MFN	0.973	0.908	0.957	0.964	0.943	0.920	0.980	0.813
SIREN	0.972	**0.916**	0.955	0.965	0.943	0.930	**0.983**	0.817
FreSh (SIREN)	0.970	0.901	0.952	0.964	0.952	0.897	0.979	0.806
Gauss	0.961	0.902	0.950	0.958	0.927	0.913	0.980	0.785
WIRE	0.970	0.911	0.957	0.957	0.933	0.921	0.982	0.793
FINER	0.972	**0.916**	0.956	0.962	0.952	0.922	**0.983**	0.809
Instant-NGP	0.967	0.894	0.865	0.953	0.925	0.907	0.978	0.779
TensoRF	0.968	0.908	0.942	**0.967**	0.935	0.933	0.980	0.818
GA-Planes	0.973	0.910	0.944	**0.967**	0.956	0.931	0.979	0.826
F-INR (SIREN)	0.973	0.850	0.904	0.960	0.952	0.941	0.976	0.731
FUTON (ours)	**0.976**	0.913	**0.960**	0.966	**0.970**	**0.948**	0.982	**0.827**
LPIPS ↓								
RFF	0.036	0.109	**0.046**	0.045	0.084	0.086	**0.026**	0.227
PE-MLP	0.039	0.122	0.059	0.053	0.060	0.072	0.029	0.195
MFN	0.040	0.102	0.056	0.044	0.070	0.084	0.033	0.192
SIREN	0.052	0.130	0.163	0.052	0.091	0.116	0.043	0.205
FreSh (SIREN)	0.076	0.191	0.210	0.056	0.086	0.230	0.087	0.228
Gauss	0.059	0.129	0.071	0.065	0.088	0.143	0.043	0.241
WIRE	0.058	0.146	0.132	0.079	0.133	0.158	0.051	0.249
FINER	0.062	0.146	0.187	0.073	0.075	0.149	0.040	0.228
Instant-NGP	0.047	0.130	0.196	0.069	0.085	0.132	0.041	0.253
TensoRF	0.041	**0.094**	0.065	**0.042**	0.075	0.070	0.032	0.185
GA-Planes	0.034	0.097	0.063	**0.042**	0.045	0.077	0.031	**0.179**
F-INR (SIREN)	0.052	0.226	0.160	0.062	0.061	0.063	0.063	0.331
FUTON (ours)	**0.033**	**0.094**	0.052	0.044	**0.030**	**0.057**	0.030	0.182

