

BANF: Band-limited Neural Fields for Levels of Detail Reconstruction

Anonymous CVPR submission

Paper ID 4179

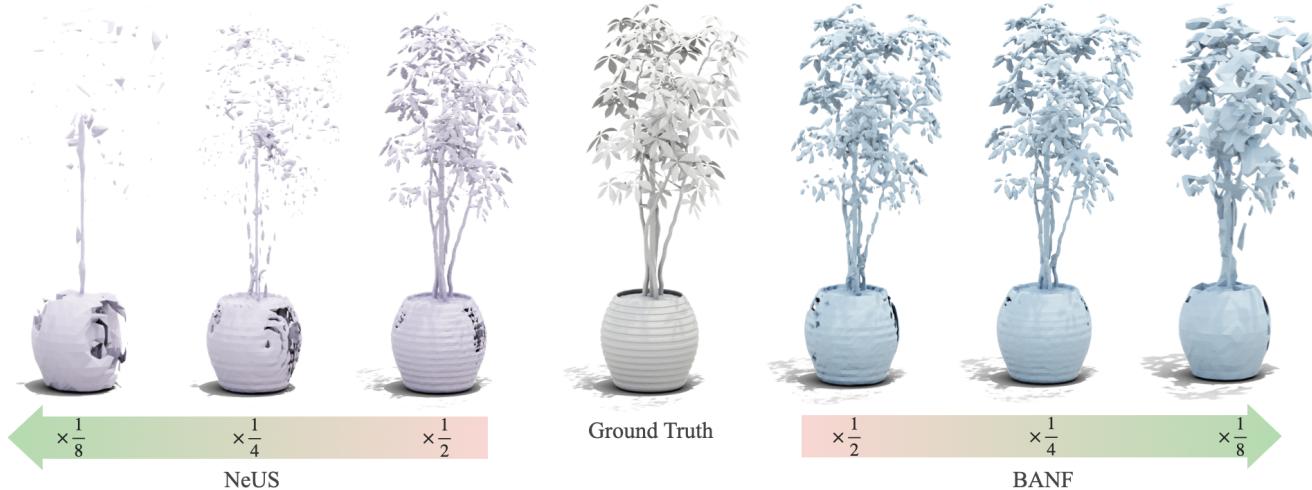


Figure 1. We introduce BANF, a method for band-limited frequency decomposition in neural fields. Our minimal yet impactful enhancements to the training process achieves anti-aliased coarse-to-fine signal reconstruction, seamlessly integrating with commonly used mesh extraction techniques [25], significantly surpassing prior methods in quality. The reconstruction artifacts due to view-dependent effects (i.e. vase bowl not being smooth) are due to the reconstruction baseline we employ, and is orthogonal to our method.

Abstract

001 *Largely due to their implicit nature, neural fields lack a di-
002 rect mechanism for filtering, as Fourier analysis from dis-
003 crete signal processing is not directly applicable to these
004 representations. Effective filtering of neural fields is critical
005 to enable level-of-detail processing in downstream ap-
006 plications, and support operations that involve sampling the
007 field on regular grids (e.g. marching cubes). Existing meth-
008 ods that attempt to decompose neural fields in the frequency
009 domain either resort to heuristics, or require extensive mod-
010 ifications to the neural field architecture. We show that via
011 a simple modification, one can obtain neural fields that are
012 low-pass filtered, and in turn show how this can be exploited
013 to obtain a frequency decomposition of the entire signal.
014 We demonstrate the validity of our technique by investigat-
015 ing level-of-detail reconstruction, and showing how coarser
016 representations can be computed effectively.¹*

¹We include a copy of the source code in the supplementary, which will later be posted at <https://banf-project.github.io>

1. Introduction

Recent research enabled neural rendering to accurately model 3D scenes from multi-view images alone. Many neural rendering techniques employ *neural fields* as the underlying 3D representation – coordinate neural networks that can represent a variety of signals including images [27, 37], surface manifolds [32, 42, 48], and volumetric densities [29]. These representations are now used in a plethora of applications that include robotics [1], semantic understanding [20], scene capture [22], editing [14], and generative modeling from natural language [33].

While neural fields are most commonly trained to represent the *entire* spectrum of signals, it is an established practice in classical signal processing to decompose the signal into separate frequency bands due to the many advantages it brings [2]. For example, signal decomposition allows 3D objects to be represented at multiple levels of detail, which

034 finds application in reducing the computational cost of ren-
 035 dering or physics-based simulations [26]. In classical signal
 036 processing, frequency decomposition can be studied via the
 037 Fourier transform [31]. Signals can be decomposed through
 038 band-pass filtering, and later recombined through linear super-
 039 position. One could apply these classical techniques
 040 by first *sampling* neural fields on a regular lattice, but the
 041 Nyquist theorem tells us that signals ought to be properly
 042 filtered before being sampled so to avoid aliasing [31]. This
 043 is particularly critical for methods that target surface recon-
 044 struction [22], as these techniques often rely on marching
 045 cubes to extract manifolds from fields [25], and this process
 046 involves sampling neural fields on *regular* grids; see Fig. 1
 047 for an example.

048 While monte-carlo filtering of neural fields would make
 049 anti-aliased sampling possible [9], a critical question is
 050 whether filtering could be realized *directly* at training
 051 time to minimize its impact on inference, rather than
 052 *a-posteriori* as distillation [13]. This can be achieved
 053 when neural fields are implemented as fully-connected net-
 054 works (MLPs). One can build over the theory of multiplica-
 055 tive filter networks [11] to realize signal decomposition for
 056 neural fields [10, 23, 46]. However, due to *significant* gains
 057 in training performance, most recent methods do not imple-
 058 ment neural fields as large MLP networks, but rather via hy-
 059 brid models combining interpolated feature grids with small
 060 MLPs mapping features to the field co-domain – this ren-
 061 ders the theory of multiplicative filter networks to hybrid
 062 neural fields not applicable. One exception is Wu et al. [43],
 063 but while they introduce a pyramid-like architecture, their
 064 main outcome is a compressed representation, and extract-
 065 ing signals at intermediate scales is simply not possible.

066 Interestingly, early research in neural fields exploited the
 067 duality of time and frequency scaling to realize (explicit)
 068 multi-scale training architectures [24, 39].² However, these
 069 methods require the multi-resolution structure of the signal
 070 to either be specified *a-priori* [39], or by monitoring the
 071 training process via heuristics [24]. Conversely, our tech-
 072 nique can be applied to any neural field, without any modi-
 073 fication to the underlying architecture, and without any as-
 074 sumption about the training data.

075 Our core insight is that regularly *sampling* a field, and
 076 then *interpolating* this field with a band-limited kernel can
 077 be seen as a low-pass filtering operation. We can then build
 078 band-pass signals by compositing these filters, which allows
 079 us to derive a suitable coarse-to-fine hierarchical training
 080 scheme. We demonstrate the validity of our method across
 081 domains (1D, 2D, 3D), and most importantly on 3D repre-
 082 sentations trained from 2D observations. We show how fil-
 083 tered representations can be extracted at any of the interme-

²Interestingly, as we will see later, bi-linear interpolation does execute a form of filtering, but this filtering affects the field feature space, rather than, as we desire, in the field co-domain.

084 diate scales at a fidelity that is superior to that of reasonable
 085 baselines. Further, these representations can be aggregated,
 086 matching the performance of representations that were di-
 087 rectly trained at high-resolution only.

2. Related work

088 Our work is built on techniques in signal processing and
 089 neural fields. We further review how signals in neural fields
 090 can be spatially decomposed, frequency decomposed by po-
 091 sitional encoding, and frequency decomposed by network
 092 design.

093 **Frequency decomposition by signal processing.** In tradi-
 094 tional signal processing, signals are often converted to the
 095 frequency domain with the Fourier Transform [31]. In the
 096 frequency domain, lower frequencies represent smoother
 097 portions of the signal while high frequencies represent finer
 098 details. The frequency domain allows the signal to be
 099 separated into sub-bands to capture different levels of de-
 100 tail (LODs) [17, 31]. Unfortunately, there is no analyti-
 101 cal Fourier Transform equivalent for signals represented as
 102 neural fields. A signal can be decomposed into coarse-to-
 103 fine LODs by convolution with appropriate kernels. How-
 104 ever, since neural fields are continuous functions, convo-
 105 lutions must be executed as a monte-carlo integration [9]
 106 whose result can be distilled in a secondary neural field [13].
 107 Conversely, we propose a simple method to directly gen-
 108 erate low-pass and band-pass signals during field optimiza-
 109 tion, with a minimal modification in the training procedure.

