Spectral Prefiltering of Neural Fields

MUSTAFA B. YALDIZ, University of California San Diego, USA
ISHIT MEHTA, University of California San Diego, USA
NITHIN RAGHAVAN, University of California San Diego, USA
ANDREAS MEULEMAN, Inria, Université Côte d'Azur, France
TZU-MAO LI, University of California San Diego, USA
RAVI RAMAMOORTHI, University of California San Diego, USA

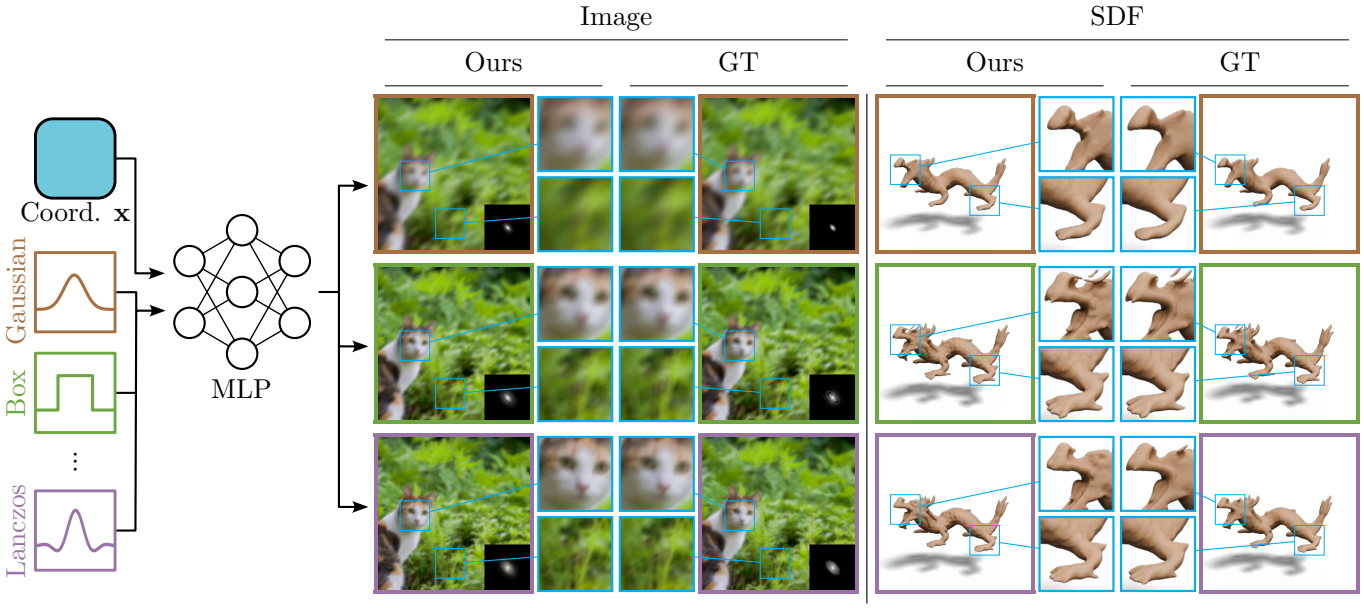

Fig. 1. We present a training method for neural fields that enables linear prefiltering with multiple reconstruction filters. At training time, the neural field sees parameters of a single symmetric filter. At test time, we support prefiltering a variety of unseen filters (*e.g.,* Box or Lanczos). Here, we show neural fields trained on an image (with bottom-right insets of frequency spectrum) and signed distance function using Gaussian filters, with generalization on Box and Lanczos filters. Images from Adobe FiveK; © original photographers/Adobe. Mesh models from the Stanford 3D Scanning Repository; © Stanford Computer Graphics Laboratory. (Project page: https://myaldiz.info/assets/spnf/)

Neural fields excel at representing continuous visual signals but typically operate at a single, fixed resolution. We present a simple yet powerful method to optimize neural fields that can be prefiltered in a single forward pass. Key innovations and features include: (1) We perform convolutional filtering in the input domain by analytically scaling Fourier feature embeddings with the filter's frequency response. (2) This closed-form modulation generalizes beyond Gaussian filtering and supports other parametric filters (Box and Lanczos) that are unseen at training time. (3) We train the neural field using single-sample Monte Carlo estimates of the filtered signal. Our method is fast during both training and inference, and imposes no additional constraints on the network architecture. We show quantitative and qualitative improvements over existing methods for neural-field filtering.