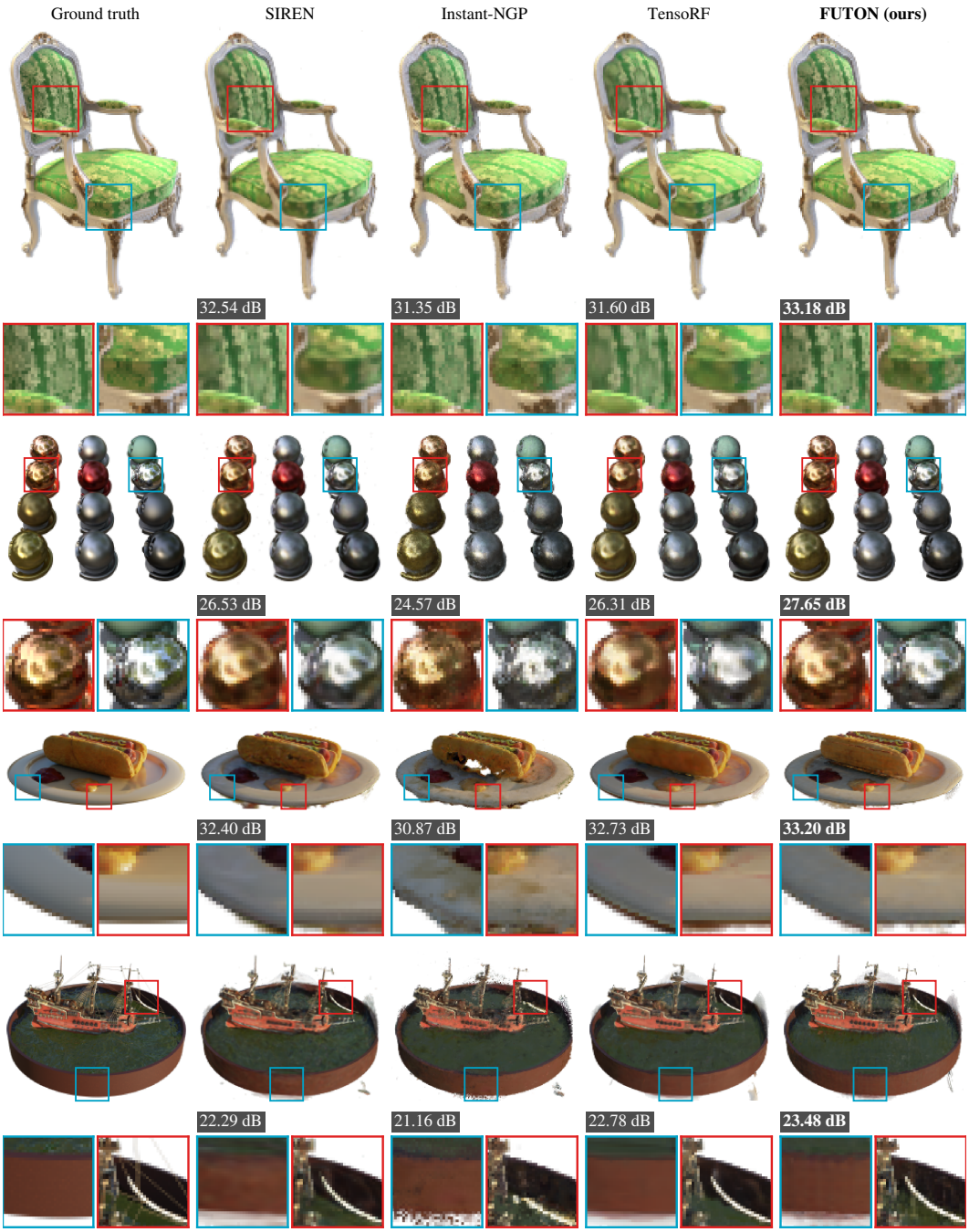
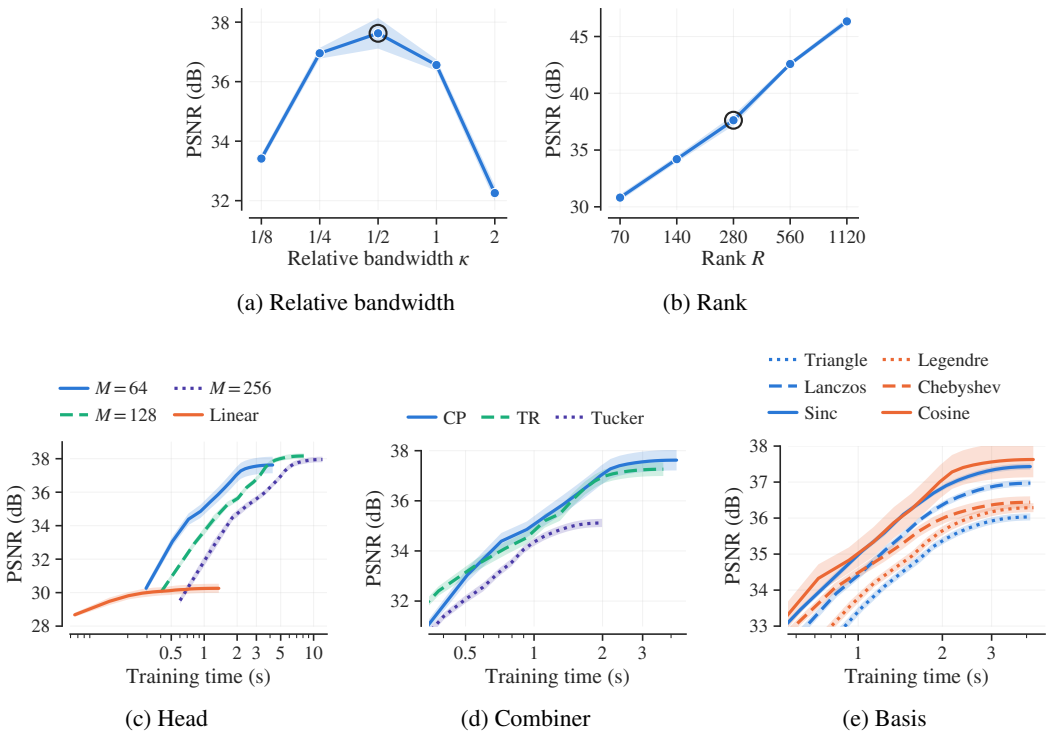


Figure 17: **Novel view synthesis** on Chair, Materials, Hotdog, and Ship, as in Fig. 7.