110 **Neural fields.** As a form of signal representation, neural
 111 fields have recently gained substantial importance [45].
 112 Neural fields can effectively represent many signals, in-
 113 cluding 2D images [27, 37], and 3D geometry in the
 114 form of Signed Distance Functions [32] and Occupancy
 115 Fields [28]. Arguably the most well-known example of
 116 neural fields is neural radiance fields, enabling novel view
 117 synthesis through volumetric [29] or surface [42] render-
 118 ing. Classical neural fields [29, 32] use a multilayer per-
 119 ceptron (MLP) to map an input coordinate to a signal value.
 120 Non-neural variants that optimize explicit feature grids later
 121 emerged [12, 18], sacrificing slightly worse reconstructions
 122 for faster training speed. Hybrid models combining explicit
 123 representations (grid [38], hash table [30], K-Planes [7])
 124 with a small MLP finally emerged, achieving both fast
 125 training speed and high quality reconstruction. Although
 126 there are many variants of neural field architectures, our
 127 sampling-based strategy can be applied on top of *any* ar-
 128 chitecture.

129 **Multi-scale fields by spatial decomposition.** Neural
 130 fields can be trained with spatial decomposition to pro-
 131 duce coarse-to-fine LODs, where varying levels of detail
 132 are modeled with multi-resolution grids such as octrees [39]

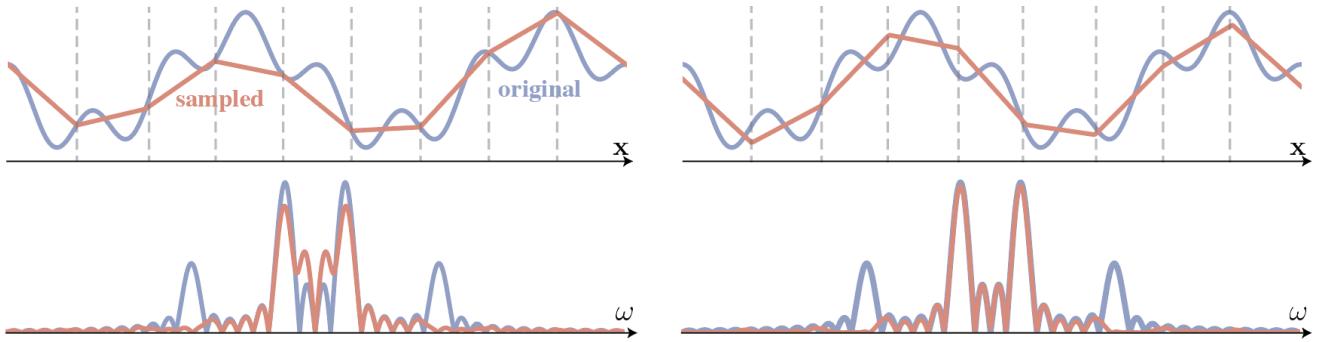


Figure 2. **Filtering by optimization** – We compare two approaches to generate a low-pass version of a neural field’s signal. Left shows training a neural field to a 1D signal, and sampling the result a-posteriori on a regular lattice. On the right, our approach trains the neural field in a way that is “sampling aware”, as the field is expressed as the (linear) interpolation of sampled field values. In other words, our method executes low-pass filtering *during* optimization. At the bottom we visualize the signal’s spectra, where the baseline’s reconstruction is clearly affected by aliasing, while our reconstruction is clearly low-pass filtered.

134 or multi-resolution tri-planes [49]. Adaptive data structures
 135 allow for increased resolution in high-frequency regions,
 136 which can be identified for further subdivision using indicators
 137 such as high local training error [27] or high density [24]. Saragadam et al. [35] suggest a memory-efficient
 138 approach using separate MLPs for data patches when multi-
 139 scale supervision available. Such methods are dependent on
 140 *complex and heuristic-based* spatial subdivision strategies.
 141 Moreover, spatial decomposition methods neglect their effect
 142 on the frequency domain, resulting in LODs that are
 143 not explicitly frequency band-limited. Our method does not
 144 suffer from these limitations, as we provide a *simple* training
 145 scheme directly providing a frequency decomposition.
 146

147 **Frequency decomposition by positional encoding.** Tancik et al. [40] showed that mapping coordinates to Fourier
 148 features overcomes spectral bias in MLPs [34], facilitating
 149 learning of high-frequency details in neural fields. Further,
 150 Mildenhall et al. [29] proposed positional encoding, a special
 151 case of Fourier feature encoding, can greatly boosts
 152 reconstruction quality for novel view synthesis. Barron
 153 et al. [4] extended positional encoding to *integrated* pos-
 154 *155*itional encoding to enable *anti-aliased* multi-scale neural
 156 rendering. Hu et al. [16] show the effectiveness of integrated
 157 positional encoding when applied to the multi-scale
 158 hybrid tri-plane representation. These methods necessitate
 159 explicit multi-scale supervision through downsampling im-
 160 ages [4, 44] and are primarily designed for filtering in 2D
 161 screen space — for novel-view synthesis applications. Our
 162 approach has broader applicability, as it can be employed on
 163 any neural field, and eliminates the need for a downsampled
 164 supervision signal.

165 **Frequency decomposition by network design.** Neural
 166 fields can decompose the frequency of the signal by
 167 specific neural network design. Multiplicative filter net-

168 works (MFNs) [11] introduce an architecture that outputs
 169 a linear combination of sinusoidal bases, enabling explicit
 170 frequency control. BACON [23] utilizes a multi-layer MFN
 171 for signal reconstruction, providing frequency upper-bound
 172 estimates at each layer. Polynomial Neural Fields [46] re-
 173 fines this approach with a more complex architecture, fa-
 174 cilitating precise frequency sub-band decomposition, and
 175 Shekarforoush et al. [36] enables progressive training of
 176 finer level-of-detail via the introduction of skip connec-
 177 tions. These methods have an inherent limitation: due to
 178 their specialized network design, they are not applicable
 179 to mainstream *hybrid* neural field representations. Some
 180 other methods perform joint spatial and frequency de-
 181 composition by applying Fourier feature encodings on local grid
 182 features [10, 43]. Despite their hybrid representation, simi-
 183 larly to MFNs, these methods rely on *specialized* architec-
 184 tures. Our approach can be applied atop *any* neural field
 185 and *independently* of its underlying architecture, therefore
 186 allowing to maintain the advantages of (mainstream) *hybrid*
 187 neural fields, while adding the ability to perform frequency
 188 decomposition.

3. Method

189 Neural fields $f(\mathbf{x}; \boldsymbol{\theta})$ represent signals with neural net-
 190 works, by randomly sampling field locations \mathbf{x} , and opti-
 191 mizing the parameters $\boldsymbol{\theta}$ to reproduce a given ground truth
 192 value $f(\mathbf{x})$:

$$\boldsymbol{\theta}^* = \arg \min_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{x}} \|f(\mathbf{x}; \boldsymbol{\theta}) - f(\mathbf{x})\|_2^2 \quad (1)$$