Authors' Contact Information: Mustafa B. Yaldiz, University of California San Diego, La Jolla, CA, USA, myaldiz@ucsd.edu; Ishit Mehta, University of California San Diego, La Jolla, CA, USA, ibmehta@ucsd.edu; Nithin Raghavan, University of California San Diego, La Jolla, CA, USA, n2raghavan@ucsd.edu; Andreas Meuleman, Inria, Université Côte d'Azur, Sophia Antipolis, France, andreas.meuleman@gmail.com; Tzu-Mao Li, University of California San Diego, La Jolla, CA, USA, tzli@ucsd.edu; Ravi Ramamoorthi, University of California San Diego, La Jolla, CA, USA, ravir@ucsd.edu.

1 Introduction

Neural fields are now widely adopted in visual computing [Xie et al. 2022]. They are used as continuous functions that map coordinates from an input domain (*e.g.*, pixel locations) to the corresponding signal values (*e.g.*, radiance). Generally, they provide point-wise estimates of the signal. As such, naively upsampling or downsampling the neural field produces sampling artifacts, preventing applications such as mipmapping. Previous works [Fathony et al. 2020; Lindell et al. 2022; Mujkanovic et al. 2024] aim to address the demand for *resolution-aware* neural fields. These methods impose significant restrictions on their network architectures, and often are restricted to a specific type of filter like Gaussian. In contrast, many graphics applications, require alternative filters that trade between sharpness and ringing. We propose a method for fitting neural fields that enables accurate filtering with a variety of low-pass symmetric reconstruction kernels, while imposing few constraints on the neural network architecture. Given the parameters of a filter and a spatial

coordinate, our network predicts filtered signal values in a single evaluation (Fig. 1).

Our key idea is to integrate the Fourier feature encoding [Tancik et al. 2020] and derive an analytical formula to modify the coefficients. We show that by supervising the network with one type of low-pass filter (*e.g.*, Gaussian), it naturally generalizes to different types of low-pass filters (*e.g.*, Box or Lanczos). We estimate the filtered signal and train the network with a Monte Carlo estimator of the convolution.

We demonstrate our method for prefiltering 2D images and 3D signed distance functions. Since we do not impose any restriction on the network architecture apart from the Fourier feature encoding, we achieve significantly higher quality results than prior work [Mujkanovic et al. 2024]. Furthermore, our Monte Carlo estimator only requires a single sample from the convolution filter; hence it induces little performance cost during training even when the signal is expensive to evaluate.

In summary, our contributions are:

(1) An analytical prefiltering approach for neural fields using Fourier feature encoding,
(2) A training regime for prefiltered neural fields that generalizes to a variety of linear and symmetric convolutional filters.

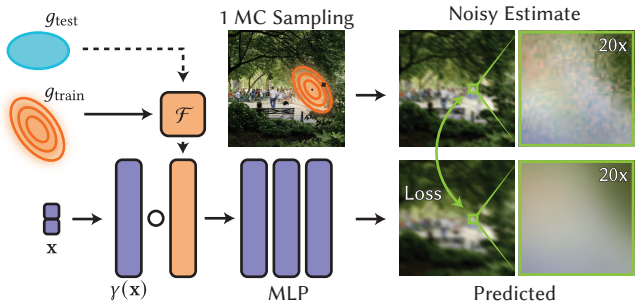

Fig. 2. *Method Overview.* Our method rests on two key ideas: (i) using the analytic Fourier transform ($\mathcal{F}$) of a symmetric linear filtering kernel to modulate Fourier-feature embeddings, and (ii) using single-sample Monte Carlo (1 MC) estimates of the filtered signal for supervision. A neural field trained with one filter type g_{train}, generalizes to unseen filters g_{test}. Images from Adobe FiveK; © original photographers/Adobe.