(a) Relative bandwidth (b) Rank

(c) Head (d) Combiner (e) Basis

Figure 18: **Ablations on Kodak**, mean PSNR ± one standard error, as in Tab. 14: (a) relative bandwidth and (b) rank, the defaults circled; (c–e) PSNR during training for four heads, three combiners, and six bases.

I Ablation Studies

We ablate FUTON's main design choices on the 24 Kodak images, changing one at a time from the image benchmark's FUTON: cosine basis, $\kappa = 1/2$, CP combiner with $R = 280$, and an MLP head of width $M = 64$, 197k parameters, trained as in App. G. Except in the rank study, every variant keeps the size by adjusting the rank. Table 14 reports the results, and Fig. 18 the curves.

Relative bandwidth. At a fixed size, a larger κ resolves higher frequencies but leaves a smaller rank (Tab. 14). The best PSNR is at $\kappa = 1/2$, and $\kappa = 1/4$ and 1 stay within 1.1 dB; $\kappa = 1/8$ truncates the images (4.2 dB lower), and $\kappa = 2$ leaves rank 75 (5.4 dB lower). Training time falls as κ grows, with the rank.

Rank. At $\kappa = 1/2$, PSNR grows by 3.4, 3.4, 5.0, and 3.8 dB per doubling of R from 70 to 1,120, with no plateau at these sizes, while training time grows from 2.2 to 10.9 s (Tab. 14).

Head. A linear head of the same size, with $R = 305$, scores 30.26 dB, 7.4 dB below the MLP head (Tab. 14). Wider MLP heads at equal size trade rank for width: $M = 128$ gains 0.5 dB on average and $M = 256$ 0.3 dB, at about two and three times the training time and three times the memory, so we keep $M = 64$.

Combiner. At about equal size (the ring holds 180k parameters, as its next rank exceeds the budget), the CP combiner leads a tensor ring, the cyclic form of a tensor train [30], by 0.4 dB and Tucker [29] by 2.5 dB (Tab. 14). On a grid, both alternatives train faster, Tucker in half the time, but neither reaches CP's quality.

Basis. Among six bases at equal size, cosines are the most accurate (37.63 dB), ahead of sinc (37.43 dB) and Lanczos (36.97 dB), shifted kernels on a uniform grid, and of the Chebyshev (36.44 dB)

Table 14: **Ablations on Kodak**: means over the 24 images ± one standard error, and median costs, with the best and second best marked within each block. Top: the relative bandwidth κ at about 197k parameters, the rank adjusted to keep the size, and the rank R at $\kappa = 1/2$, from 49.5k to 789k parameters. Bottom: at equal size, MLP heads of width $M = 64$, 128, and 256 and a linear head, each with the largest rank that keeps the size; CP, a tensor ring (TR, 180k parameters), and Tucker, at about equal size; and six bases.

		Training		Inference				
κ	R	Time (s) ↓	Memory (MB) ↓	Throughput (MPix/s) ↑	Memory (MB) ↓	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
Relative bandwidth, at about 197k parameters								
1/8	885	9.0	442	111.9	219	33.41±0.15	0.914±0.003	0.167±0.004
1/4	515	6.1	392	156.9	**201**	36.95±0.18	0.948±0.003	0.118±0.004
1/2	280	4.1	359	228.7	**201**	**37.63**±0.51	**0.955**±0.003	**0.098**±0.003
1	145	2.8	341	254.1	**201**	36.56±0.19	0.936±0.002	0.140±0.003
2	75	**2.2**	**331**	**335.5**	**201**	32.26±0.26	0.860±0.007	0.271±0.008
Rank, at $\kappa = 1/2$								
1/2	70	**2.2**	**330**	**305.8**	**201**	30.81±0.28	0.826±0.010	0.304±0.008
1/2	140	2.7	340	249.8	**201**	34.20±0.29	0.903±0.003	0.190±0.004
1/2	280	4.1	359	228.7	**201**	37.63±0.46	0.955±0.003	0.098±0.003
1/2	560	6.2	398	152.0	**201**	42.58±0.20	0.983±0.005	0.040±0.004
1/2	1120	10.9	473	95.2	251	**46.33**±0.30	**0.992**±0.005	**0.016**±0.005