193 Thanks to the commutativity of the Fourier transform and
 194 linear operators [31], we can represent *any* signal, as the
 195 superposition of band-limited signals:

$$f(\mathbf{x}) = f_0(\mathbf{x}) + \dots + f_K(\mathbf{x}) \quad (2)$$

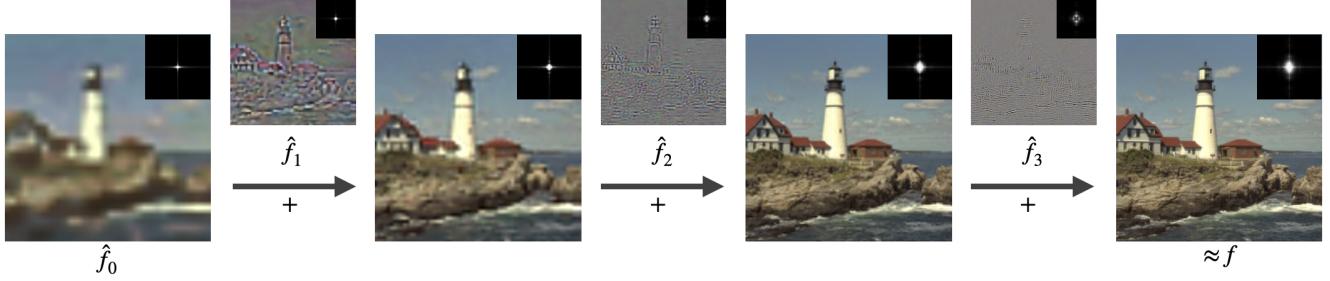


Figure 3. **Cascaded training** – We visualize our neural field decomposition where each level produces a signal with frequency band $[\omega_i, \omega_{i+1}]$. The decomposition is generated by a cascaded optimization akin to the generation of Laplacian pyramids in signal processing. Analogously, as the decomposition is quasi-orthogonal, we can generate the signal at any level though superposition. Image taken from the Kodak dataset [8].

where the discrete set of frequencies is denoted as $\{\omega_k\}$, and $\omega_0=0$, $\omega_{K+1}=\infty$, and $f_k(\mathbf{x})=f_{[\omega_k, \omega_{k+1}]}(\mathbf{x})$ is the band-pass filtered $f(\mathbf{x})$, with frequency band $[\omega_k, \omega_{k+1}]$. In our work, we seek to train a neural field with the same characteristics, so that the field decomposes as:

$$f(\mathbf{x}; \{\theta_k\}) = f_0(\mathbf{x}; \theta_0) + \dots + f_K(\mathbf{x}; \theta_K). \quad (3)$$

Note that θ_k only represents the fraction of the signal with spectra in the range $[\omega_k, \omega_{k+1}]$. Further, we will not assume a filtered version of the ground truth signal $f_k(\mathbf{x})$ to be readily available, as this is an unreasonable request for inverse problems; e.g. we cannot ask for a 3D representation to be filtered, as we only have 2D information about the 3D scene at hand. In theory, this could be achieved if a filtered version of the signal could be derived by convolving with a low-pass filter κ_{ω_k} with cut-off frequency ω_k :

$$f_{[0, \omega_k]}(\mathbf{x}) = \kappa_{\omega_k}(\mathbf{x}) \circledast f(\mathbf{x}) \quad (4)$$

from which we can obtain the desired *band-pass* signal:³

$$f_k(\mathbf{x}) = \kappa_{\omega_{k+1}}(\mathbf{x}) \circledast f(\mathbf{x}) - \kappa_{\omega_k}(\mathbf{x}) \circledast f(\mathbf{x}) \quad (5)$$

In what follows, we will first show that (4) can be achieved via *optimization* rather than explicit convolution (Sec. 3.1), and how (5) leads to a convenient cascaded training scheme (Sec. 3.2).

3.1. Filtering via optimization

Denote with $f[t]$ a discrete signal with uniform sampling of period T . From classical signal processing [31], such a discrete signal can be *reconstructed* as a continuous signal by applying a discrete-continuous convolution with a reconstruction kernel $\hat{\kappa}(\mathbf{x})$:

$$\hat{f}(\mathbf{x}) = \hat{\kappa}(\mathbf{x}) \circledast \sum_t f[t] \cdot \delta(\mathbf{x} - \mathbf{x}_t) \quad (6)$$

³Note the above resembles the filters employed to construct Laplacian pyramids via repeated applications of the so-called ‘‘mexican hat’’ kernel $\kappa_{\omega_{k+1}}(\mathbf{x}) - \kappa_{\omega_k}(\mathbf{x})$ [6].

where $\{\mathbf{x}_t\}$ are the sampling locations. Via the Convolution Theorem, it can be shown that $\hat{f}(\mathbf{x})$ has frequency that is upper-bounded by $\omega \leq \frac{2\pi}{T}$, and use notation $\hat{f}_{\omega}(\mathbf{x})$ to indicate this upper bound.

We now replace the discrete signal $f[t]$ in (6) with a neural field sampled on a uniform lattice $\{\mathbf{x}_t\}$:

$$\hat{f}_{\omega}(\mathbf{x}; \theta) = \kappa_{\omega}(\mathbf{x}) \circledast \sum_t f(\mathbf{x}_t; \theta) \cdot \delta(\mathbf{x} - \mathbf{x}_t) \quad (7)$$

We note that when $\kappa_{\omega}(\mathbf{x})$ is a triangle kernel $\Lambda_{\omega}(\mathbf{x})$ with local support equal to the period T , then the convolution above amounts to typical linear interpolation that is used to propagate features within voxels [39] that is found in many NeRF implementations [30]:

$$\hat{f}_{\omega}(\mathbf{x}; \theta) = \Lambda_{\omega}(\mathbf{x}) \circledast \sum_t f(\mathbf{x}_t; \theta) \cdot \delta(\mathbf{x} - \mathbf{x}_t) \quad (8)$$

$$\equiv \text{Interp}(\mathbf{x}; \{f(\mathbf{x}_t; \theta)\}) \quad (9)$$

Let us now incorporate (9) into the field training loop (1):

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{\mathbf{x}} \|\hat{f}_{\omega}(\mathbf{x}; \theta) - f(\mathbf{x})\|_2^2 \quad (10)$$

Under certain conditions (Appendix A), we can show that:

$$\hat{f}_{\omega}(\mathbf{x}; \theta^*) \approx \kappa_{\omega}(\mathbf{x}) \circledast f(\mathbf{x}) \quad (11)$$

which shows that an optimized neural field $\hat{f}_{\omega}(\mathbf{x}; \theta^*)$ can approximate a low-pass filtered signal $f(\mathbf{x})$ with frequency band $[0, \omega]$.

3.2. Cascaded training

Recall our objective is to obtain a decomposition of the field like (3). With (10), we can optimize our first neural field $\hat{f}_0(\mathbf{x}; \theta_0)$ with frequency band $[0, \omega_1]$:

$$\theta_0^* = \arg \min_{\theta_0} \mathbb{E}_{\mathbf{x}} \|\hat{f}_0(\mathbf{x}; \theta_0) - f(\mathbf{x})\|_2^2 \quad (12)$$

254 We proceed with our next neural field $\hat{f}_1(\mathbf{x}; \boldsymbol{\theta}_1)$ with frequency band $[\omega_1, \omega_2]$:
 255

256
$$\boldsymbol{\theta}_1^* = \arg \min_{\boldsymbol{\theta}_1} \mathbb{E}_{\mathbf{x}} \|\hat{f}_1(\mathbf{x}; \boldsymbol{\theta}_1) - (f(\mathbf{x}) - \hat{f}_0(\mathbf{x}; \boldsymbol{\theta}_0^*))\|_2^2 \quad (13)$$

257 We note that $(f(\mathbf{x}) - \hat{f}_0(\mathbf{x}; \boldsymbol{\theta}_0^*))$ is a high-pass filtered signal with frequency band $[\omega_1, +\infty]$, and as we have shown in (11), $\hat{f}_1(\mathbf{x}; \boldsymbol{\theta}_1)$ will be a low-pass signal with frequency band $[0, \omega_2]$. Therefore, the optimization above will capture a representation whose frequency band $[\omega_1, \omega_2]$ is the intersection of these two ranges. The optimization scheme for subsequent levels can then be derived by induction:
 258
 259
 260
 261
 262
 263

264
$$\arg \min_{\boldsymbol{\theta}_k} \mathbb{E}_{\mathbf{x}} \|\hat{f}_k(\mathbf{x}; \boldsymbol{\theta}_k) - (f(\mathbf{x}) - \sum_{l=0}^{k-1} \hat{f}_l(\mathbf{x}; \boldsymbol{\theta}_l^*))\|_2^2 \quad (14)$$

265 Once all levels are trained, the signal can then be recomposed via superposition of frequency bands:
 266

267
$$\hat{f}(\mathbf{x}; \boldsymbol{\theta}^*) = \sum_k \hat{f}_k(\mathbf{x}; \boldsymbol{\theta}_k^*) \approx f(\mathbf{x}) \quad (15)$$