2 Related Work

Multi-scale representations. Computer graphics and vision methods often rely on data structures (*e.g.*, mipmapping) that represent a signal at multiple scales [Williams 1983; Witkin 1987] to avoid expensive postfiltering. The multi-scale representations can be useful for texture filtering [Greene and Heckbert 1986; Heckbert 1989; Williams 1983], image processing [Adelson et al. 1984; Lowe 2004], level of detail [Hoppe 1996], and multi-resolution editing of geometry [Zorin et al. 1997]. Our work shares the same motivation as early texture filtering works. Post-filtering neural fields requires approximating the filtering integral through either a) cubature discretization; which is both memory and compute expensive or b) Monte Carlo sampling, which results in excessive noise (see Fig. 2) We focus on building a multi-scale representation of a single-pass coordinate neural network [Song et al. 2015] for a given filtering kernel.

Neural fields. Neural fields compactly represent continuous signals using multilayer perceptrons conditioned on spatial coordinates [Xie et al. 2022]. Their versatility makes them appealing for representing images [Belhe et al. 2023; Song et al. 2015], geometry [Park et al. 2019; Sivaram et al. 2024], light fields [Sitzmann et al. 2021], radiance fields [Mildenhall et al. 2021], and spatially varying reflectance [Bi et al. 2020; Rainer et al. 2019].

Input embeddings. Directly mapping coordinates to the output using a neural network can fail to represent high-frequency details due to the spectral bias of multilayer perceptrons [Rahaman et al. 2019]. To alleviate this, modern approaches often map spatial coordinates through diverse activations as embeddings (*e.g.*, Gaussian, sine) [Rahimi and Recht 2007; Stanley 2007; Vaswani et al. 2017]. In this work, we focus on the Fourier feature mapping introduced by Tancik et al. [2020], which encodes input coordinates using sines and cosines at multiple frequencies. We show that this representation enables the derivation of closed-form expressions for modifying feature weightings in accordance with a given filter kernel, leading to significant improvements in filtering accuracy.

Scale-aware neural rendering. Mip-NeRF [Barron et al. 2021] introduced conical ray integration to reduce aliasing via scale-aware positional encodings. Zip-NeRF [Barron et al. 2023], Tri-MipRF [Hu et al. 2023], and Rip-NeRF [Liu et al. 2024] extend this idea using hierarchical grids, mipmaps, and directional ripmaps. These methods optimize a photometric loss on down-sampled training images. They focus on anti-aliasing in radiance fields, but lack extensive analysis on whether internal signals such as volumetric density are correctly filtered, nor can they change the filtering at test-time.

Learned multi-scale representations. Song et al. [2015] represent a multilayer-perceptron-based mipmap using coordinates (x, y, l) where l is a continuous mipmapping level. Recently, several neural multi-scale representations are proposed [Fathony et al. 2020; Lindell et al. 2022; Saragadam et al. 2022; Shekarforoush et al. 2022; Yang et al. 2022]. However, they typically put significant constraints on the network architectures, and can only handle limited anisotropy. Beyond multi-scale architectures, Xu et al. [2022] directly learn operators that act on the neural fields. Nsampi et al. [2023] derive continuous convolutions for neural fields by convolving repeated derivatives of the kernel with repeated antiderivatives of the signal, which is exact for piecewise-polynomial kernels [Nsampi et al. 2023]. Lindell et al. [2022] directly modify the network architecture to handle box-filtering and Gaussian-filtering but their method is not continuous in filter space. Closest to our work, NGSSF [Mujkanovic et al. 2024] learns an anisotropic Gaussian scale-space via a scale-conditioned MLP, but remains restricted to Gaussian filters and relies on Lipschitz regularization plus a calibration stage for continuous smoothing. In contrast, our approach uses frequency-aware Fourier features and a single-sample Monte Carlo estimator

to achieve filter-agnostic and anisotropic prefiltering without further architectural constraints, resulting in both greater flexibility and superior reconstruction quality.