	Training		Inference				
Variant	Time (s) ↓	Memory (MB) ↓	Throughput (MPix/s) ↑	Memory (MB) ↓	PSNR (dB) ↑	SSIM ↑	LPIPS ↓
Head							
M=64 (R=280)	4.2	359	248.6	201	37.62±0.49	**0.955**±0.002	**0.098**±0.003
M=128 (R=255)	8.1	1024	131.7	602	**38.16**±0.15	0.954±0.002	0.099±0.001
M=256 (R=220)	12.0	1176	90.4	749	37.95±0.17	0.951±0.002	0.106±0.002
Linear (R=305)	**1.4**	**37**	**346.1**	**19**	30.26±0.27	0.874±0.006	0.201±0.005
Combiner							
CP	4.2	359	203.5	**201**	**37.62**±0.41	**0.955**±0.001	**0.098**±0.002
TR	3.7	356	216.9	**201**	37.26±0.27	0.947±0.001	0.112±0.002
Tucker	**2.0**	**330**	**325.3**	**201**	35.12±0.16	0.915±0.002	0.170±0.003
Basis							
Cosine	4.2	**359**	230.4	**201**	**37.63**±0.49	**0.955**±0.001	**0.098**±0.002
Chebyshev	**4.1**	**359**	230.3	**201**	36.44±0.16	0.939±0.001	0.124±0.002
Legendre	4.2	360	210.4	**201**	36.29±0.12	0.937±0.001	0.129±0.001
Triangle	**4.1**	**359**	242.6	**201**	36.03±0.10	0.934±0.001	0.126±0.001
Lanczos	**4.1**	**359**	223.4	**201**	36.97±0.09	0.944±0.001	0.112±0.001
Sinc	**4.1**	**359**	**248.4**	**201**	37.43±0.09	0.949±0.001	0.108±0.001

and Legendre (36.29 dB) polynomials and the triangle (hat) functions (36.03 dB). All train in about the same time (Tab. 14).

J Limitations

While FUTON demonstrates strong performance across the tasks considered, we identify several limitations and directions for improvement:

1. **Separability assumption.** FUTON's basis functions are constructed as separable products of one-dimensional functions (2). While this is theoretically complete (Thm. 1) and practically effective for the tasks we evaluate, signals with strong cross-dimensional correlations (*e.g.*, oriented textures at arbitrary angles) may be less efficiently represented than with non-separable bases. The CP rank R compensates for this by mixing dimensions, but at increased parameter cost: an oblique edge needs a rank that grows with its sharpness (Tab. 4).

2. **Low-rank assumption.** FUTON assumes that the coefficient tensor $\mathcal{W}$ admits an effective low-rank CP approximation. For signals whose Fourier spectra lack low-rank structure (*e.g.*, white noise or highly irregular textures), the required rank R may become large, diminishing the computational advantage over the full-rank GFS. Proposition 1 measures the rank a signal needs, and Prop. 3 shows that no signal-independent bound can do better than Thm. 3's order, so the assumption must be checked on the data at hand.
3. **Scaling to high dimensions.** While FUTON's complexity is linear in C for fixed R and K_c, the sufficient rank for universal approximation grows as $R \leq K^{C-1} \min\{K, D\}$ (Thm. 3). For very high-dimensional inputs ($C \gg 3$), the practical rank required for accurate representation remains an open question. Our experiments cover $C \in \{2, 3\}$; validation on higher-dimensional tasks (*e.g.*, video, 4D light fields) is left for future work.
4. **Single-scene optimization.** Like other coordinate-based INRs, FUTON is optimized per scene and does not generalize across scenes without retraining. Meta-learning or hypernetwork-based extensions could address this but are beyond the current scope.
5. **Evaluation scope.** Our experiments cover images, occupancy volumes, and radiance fields against twelve baselines, with one seed per run. Broader evaluation on additional signal modalities (*e.g.*, audio, signed distance fields) and on inverse problems would provide a more comprehensive picture of FUTON's relative strengths.
6. **Scope of the error analysis.** Proposition 1 covers the linear head on a fully sampled signal: the coefficient tensor is computed from every sample, so radiance fields and inverse problems, where the signal is observed through a renderer or an operator, are outside it, and so are the MLP head, to which only the upper bound transfers, and the $\tanh$ output, which Rem. 1 covers only for targets inside $(-1, 1)$, unlike the benchmarks'. On a grid, its norm says nothing off the grid. For three or more axes, the lower bound falls to the truncation once R passes the largest K_c, so the bracket is wide on volumes; a tighter lower bound for that regime is open, and a search that improves on the certified FUTON's terms would tighten the upper one.
7. **Implementation-dependent speed.** Wall-clock comparisons depend on implementations. All models here run in PyTorch, compiled; hash-grid and factorized models with fused CUDA kernels, as in their authors' code, can train faster than in our implementation.