268 This straightforward, yet powerful, idea can be applied to a
 269 variety of signal reconstruction tasks.

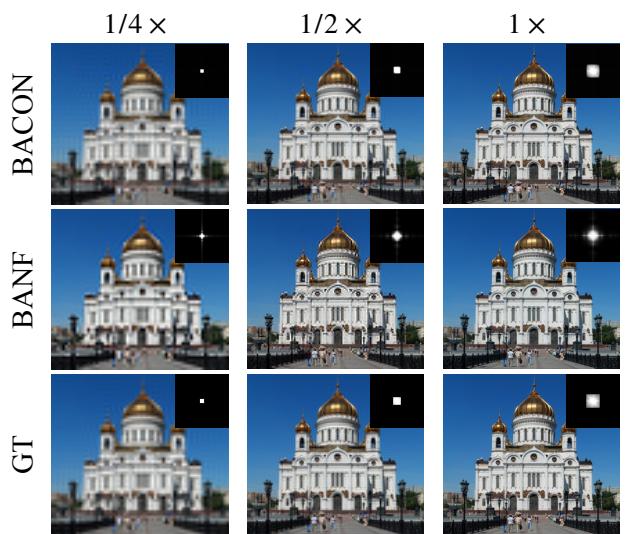
270 **Implementation.** To implement cascaded training, unless
 271 specified otherwise, we initialize all neural field parameters of all levels with random zero-mean weights of near-zero variance and zero bias. Further, the lowest resolution model $\hat{f}_0(\mathbf{x}; \boldsymbol{\theta}_0)$ is trained gradually; following [15, 47], we initially train the model in a warm-up phase at $1/4\times$ and then $1/2\times$ of its intended final resolution. We empirically found that having this warm-up phase encourages smoothness and stabilizes optimization. We ablate our choice in Sec. 4.4.
 272
 273
 274
 275
 276
 277
 278
 279

280 4. Experiments

281 We demonstrate the efficacy of our method in frequency
 282 decomposition with three distinct applications: 2D image
 283 fitting (Sec. 4.1), 3D shape fitting with signed distance
 284 field supervision (Sec. 4.2), and 3D shape recovery from
 285 inverse rendering (Sec. 4.3). We then ablate our design
 286 choices (Sec. 4.4).

287 4.1. Image filtering (2D)

288 We evaluate our method qualitatively by fitting a neural field
 289 to 2D images and reconstructing each image at four different
 290 frequency levels. We use an iNGP [30] backbone for
 291 images, with an MLP with 3 hidden layers, each with 32
 292 neurons to convert deep features into color. The ground
 293 truth signal is a 2D image of resolution 256^2 , for which
 294 we learn to represent it at resolution $r \in \{64^2, 128^2, 256^2\}$.
 295 We train the network with RMSProp [41] for 1K iterations
 296 with a learning rate of 2×10^{-3} . We use a batch size of 2^{16} .



297 **Figure 4. Image fitting and filtering –** We visualize both
 298 multi-scale reconstruction and Fourier spectra of BANF compared
 299 against BACON and the ground truth image.
 300
 301

302 randomly sampled points. We note that when sampling, we
 303 sample from a continuous 2D space with bilinear interpolation,
 304 as we are interested in the behavior of each method related
 305 to the underlying *continuous* signal. We use the mean
 306 squared error as the training objective.
 307
 308

309 We evaluate our method qualitatively on images from the
 310 DIV2K dataset [3], resampled to a resolution of 256^2 following
 311 [23]. We select BACON [23] as a representative
 312 baseline in 2D image decomposition using the Multiplicative
 313 Filter Network (MFN) [11] architecture. As a reference,
 314 we provide images decomposed directly with classical
 315 signal processing, which we refer to as ground truth.
 316
 317

318 As shown in Figure 4, our method successfully approximates
 319 low-pass filtering, being similar to the one provided
 320 by BACON as well as the ground truth. We note that as we
 321 use a linear interpolation, we do not obtain exact low-pass
 322 filtering, but rather an approximation (The Fourier trans-
 323 form of a triangle kernel is $sinc^2$, which is only an approx-
 324 imate low-pass filter). Yet, as shown, it approximates well,
 325 and leads to a meaningful frequency decomposition of the
 326 signal. Also note that due to the duality of supports (com-
 327 pactly supported functions in the frequency domain, like an
 328 exact lowpass filter, are globally supported in the spatial do-
 329 main), there are visible Gibbs ringings in both BACON and
 330 the ground truth at the lower scales. While for “perfect”
 331 low-pass filtering Gibbs artifacts are expected, their exis-
 332 tence is typically harmful to many downstream tasks. Our
 333 results, on the other hand, while faithfully representing the
 334 details at lower frequency, do not suffer from these artifacts.
 335
 336

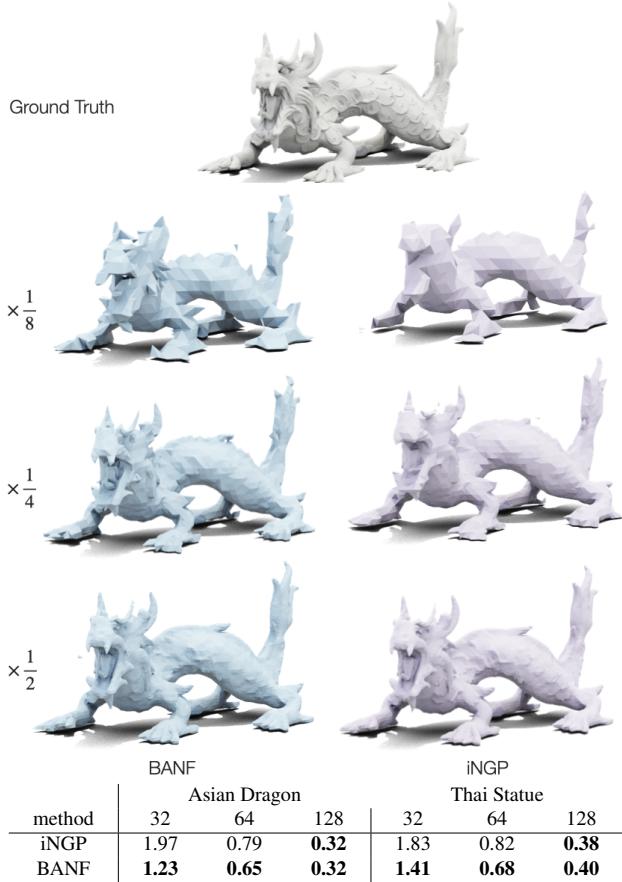


Figure 5. **Fitting SDFs** – We show qualitative results and report the Chamfer-L2 distance ($\times 10^2$) ↓. By filtering the signal during training, our method implements anti-aliasing, so that the extracted signal at coarser scales approximates better.

4.2. Signed Distance Fields (3D)

We now evaluate our method for learning Signed Distance Fields (SDF) at various levels of detail through *direct* 3D supervision. We assume that exact meshes are unavailable, and only implicit signed distances are, which is commonly the case for real-world captures. We demonstrate that our method can directly decompose frequencies.

Dataset. We evaluate with two commonly used meshes “Asian Dragon” and “Thai Statue”. Both have intricate details that are of high frequency, making the SDF fitting task and, more specifically frequency decomposition, challenging. We first normalize the meshes to fit into a unit sphere. We then sample 500k points to get their respective SDF values, with 40% of the points on the surface, 40% near the surface, and the rest uniformly within the bounding box. We use the same points to train all models, and optimize via mean square error for the predicted SDF values.

Baseline. We compare our method against iNGP. We use a multi-resolution hash grid of resolutions 16 to 2048, for 100k iterations with a batch size of 10k points until full

convergence. We then extract the meshes at the same resolutions as before using marching cubes [25]. We evaluate each method both qualitatively and quantitatively via Chamfer-L2 distance.

Implementation. We use the Adam optimizer [19] with a learning rate of 10^{-3} and the default training settings provided in iNGP. We learn the signal at four different resolutions $r \in \{32^3, 64^3, 128^3, 256^3\}$. As in Wang et al. [42], the coarsest level’s MLP is specifically initialized to produce a sphere for stable training. We train with a batch size of 100k points. We train each level for 10k iterations, except for the coarsest level, which we found to converge already at 5k iterations. After training, we use marching cubes [25] to extract meshes at each resolution.