3 Background

Let $\mathbf{x} \in \mathbb{R}^{d_i}$ be a d_i-dimensional *input coordinate*, and consider a continuous signal $f : \mathbb{R}^{d_i} \to \mathbb{R}^{d_o}$, where d_i and d_o are typically small (*e.g.*, $d_i = 2$, $d_o = 3$ for RGB images, and $d_i = 3$, $d_o = 1$ for a signed-distance field). A *neural field* approximates this signal:

$$F_\theta(\mathbf{x}) \approx f(\mathbf{x}), \tag{1}$$

where θ are the learned parameters. To incorporate a filter K parameterized by a parameter Σ (*e.g.*, a Gaussian kernel) the *filtered signal* is defined as the following convolution:

$$f_{K,\Sigma}(\mathbf{x}) \doteq (K_\Sigma * f)(\mathbf{x}) = \int K_\Sigma(\mathbf{x} - \mathbf{x}')\, f(\mathbf{x}')\, \mathrm{d}\mathbf{x}'. \tag{2}$$

Throughout this paper, we assume the spatial kernel K_Σ is *symmetric*, *i.e.*, $K_\Sigma(\mathbf{x}) = K_\Sigma(-\mathbf{x})$. After Fourier transform, its frequency response $\mathcal{F}\{K_\Sigma\}(\boldsymbol{\omega})$ is therefore real and even, yielding zero phase shift which simplifies our frequency representation (see Sec. 4.1). In the anisotropic filter case, symmetry still holds, but magnitude varies by direction (*e.g.*, elliptical Gaussian).

Conventionally, filtering a signal can be done through Monte Carlo estimation of the convolution integral [Hermosilla et al. 2018]. Naive Monte Carlo estimation for filtering neural fields suffers from two standard drawbacks, (i) high variance and (ii) reliance on full network evaluation for each Monte Carlo sample, making it expensive. To address these limitations, we seek to prefilter neural fields with a single compact model that approximates the filtered signal:

$$F_{K,\Sigma}(\mathbf{x}; \theta) \approx f_{K,\Sigma}(\mathbf{x}). \tag{3}$$

Our goal is to learn the neural field such that it produces filtered output in a single forward pass and is continuous in the parameter space Σ.

A common way to avoid per-query integration at inference is to relocate the convolution to the input features, *i.e.*, convolve the embedding once and then feed it to the network. Prior works [Barron et al. 2021; Liu et al. 2024; Wu et al. 2024] learn integrated input feature embeddings (*e.g.*, positional encoding [Vaswani et al. 2017], hashgrids, etc.) to filter neural fields:

$$\gamma_{K,\Sigma}(\mathbf{x}) = \int K_\Sigma(\mathbf{x} - \mathbf{x}')\, \gamma'(\mathbf{x}')\, \mathrm{d}\mathbf{x}'. \tag{4}$$

Here, γ' is a feature embedding at original resolution and $\gamma(\cdot, \Sigma)$ is the convolved embedding. These features are then passed through the network (MLP) to approximate the filtered signal in one pass:

$$F_{K,\Sigma}(\mathbf{x}; \theta) = \mathrm{MLP}_\theta\big(\gamma_{K,\Sigma}(\mathbf{x})\big). \tag{5}$$

We build on the feature-space view in Equations (4)–(5): we choose an embedding whose convolution with K_Σ is available in closed form, enabling a single-pass filtered output. We develop this idea further in Section 4.1, showing how it generalizes beyond a single kernel to other symmetric filter families with continuous control over Σ.