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: The abstract and introduction state the contributions: the FUTON architecture (Sec. 4.1), its complexity (Sec. 4.2) and error analysis (Sec. 4.3, App. F), and experiments on three tasks against twelve baselines (Sec. 5). The claims match the results in Tabs. 1 to 3, Figs. 4 to 7, and Apps. F.3, H and I, and their scope is stated in App. J.

   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: The paper includes a dedicated "Limitations" section in the supplementary material (App. J), which discusses the separability assumption, low-rank assumption, scaling to high dimensions, and the single-scene optimization paradigm, as well as the evaluation scope (one seed per run), the scope of the error analysis, and the dependence of wall-clock comparisons on implementations; the conclusion points to them.

   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: For each theoretical result (Thms. 1 to 5, Props. 1, 2 and 3, Cor. 1, and the lemmas), the paper provides a full set of assumptions and refers to complete proofs in the supplementary material (Apps. C to E, App. F).

   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: The paper provides extensive details for reproducibility, including model configurations, hyperparameters, optimizer settings, and loss functions in Sec. 5 and App. G of the supplementary material, with every model's configuration in Tab. 9.

   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: The code, the library Inro (MIT license), is public at `https://github.com/pashtari/inro`, with the configuration files, training scripts, and instructions to download the public datasets and rerun every experiment (App. G).

   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: All training and test details, including data splits, hyperparameters, optimizer choices, and model architectures, are specified in Sec. 5 and further detailed in App. G of the supplementary material.

   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: Qualities are reported as means over the signals of each task with one standard error, computed within signals (Cousineau–Morey) for PSNR, SSIM, and LPIPS and plainly for IoU, in the tables and as bands in the figures (App. G); costs are medians. Each configuration is trained once, with one seed, so the errors capture variability across signals, not across seeds.

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: [Yes]

   Justification: Every run uses one NVIDIA A100 (80 GB) (Sec. 5); training time, memory, and throughput are reported for every model and task (App. H), and the total compute, 11.3 GPU-hours of training for the 985 reported runs plus under an hour for the error-bound study, with additional compute for searches and preliminary experiments, in App. G.

   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: The research adheres to the NeurIPS Code of Ethics. The work is based on publicly available datasets and does not involve human subjects or sensitive data.

   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: [No]

    Justification: The paper does not include a discussion of broader societal impacts. The work is foundational research on signal representation, and direct societal impacts are not immediately apparent.

    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: The paper does not release new models or datasets that have a high risk for misuse. The research focuses on a novel architecture for representing signals.

    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: The paper uses public datasets, the Kodak images, the Stanford 3D Scanning Repository, and NeRF's Blender scenes, cites their sources (Sec. 5), and uses them for non-commercial research as their providers permit; baselines are cited and follow their authors' published settings (Tab. 9), except for the deviations listed in App. G.

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: The new asset is the code, the library Inro, released under the MIT license at `https://github.com/pashtari/inro` with documentation and installable from PyPI.

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: The research does not involve crowdsourcing or research with human subjects.

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: The research does not involve human subjects, so no IRB approval was required.

    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: The core methodology of this research does not involve the use of Large Language Models (LLMs).

    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.