Discussions. In Figure 5, we show quantitatively and qualitatively that our method better respects the Nyquist sampling theorem [31] due to the filtering that our method performs, resulting in higher quality approximations at coarse resolutions. As the resolution increases, the marching cubes sampling frequency exceeds the Nyquist rate, thus, as expected, both our method and vanilla iNGP achieve similar reconstruction quality.

4.3. Inverse rendering (3D from 2D)

We demonstrate the applicability of our method in a more complex scenario where only 2D images of a 3D signal are available. Here, we tackle the inverse problem of 3D surface reconstruction at *multiple* levels of detail. Note that as a uniformly sampled 3D signal is not available (only 2D observations are available) Fourier analysis cannot be used to create supervision.

Dataset and baseline. For the baseline we use vanilla NeUS [42] followed by marching cubes [25] to extract meshes. We evaluate our method on the (Synthetic) NeRF dataset [4] and the (real) MobileBrick dataset [21]. For MobileBrick, for both our method and NeUS, we use mask supervision and train without background modeling. We evaluate each method both qualitative, and quantitatively via the Chamfer-L2 distance.

Implementation. We extend the surface reconstruction baseline provided by NeUS [42] with our method. Following NeUS, we optimize the 3D surface field using volume rendering, guided by supervision from 2D images. Specifically, given a 3D position $\mathbf{x} \in \mathbb{R}^3$ and a viewing direction $\mathbf{v} \in \mathbb{S}^2$, we train neural fields for both signed distance $\hat{s}(\mathbf{x}; \theta)$ and color $\hat{c}(\mathbf{x}, \mathbf{v}; \theta)$. We apply cascaded training (Sec. 3.2) only to the SDF MLP and directly train separate color MLPs at each level – this is due to the challenge in color MLP optimization with signal decomposition (details in Sec. 4.4). We use an iNGP [30] backbone with an MLP that predicts SDF values and a feature vector. The feature

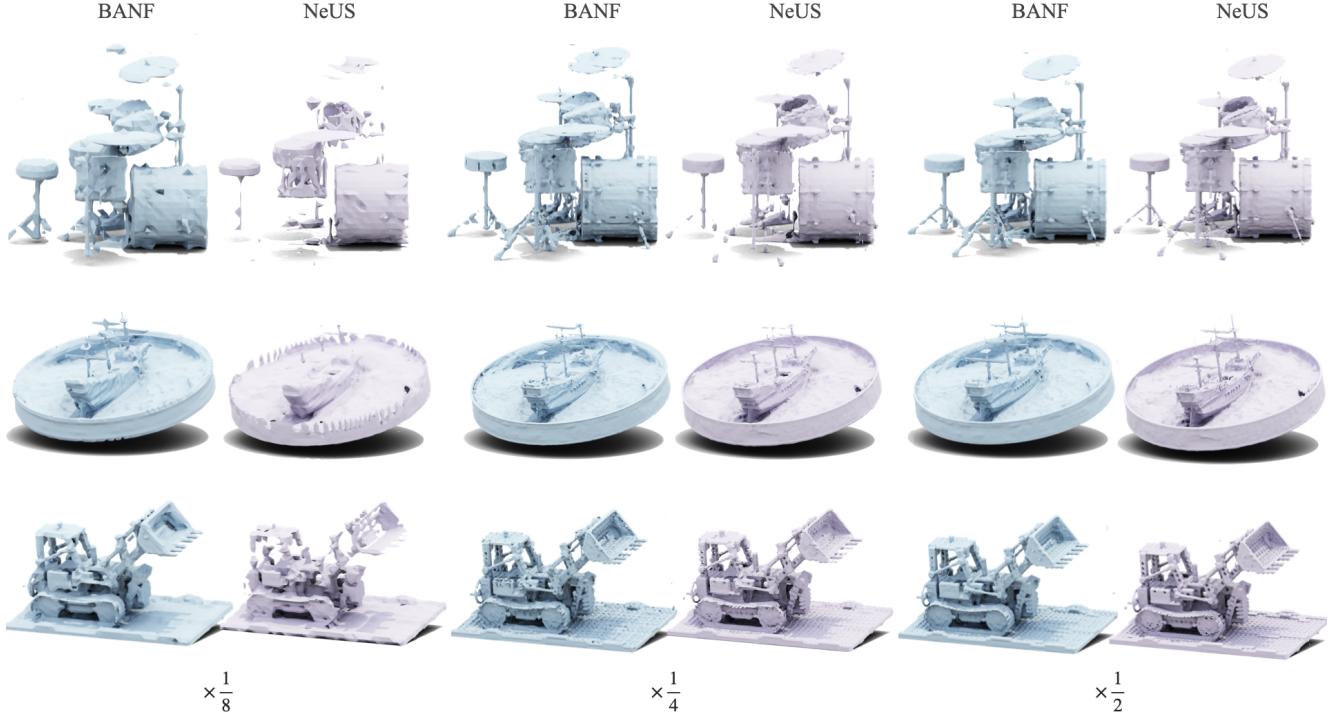


Figure 6. **Inverse rendering** – We integrate our technique into NeUS [42] for level-of-detail surface reconstruction. We show qualitative results on the Synthetic NeRF Dataset [4]. Note how, *especially* at coarser scales, the baseline lacks approximation of fine-grained details (caused by aliasing, as the baseline lacks filtering). Quantitative results are provided in Table 1.

Scale	Method	Chair	Drums	Ficus	Hotdog	Lego	Material	Mic	Ship	Average
1/8x	NeUS	6.05	21.8	40.3	8.82	10.2	24.5	10.5	16.4	17.3
	BANF	7.79	14.4	11.1	15.0	9.36	11.6	10.2	11.6	11.4
1/4x	NeUS	4.09	11.41	18.9	7.54	5.89	17.0	5.60	13.2	10.5
	BANF	6.29	8.66	7.16	10.2	6.92	9.44	7.53	9.32	8.19
1/2x	NeUS	3.94	9.03	8.06	7.17	4.74	14.7	4.32	12.1	8.01
	BANF	5.92	8.43	5.78	10.2	6.89	8.90	6.66	9.15	7.74
1x	NeUS	4.08	8.45	5.75	7.12	4.48	13.7	4.30	11.8	7.45
	BANF	5.82	8.31	5.36	10.0	6.69	8.68	5.79	8.95	7.45

Table 1. **Inverse rendering** – Quantitative results for NeUS [42] LOD reconstruction on the Synthetic NeRF Dataset [4]. See qualitative results in Figure 6. Metric is Chamfer Distance ($\times 10^{-2}$) ↓.

vector is then fed into the color MLP to output RGB values. We supervise training via classical photometric reconstruction error, along with Eikonal and Laplacian regularizers. When implementing our method, we observe a performance degradation when the color MLP relies solely on the output feature vector of the decomposed SDF MLP (see Sec. 4.4 for details). We thus additionally concatenate the features from the hashgrid to the inputs of the color MLP. We note that this minor architectural change does not lead to any significant changes in the performance of the baseline NeUS.

As done in Sec. 4.2, the coarsest SDF MLP head is initialized to produce a sphere. For initializing the color MLP heads, we first initialize the coarsest level randomly without any special treatment. For the subsequent levels, we boot-

strap their training by initializing them with the previous level’s (converged) weights. Our models are trained at four different resolutions $r \in \{64^3, 128^3, 256^3, 512^3\}$, trained in cascade for 20k iterations, with a batch size of 5k rays. To compute gradients of SDF that are used both as an input for the color MLP, and for computing the Eikonal and Laplacian regularizers, we use finite differences with the delta value set to $1/(4 \times r)$ for each level of our method, and $1/1024$ for the baseline model.