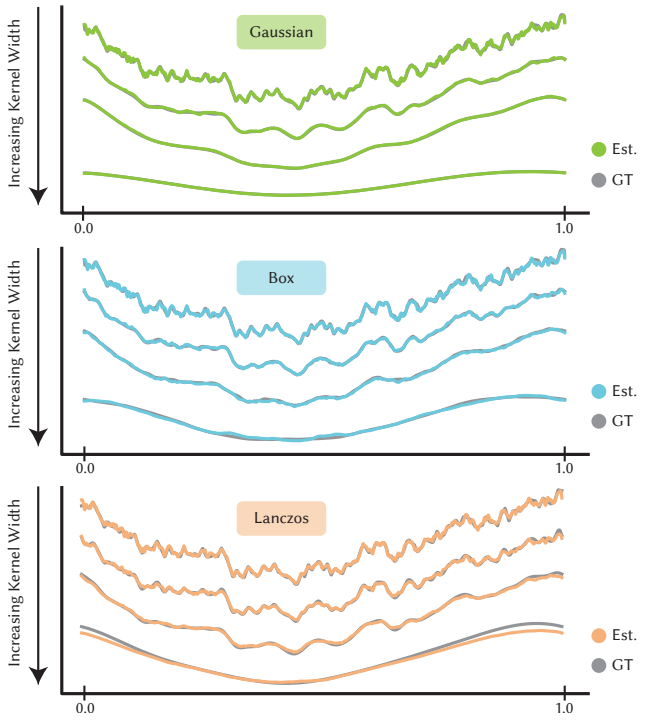

Fig. 3. *1D Example.* Given a discretely sampled 1D signal and a finite set of uniformly sampled Gaussian kernels, we train a neural field to encode the continuous scale space of the signal (*top*). By modulating the Fourier feature embeddings as per § 4.1, we prefilter the neural field with Box (*middle*) and Lanczos (*bottom*) filters with no additional supervision.

4 Method

Given a symmetric kernel K_Σ, our method: (i) prefilters Fourier features that are subsequently used by a neural network to predict the filtered signal; (ii) trains on a single-sample Monte Carlo estimate of $(K_\Sigma * f)$ (Sec. 4.2). Implementation details are provided in Section 4.3. At test time, the same model supports arbitrary covariance Σ (isotropic or anisotropic) and previously unseen filters, such as Box and Lanczos, in a single forward pass (see Fig. 2).

4.1 Prefiltering Fourier Features

Our goal is to use a feature encoding $\gamma'(\mathbf{x})$ with the following two properties:

(1) **Closed-form filtering.** Convolution of γ' with a given filter can be computed analytically, avoiding the runtime overhead and accuracy tradeoffs of Monte Carlo (MC) or quadrature-based methods.
(2) **Spectral-bias control.** The proposed encoding ensures robustness to MC noise during training and provides a direct mechanism to tune the MLP's sensitivity to different frequency bands.

Random Fourier feature embeddings meet these requirements:

$$\gamma'(\mathbf{x}) = \big[a_i \cos(2\pi\, \mathbf{b}_i^\top \mathbf{x}),\ a_i \sin(2\pi\, \mathbf{b}_i^\top \mathbf{x})\big]_{i=1}^{m}, \tag{6}$$

where the choice of frequencies $\{\mathbf{b}_i\}$ and amplitudes $\{a_i\}$ controls the spectral bias [Tancik et al. 2020]. In this work, we extend this idea by deriving an exact, filter-dependent modulation of $\{a_i\}$, giving precise control over the network-output's frequency spectrum. Consistent with Equation (4) in Section 3, we *instantiate* $\gamma(\mathbf{x}, \Sigma)$ by exactly convolving this embedding with K_Σ.

In Equation (6), each entry g_i of $\gamma'(\mathbf{x})$ is a sinusoid at frequency $\mathbf{b}_i$, whose spectrum is a pair of impulses at $\pm\mathbf{b}_i$. By the convolution theorem,

$$\mathcal{F}\{K_\Sigma * g_i\} \;=\; \mathcal{F}\{K_\Sigma\} \cdot \mathcal{F}\{g_i\}. \tag{7}$$

For a *symmetric* kernel, $\mathcal{F}\{K_\Sigma\}$ is real and even, so multiplying the two impulses at $\pm\mathbf{b}_i$ by $\mathcal{F}\{K_\Sigma\}(\mathbf{b}_i)$ only rescales the amplitude (no phase change). Inverting the transform gives:

$$(K_\Sigma * g_i)(\mathbf{x}) \;=\; \mathcal{F}\{K_\Sigma\}(\mathbf{b}_i)\, g_i(\mathbf{x}). \tag{8}$$

Thus feature-space convolution amounts to scaling each cosine/sine pair at $\mathbf{b}_i$ by the filter's magnitude at that frequency. We set the Fourier-feature amplitudes to the kernel's magnitude at each frequency,

$$a_i(K_\Sigma) \;=\; \mathcal{F}\{K_\Sigma\}(\mathbf{b}_i). \tag{9}$$

Using Equation (6) with $a_i = a_i(K_\Sigma)$ yields an embedding whose convolution with K_Σ is exact. When supervised with an estimate of the filtered signal, the network learns to condition on this prefiltered embedding so that a single forward pass outputs $(K_\Sigma * f)(x)$. Matching a_i to $\mathcal{F}\{K_\Sigma\}(\mathbf{b}_i)$ provides direct spectral-bias control.

Kernel-specific magnitudes. We now list the closed-form $a_i(K_\Sigma)$ used in this paper. Given that $\Sigma \in \mathbb{R}^{n\times n}$ is a symmetric positive–definite covariance matrix, we extend 1D kernels (Gaussian, Box, Lanczos) to nD by evaluating them at the Mahalanobis distance $\|\mathbf{x}\|_\Sigma = \sqrt{\mathbf{x}^\top \Sigma^{-1}\,\mathbf{x}}$, yielding anisotropic nD filters. We provide full derivations and constants in the supplemental material. For a Gaussian kernel:

$$a_i(K_\Sigma) = \exp\left(-2\pi^2 \mathbf{b}_i^\top \Sigma \mathbf{b}_i\right). \tag{10}$$

For an n-dimensional Box kernel, where $J_{n/2}$ is the Bessel function and Γ is the Gamma function:

$$a_i(K_\Sigma) = \frac{\Gamma(\frac{n}{2}+1)}{\pi^{n/2}} \frac{J_{n/2}\left(2\pi\sqrt{\mathbf{b}_i^\top \Sigma \mathbf{b}_i}\right)}{\left(\sqrt{\mathbf{b}_i^\top \Sigma \mathbf{b}_i}\right)^{n/2}}. \tag{11}$$

For a Lanczos kernel, let $p > 0$ denote the Lanczos *order* (we use p to avoid a clash with the Fourier–feature weights a_i), The Fourier transform of the nD Lanczos kernel is then:

$$a_i(K_\Sigma) = \frac{p}{z_n}\, \max\Big(\min\big(\frac{p+1}{2p} - \sqrt{\mathbf{b}_i^\top \Sigma \mathbf{b}_i},\; \min(1, \frac{1}{p})\big),\; 0\Big). \tag{12}$$

The constant z_n is a dimension n dependent normalization factor for the Lanczos kernel. Finally, following Equation (5) in Section 3, we pass the convolved features through the network to approximate the filtered signal.

As an illustration, consider the toy example in Fig. 3, where we encode the scale-space of a 1D signal in a neural field. We analytically integrate the Fourier features for a given filtering kernel, which acts as proxy for prefiltering the signal. We observe that using the proposed feature encoding modulation (Eq. (9)), enables prefiltering the neural field with additional filters without fine-tuning or re-training the MLP network.

4.2 Monte Carlo–Based Training

The previous section defines a deterministic, filter-conditioned encoding $\gamma_{K,\Sigma}(\mathbf{x})$. In our training scheme, we optimize only the MLP parameters θ to map $\gamma_{K,\Sigma}(\mathbf{x})$ to the filtered signal; the encoding itself is fixed given $(\mathbf{x}, K, \Sigma)$.

Most scale-aware neural-field methods avoid explicit or approximated convolution of the ground truth signal by baking the filter, either directly into the dataset (via precomputation) or by incorporating the filter into the network architecture. While these methods show good results, they nonetheless come with limitations as discussed in Section 2.

Objective and estimator. An unbiased estimate of the exact convolution can be given via Monte Carlo estimation; we draw N samples $\{\mathbf{x}'_i\}$ from a probability distribution $p(\cdot \mid \mathbf{x})$ centered at the point of convolution $\mathbf{x}$,

$$f_{K,\Sigma}(\mathbf{x}) = \mathop{\mathbb{E}}_{\mathbf{x}'\sim p(\cdot|x)}\left[\frac{K_\Sigma(\mathbf{x}-\mathbf{x}')}{p(\mathbf{x}'|\mathbf{x})} f(\mathbf{x}')\right] \tag{13}$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} \frac{K_\Sigma(\mathbf{x}-\mathbf{x}'_i)}{p(\mathbf{x}'|\mathbf{x})} f(\mathbf{x}'_i), \quad \mathbf{x}'_i \sim p(\cdot \mid \mathbf{x}). \tag{14}$$

We choose the density p to be proportional to $|K_\Sigma|$. However, a low-variance estimate given by dense multi-sampling is prohibitively slow. We therefore *train with a single Monte Carlo sample* (N=1) at each iteration.