Discussions. As shown in Tables 1 and 2, our method significantly outperforms NeUS on average. In Figure 6, we visually see improvements on thin structures (see drum stool legs, ship’s mast, and loader’s bucket). We further ob-

Scale	Method	Ben	Boat	Bridge	Cabin	Camera	Castle	Colos	Jeep	Bus	Motor	Satel	Shuttle	Avg.
1/4x	NeUS	20.4	59.5	9.25	5.52	5.40	26.6	15.5	3.67	10.0	3.54	7.08	19.3	15.5
	BANF	10.0	6.40	4.61	4.64	2.78	17.4	12.9	2.96	5.48	2.73	5.27	13.5	7.39
1/2x	NeUS	8.04	6.15	5.01	3.95	3.94	11.4	9.90	2.60	5.66	2.15	4.17	6.78	5.81
	BANF	7.13	4.21	4.63	4.42	2.76	10.7	10.0	2.58	5.18	2.37	4.56	5.93	5.38

Table 2. **Inverse rendering** – We show qualitative results on the MobileBrick dataset [21]. Metric is Chamfer Distance ($\times 10^{-3}$) \downarrow . Again, larger gains are observable at coarser scales, when signals are sampled at a frequency that is below the one required by Nyquist.

serve improvement in anti-aliasing on the ship’s rim.

4.4. Ablations

We validate our design choices in terms of Chamfer Distance (CD \downarrow) evaluated on the NeRF Synthetic dataset [4] averaged over all objects, and at $1/8 \times$ scale.

Color MLP decomposition. When we train the color MLP output in a cascaded manner (see Sec. 3.2), performance drops CD=8.19 \rightarrow 9.16. We attribute this to the color MLP relying on the SDF MLP output. As color and geometry exhibit different frequencies in terms of their signal, (e.g., a highly complex shape with solid coloring), this reliance would be detrimental. For this ablation study we evaluate at $1/4 \times$ scale, as the difference is less marked at $1/8 \times$ scale.

Color MLP Input. Removing the hash features input to the color MLP (Section 4.3) results in a performance drop from CD=11.4 \rightarrow 13.6. Without the hash feature input, the color MLP relies only on the SDF features. At low scales, the SDF features will contain mostly low frequency content. We hypothesize that relying only on band-limited features reduces color reconstruction accuracy, resulting in inaccurate optimization of the the neural fields.

Resolution warmup. When we remove the resolution warmup from the coarsest level field, the performance drops CD=11.4 \rightarrow 11.9. This is because starting optimization with lower resolution encourages smooth structures early on, leading to an easier optimization landscape as also shown in [15, 47].

5. Conclusions

We introduce a way to train neural fields that enables frequency decomposition of the represented signals. Departing from heuristic- or architecture-based approaches, we realize this by a simple modification to the training process. Our approach is versatile, accommodating various neural field architectures (both fully neural and hybrid representations), and makes *no assumptions* about the training data, such as the availability of pre-filtered signals. We demonstrate its applicability across a number of neural fields workloads, and in particular by testing its effectiveness for anti-aliased level-of-detail reconstruction.

Limitations. In this paper, we primarily investigated fil-

tering of the *geometric* SDF signal in NeRF reconstruction workloads, as that is the signal that is typically extracted by marching cubes (at low-resolution, and therefore requiring filtering). However, by following a parameterization of color similar to the one in [12], one should be able achieve multi-band decomposition of both geometry and color, leading to a mipNeRF-like outcome for hybrid representations [4] for anti-aliased novel-view synthesis. Further, in this work we focused on uniformly sampled signals, and would be interesting how to extend the method to “contracted” representations that are commonly used in NeRF to deal with unbounded signals. Finally, while most of our results employ a linear interpolation kernel, it would be interesting to see whether, especially at lower scales, a smoother kernel with larger spatial support could provide a stronger filtering, and better anti-aliasing.

A. Appendix

Without loss of generalization to higher dimensions, let us consider 1D signals. Let us start by replacing \mathbb{E}_x in (10) with a sum operator, hence reducing our stochastic gradient descent optimization with least square optimization. Samples $\{\mathbf{x}_n\}$ are drawn uniformly with a period sufficient to satisfy the Nyquist theorem for the ground truth signal f :

$$\arg \min_{\theta} \sum_{\{\mathbf{x}_n\}} \|\text{Interp}(\mathbf{x}_n; \{f(\mathbf{x}_n; \theta)\}) - f(\mathbf{x}_n)\|_2^2 \quad (16)$$

Let us rewrite this expression in matrix form by denoting $\{f(\mathbf{x}_n)\}$ as $\mathbf{b} \in \mathbb{R}^{N \times D}$, $\{f(\mathbf{x}_n; \theta)\}$ as $\mathbf{X} \in \mathbb{R}^{T \times D}$, and by storing in the *skinny* matrix $\mathbf{A} \in \mathbb{R}^{N \times T}$ the linear interpolation coefficients corresponding to the positions $\{\mathbf{x}_n\}$:

$$\arg \min_{\mathbf{X}} \|\mathbf{AX} - \mathbf{b}\|_2^2 \quad (17)$$

Assuming the matrix \mathbf{A} is invertible, the optimization above provides the closest point projection of \mathbf{b} onto the range of \mathbf{A} [5, Sec. 5-3]. Recalling \mathbf{A} represents functions whose frequency is upper-bounded by ω , this implies the optimization projects our true signal onto the closest function with bounded spectra, which is equivalent to convolving the signal with a low-pass filter as in (11). Our approximations stem from the fact that *linear* interpolation is only an *approximation* of a low-pass filter (with a sinc² frequency spectra), and that our optimization is a stochastic gradient descent rather than a close-form solve.

503 **References**

- [1] Michal Adamkiewicz, Timothy Chen, Adam Caccavale, Rachel Gardner, Preston Culbertson, Jeannette Bohg, and Mac Schwager. Vision-only robot navigation in a neural radiance world. *IEEE Robotics and Automation Letters*, 7(2):4606–4613, 2022. 1
- [2] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984. 1
- [3] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. *CVPRW*, 2017. 5
- [4] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 3, 6, 7, 8
- [5] Stephen Boyd. Introduction to linear dynamical systems (course reader for ee263). <https://web.stanford.edu/class/archive/ee/ee263/ee263.1082/notes/ee263coursereader.pdf>, 2007. 8
- [6] P. Burt and E. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 1983. 4
- [7] Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. In *European Conference on Computer Vision (ECCV)*, 2022. 2
- [8] Eastman Kodak Company. Kodak lossless true color image, 1999. 4
- [9] Boyang Deng, JP Lewis, Timothy Jeruzalski, Gerard Pons-Moll, Geoffrey Hinton, Mohammad Norouzi, and Andrea Tagliasacchi. Neural articulated shape approximation. In *The European Conference on Computer Vision (ECCV)*. Springer, 2020. 2
- [10] Yishun Dou, Zhong Zheng, Qiaoqiao Jin, and Bingbing Ni. Multiplicative fourier level of detail. In *CVPR*, 2023. 2, 3
- [11] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. Multiplicative filter networks. In *ICLR*, 2021. 2, 3, 5
- [12] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 2, 8
- [13] Lily Goli, Daniel Rebain, Sara Sabour, Animesh Garg, and Andrea Tagliasacchi. nerf2nerf: Pairwise registration of neural radiance fields. In *International Conference on Robotics and Automation (ICRA)*. IEEE, 2023. 2
- [14] Ayaan Haque, Matthew Tancik, Alexei Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *CVPR*, 2023. 1
- [15] Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. Sape: Spatially-adaptive progressive encoding for neural optimization. *arXiv preprint arXiv:2104.09125*, 2021. 5, 8
- [16] Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *ICCV*, 2023. 3
- [17] Anil K Jain. *Fundamentals of digital image processing*. Prentice-Hall, Inc., 1989. 2
- [18] Animesh Karnewar, Tobias Ritschel, Oliver Wang, and Niloy Mitra. Relu fields: The little non-linearity that could. 2022. 2
- [19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015. 6
- [20] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caroline Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank Dellaert, and Thomas Funkhouser. Panoptic Neural Fields: A Semantic Object-Aware Neural Scene Representation. In *CVPR*, 2022. 1
- [21] Kejie Li, Jia-Wang Bian, Robert Castle, Philip H.S. Torr, and Victor Adrian Prisacariu. Mobilebrick: Building lego for 3d reconstruction on mobile devices. *CVPR*, 2023. 6, 8
- [22] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2
- [23] David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. *CVPR*, 2022. 2, 3, 5
- [24] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *NeurIPS*, 2020. 2, 3
- [25] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. Association for Computing Machinery, 1987. 1, 2, 6
- [26] David Luebke, Martin Reddy, Jonathan D. Cohen, Amitabh Varshney, Benjamin Watson, and Robert Huebner. *Level of Detail for 3D Graphics*. Morgan Kaufmann Publishers Inc., 2002. 2
- [27] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation, 2021. 1, 2, 3
- [28] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 2
- [29] 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*, 2020. 1, 2, 3
- [30] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *SIGGRAPH*, 2022. 2, 4, 5, 6
- [31] Sophocles J Orfanidis. *Introduction to Signal Processing*. Prentice-Hall, Inc., 1988. 2, 3, 4, 6
- [32] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 1, 2
- [33] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 1