Sampling procedure. During training, we uniformly sample a batch of coordinates $\mathbf{x}$ from the valid signal area, and we sample positive semidefinite covariances Σ by drawing principal variances uniformly in log-space with a random rotation.

(i) Given a filter (Gaussian, Box, Lanczos), we first normalize the filter $|K_\Sigma|$, treat it as a distribution and sample $\mathbf{x}'$. Distributions of Gaussian and Box filters yield normal and uniform sampling respectively, which is easy to sample. For Lanczos filter, we use rejection sampling, and cache those samples. During training and evaluation of Lanczos filter, we draw cached samples at random and stretch them according to the covariance.

(ii) Next, we sample the corresponding signal at $f(\mathbf{x}')$ as a MC estimate of the signal. Our single-sample Monte Carlo training is similar to the coordinate perturbation introduced by Ling et al. [2025] (See Eq. (15)). We also perturb the sampling coordinate, but unlike their method, we modulate the encoding according to the kernel K_Σ.

4.3 Implementation Details

We use standard settings for stable single-sample training.

Optimizer and schedule. We use Adam [Kingma and Ba 2014] for its robustness to noise. We found that an exponential learning-rate decay is critical: larger early steps drive progress despite target noise, while smaller later steps average out variance. In our experiments, adding exponential decay to $1e{-}3$ times of the original learning rate improved PSNR (*e.g.*, +2.19 dB on *Alien*) and reduced speckle. After

sweeping learning rates from 5×10^{-5} to 5×10^{-1}, stochastic gradient descent (with/without momentum) either converged to poor minima or diverged, so we adopt Adam with learning rate 5×10^{-4} with batch size $100k$ for images and $200k$ for SDF by default.

Loss. We minimize the minibatch MSE between the MLP on the prefiltered encoding and a single-sample MC target:

$$\hat{\mathcal{L}} = \frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x},K,\Sigma)\in\mathcal{B}} \left\| \mathrm{MLP}_\theta\big(\gamma_{K,\Sigma}(\mathbf{x})\big) - \frac{K_\Sigma(\mathbf{x}-\mathbf{x}')}{p(\mathbf{x}' \mid \mathbf{x}, K_\Sigma)}\, f(\mathbf{x}') \right\|_2^2, \quad \mathbf{x}' \sim p(\cdot \mid \mathbf{x}, K_\Sigma) \propto |K_\Sigma|. \tag{15}$$

Because the MC target is unbiased for $(K_\Sigma * f)(\mathbf{x})$, this MSE is an unbiased objective. For datasets with heavy-tailed, high-dynamic-range noise, one can instead explore objectives from Noise2Noise that down-weight extreme outliers [Mansour and Heckel 2023].

Architecture details. Unless otherwise noted, we follow the configuration from NGSSF [Mujkanovic et al. 2024]: a 3-layer multi-layer perceptron of width 1024 and (m=512) 1024 channel Fourier features. We use a basis scale of 2000 for images and 40 for SDFs. The Fourier basis scale should match the signal's frequency distribution: if set too small, the model is biased toward low frequencies (underfitting high-frequency detail); if set too large, it over-emphasizes high-frequency noise due to MC sampling. We found that our Fourier feature prefiltering helps suppress the noise since it biases updates towards low-frequency content.

5 Results

We first compare against prior methods under Gaussian smoothing (Sec. 5.1). We then study generalization to other filter families, Box and Lanczos (Sec. 5.2).

We evaluate on images and signed distance fields (SDFs). For images, we follow the NGSSF evaluation benchmark [Mujkanovic et al. 2024] on 100 high-resolution 2048×2048 Adobe FiveK images [Bychkovsky et al. 2011]. We treat signals as periodic and evaluate pixels whose convolution windows lie within the image boundaries. We report PSNR (Peak Signal-to-Noise Ratio), SSIM [Wang et al. 2004], and LPIPS [Zhang et al. 2018].

For SDFs, following the NGSSF benchmark [Mujkanovic et al. 2024] and training coordinate sampling, we voxelize meshes into SDFs at 1024^3 during training and 256^3 at test time. We extend the ground truth SDF values 1.2 times beyond the normalized coordinates, calculate filtered fields, and again crop borders. We use the Lucy, Dragon, Thai Statue, and Armadillo meshes from the Stanford 3D Scanning Repository (models courtesy of the © Stanford Computer Graphics Laboratory). We report MSE (Mean Squared Error), Chamfer distance [Fan et al. 2017], and IoU [Mescheder et al. 2019].