- 616 [34] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *ICML*, 2019. 3
- 617 [35] Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G. Baraniuk, and Ashok Veeraraghavan. Miner: Multiscale implicit neural representations. In *ECCV*, 2022. 3
- 618 [36] Shayan Shekarforoush, David B Lindell, David J Fleet, and Marcus A Brubaker. Residual multiplicative filter networks for multiscale reconstruction. In *NeurIPS*, 2022. 3
- 619 [37] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 1, 2
- 620 [38] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 2
- 621 [39] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. 2021. 2, 4
- 622 [40] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 3
- 623 [41] Geoffrey Hinton Tijmen Tieleman. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 2012. 5
- 624 [42] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021. 1, 2, 6, 7
- 625 [43] Zhijie Wu, Yuhe Jin, and Kwang Moo Yi. Neural fourier filter bank. In *CVPR*, 2023. 2, 3
- 626 [44] Yuanbo Xiangli, Lining Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. Bungeenerf: Progressive neural radiance field for extreme multi-scale scene rendering. In *ECCV*, 2022. 3
- 627 [45] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022. 2
- 628 [46] Guandao Yang, Sagie Benaim, Varun Jampani, Kyle Genova, Jonathan T. Barron, Thomas Funkhouser, Bharath Hariharan, and Serge Belongie. Polynomial neural fields for subband decomposition and manipulation, 2022. 2, 3
- 629 [47] Jiawei Yang, Marco Pavone, and Yue Wang. Freenerf: Improving few-shot neural rendering with free frequency regularization. In *CVPR*, 2023. 5, 8
- 630 [48] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 1
- 631 [49] Zhuang Yiyu, Zhang Qi, Feng Ying, Zhu Hao, Yao Yao, Li Xiaoyu, Cao Yan-Pei, Shan Ying, and Cao Xun. Anti-aliased neural implicit surfaces with encoding level of detail. In *SIGGRAPH*, 2023. 3
- 632 [673] Zhuang Yiyu, Zhang Qi, Feng Ying, Zhu Hao, Yao Yao, Li Xiaoyu, Cao Yan-Pei, Shan Ying, and Cao Xun. Anti-aliased neural implicit surfaces with encoding level of detail. In *SIGGRAPH*, 2023. 3
- 633 [674]
- 634 [675]
- 635 [676]

BANF: Band-limited Neural Fields for Levels of Detail Reconstruction

Supplementary Material

001 1. Radiance field decomposition

002 We extend the application of our frequency decomposition
 003 technique to the color field in neural radiance fields. Here,
 004 we adopt the spherical harmonics (SH) representation pro-
 005 posed in Plenoxels [2] and [6] to perform color frequency
 006 decomposition. Specifically, in addition to implementing
 007 frequency-bounded grids for the density field, we apply fre-
 008 quency constraints to the Spherical Harmonics (SH) coef-
 009 ficients. At each level of detail, both the density and SH
 010 coefficients are queried at a specific resolution and then tri-
 011 linearly interpolated to determine the density and SH coef-
 012 ficients of the target point. Subsequently, the computed SH
 013 coefficients are transformed into RGB values. It's important
 014 to note that, due to the linearity of spherical harmonics, con-
 015 straining the frequency of SH coefficients directly imposes
 016 a constraint on the predicted color frequency. Further, both
 017 the density and color heads are trained using the previously
 018 proposed cascaded scheme. We compare the results of
 019 this method to a vanilla iNGP queried at target resolutions
 020 at test time. In Figure 1, we report superior performance to
 021 the baseline, highlighting the robustness of our method to
 022 aliasing effects.

023 **Implementation.** We train iNGP and our variation of it for
 024 50K iterations with a batch size of 4096 rays. We evaluate
 025 our method on the NeRF Synthetic Dataset [5] at resolu-
 026 tions $\{64^2, 128^2\}$, while the input images are at 800^2 res-
 027 olution. These evaluations are performed using grid reso-
 028 lutions $\{32^3, 64^3\}$ respectively, which were empirically de-
 029 termined to yield the best results. The SH coefficients of
 030 second order were used similar to [2] and [6]. Further, we
 031 note that we do not employ the extra skip connection from
 032 hash grid to the color MLP, as the color is being filtered.

033 2. Image filtering (cont'd)

034 We show additional quantitative and qualitative results on
 035 the DIV2K [1] dataset in Figure 2 and Table 1. Our method
 036 is trained on original images downsampled to 256^2 res-
 037 olution and compared to BACON [4] and PNF [3], trained
 038 in a similar fashion. The networks are trained on 5K sam-
 039 ples of the images. Evaluation is done at resolution 512^2 ,
 040 comparing to original images downsampled to this resolution
 041 to compute PSNR. Qualitative results are shown at resolu-
 042 tions $\{64^2, 128^2, 256^2\}$. As our method is compatible with
 043 any neural field, we demonstrate that, being based on an
 044 efficient backbone, it can reconstruct high-quality multi-
 045 resolution images while maintaining the same number of
 046 parameters as other techniques.

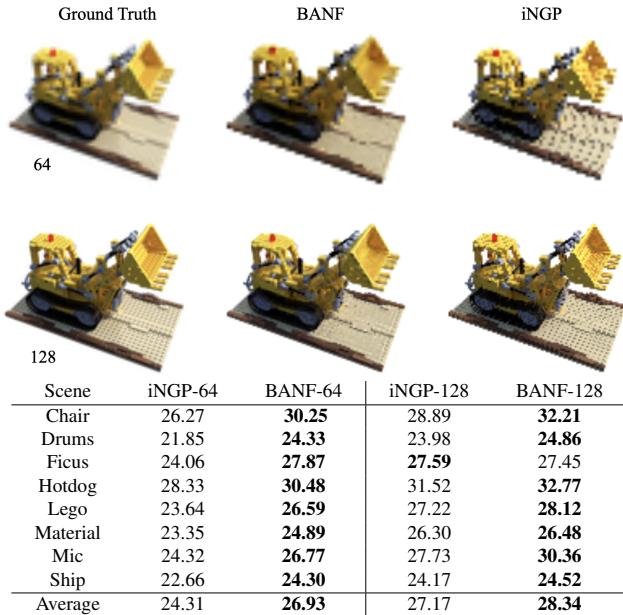


Figure 1. **Color decomposition** – We show an application of our method in frequency decomposition of the color field in NeRFs. Please refer to the [supplementary video](#) for more results.

Method	BACON	PNF	BANF
PSNR \uparrow	29.266	29.470	30.455
# Parameters \downarrow	0.268M	0.276M	0.244M

Table 1. **Image fitting** – quantitative results on DIV2K [1].

Resolution	64	128	256
Bilinear	25.650	25.364	30.455
Bicubic	26.559	26.380	30.697
Lanczos	26.050	26.415	30.210

Table 2. PSNR reported on interpolation with higher order kernels.

047 3. Higher order kernels

In our main results we used *linear* interpolation in our filter-
 048 ing algorithm. However, it was noted that this interpolation
 049 exhibits leakage, attributed to the sinc^2 Fourier transform
 050 of the filter. We further investigate how higher-order inter-
 051 polation kernels such as Lanczos and Bicubic can further
 052 improve the filtering quality. In Figure 3, We evaluate
 053 results on the DIV2K [1] dataset, with images downsampled
 054 to 256^2 . We then evaluate both quantitatively and qualita-
 055 tively at resolutions $\{64^2, 128^2, 256^2\}$. We show in Table 2
 056 that higher-order interpolations can help give an extra boost
 057 to the performance of our method, while producing (perceptu-
 058 ally) better filtered images Fig. 3.

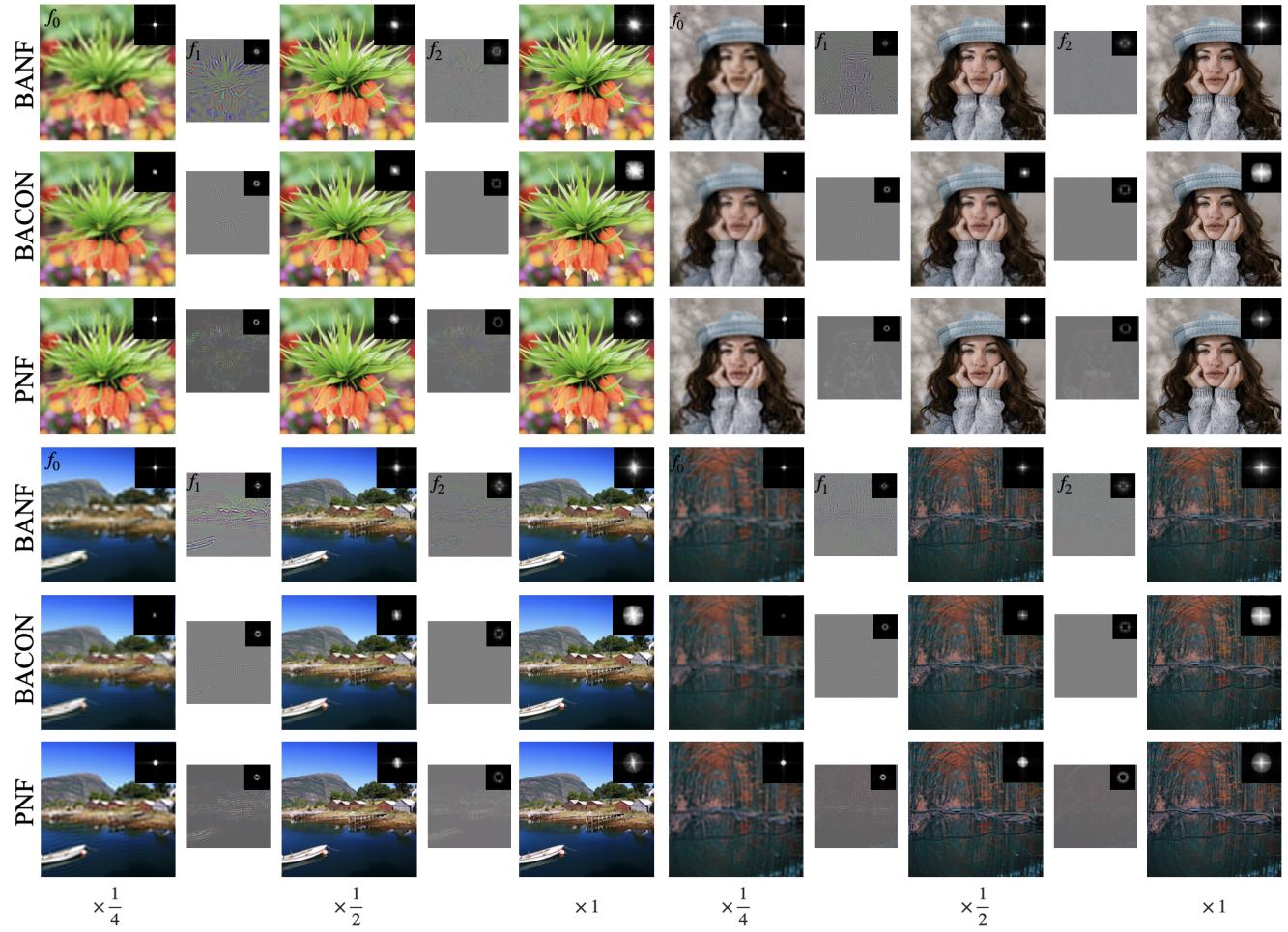


Figure 2. **Image Filtering** – Comparison of 2D filtering results on DIV2K [1] to BACON [4] and PNF [3]. Quantitative results are provided in Table 1.

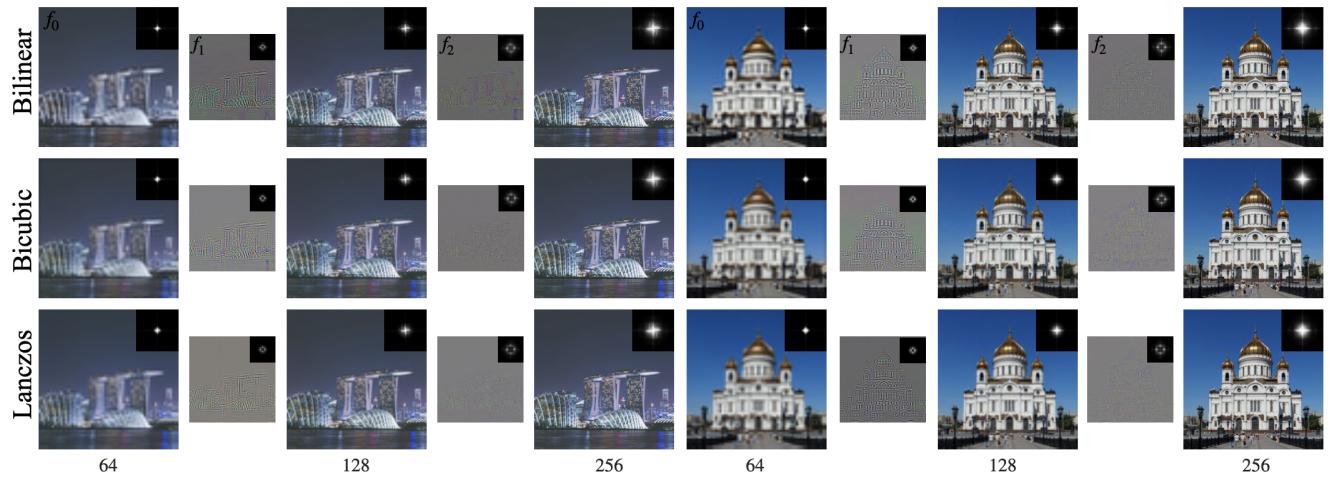


Figure 3. **Higher Order Kernels** – We evaluate the effect of higher order interpolation kernels on DIV2K [1].

060 **4. Evaluation of geometry reconstruction on**
061 **NeRF Synthetic Dataset**

062 In the NeRF Synthetic Dataset [5], not all aspects of the ge-
063 ometries are observable from the training/validation cam-
064 eras. This includes:

065 1) Internal structures that evade capture by any means,
066 such as the stem of the plant inside the pot in the "Ficus"
067 scene.

068 2) Some parts of the geometry that remain invisible from
069 all cameras (both during training and testing). For example,
070 the bottom of the chair in the "Chair" scene.

071 This inherent limitation results in a skewed assessment
072 of metrics designed to measure the quality of reconstruc-
073 tion. To address this issue when computing the Chamfer
074 Distance, we modify the process for densely sampled points
075 on the surface of a mesh. Specifically, we filter the point
076 cloud so that only points visible from at least one camera
077 are retained. This filtering procedure is applied to both the
078 ground truth and predicted meshes, ensuring a more accu-
079 rate evaluation.

080 **References**

- 081 [1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge
082 on single image super-resolution: Dataset and study. *CVPRW*,
083 2017. [1](#), [2](#)
- 084 [2] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong
085 Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels:
086 Radiance fields without neural networks. In *CVPR*, 2022. [1](#)
- 087 [3] Abhijit Kundu, Kyle Genova, Xiaoqi Yin, Alireza Fathi, Caro-
088 line Pantofaru, Leonidas Guibas, Andrea Tagliasacchi, Frank
089 Dellaert, and Thomas Funkhouser. Panoptic Neural Fields:
090 A Semantic Object-Aware Neural Scene Representation. In
091 *CVPR*, 2022. [1](#), [2](#)
- 092 [4] David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gor-
093 don Wetzstein. Bacon: Band-limited coordinate networks for
094 multiscale scene representation. *CVPR*, 2022. [1](#), [2](#)
- 095 [5] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik,
096 Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf:
097 Representing scenes as neural radiance fields for view synthe-
098 sis. In *ECCV*, 2020. [1](#), [3](#)
- 099 [6] Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and
100 Angjoo Kanazawa. Plenoctrees for real-time rendering of neu-
101 ral radiance fields, 2021. [1](#)