5.1 Gaussian Filtering

Image filtering. We report quantitative results for Gaussian smoothing in Table 1 (isotropic) and Table 2 (anisotropic). In addition, the first two rows (Gaussian) and first two columns (NGSSF and Trained w/ Gaussian) of Figure 6 show qualitative comparisons. We include additional qualitative results in the supplemental material.

In the isotropic setting, against the most relevant continuous-scale baselines—NFC [Nsampi et al. 2023] and NGSSF [Mujkanovic et al. 2024]—our method improves PSNR at least by **+4.9** to **+13.4** dB across scales. It also outperforms discrete-scale architectures (BACON [Lindell et al. 2022], MINER [Saragadam et al. 2022], INSP [Xu et al. 2022]) when filtering at varying scales. MINER gives higher PSNR on the *unfiltered* signal only (σ^2=0; +2.96 dB), which reflects differences in representation capacity rather than filtering behavior.

For anisotropic Gaussian smoothing, our method yields large, consistent gains over all baselines (Table 2; see also the Gaussian rows/columns in Figure 6). Quantitatively, we improve PSNR by **+14.5** dB over NGSSF [Mujkanovic et al. 2024] (49.33 vs. 34.82 dB), and by **+19.0**/**+25.2** dB over NFC [Nsampi et al. 2023]/PNF [Yang et al. 2022] respectively, with corresponding LPIPS and SSIM also best (*e.g.*, LPIPS 0.054 vs. 0.069 for NGSSF; SSIM 0.991 vs. 0.940).

Among prior methods, NGSSF is the strongest baseline for Gaussian filtering in both isotropic and anisotropic settings. However, its calibration scheme does not perfectly match the ground-truth filtering magnitude across scales and underfits frequency content which is visible near edges of the filtered signal.

SDF smoothing. We present the quantitative results of isotropic and anisotropic Gaussian filtering of SDFs in Tables 3 and 4 respectively. Also the first row (Gaussian) of Figure 7 shows qualitative SDF results.

Across isotropic kernels (Table 3), our method gives the best overall geometry—lowest Chamfer distance and highest IoU—at all non-zero blur levels. MINER reports the lowest MSE at $\sigma^2 \in \{0, 10^{-4}, 10^{-3}\}$ and NFC has the lowest MSE at $\sigma^2 = 10^{-2}$, but both either lack continuity in σ^2 or degrade geometry (higher Chamfer, lower IoU). In contrast, our model remains continuous in filter space and preserves geometry across scales. For anisotropic kernels (Table 4), our method improves over NGSSF and NFC by large margins on all metrics (*e.g.*, MSE 2.2×10^{-5} vs. 2.8×10^{-3} for NGSSF; Chamfer 3.6×10^{-3} vs. 1.2×10^{-1}; IoU 0.83 vs. 0.42), removing streaking and floaters while matching the ground-truth shape more closely.

5.2 Filter Generalization

We test generalization across low-pass filter families, Gaussian, Box, and Lanczos, under two training schemes: (i) training with a single filter family and evaluating on the others, and (ii) training jointly with multiple filter families, Gaussian, Box, and Lanczos, at each iteration. (See also the 1D example in Fig. 3).

Baselines. Neural Field Convolutions (NFC) [Nsampi et al. 2023] models a filter as a sparse set of Dirac impulses, interpolated through a piecewise-polynomial approximation to the target kernel. This construction yields continuity in *isotropic* and *axis-aligned* settings, but for general *anisotropic* kernels, it requires re-optimizing Dirac locations and weights for each covariance. Therefore, the representation is not continuous across anisotropic scale space. In practice, following the prescribed guidelines, we use piecewise-linear kernels only for 2D isotropic comparisons; for anisotropic cases, piecewise-constant kernels are required because higher-order models fail to optimize reliably. Even with substantial per-kernel tuning, representing non-polynomial filter families (*e.g.*, Gaussian and Lanczos) and highly anisotropic filters remain limited by the NFC construction (see the optimized Dirac deltas in Fig. 6).

Table 1. *Image filtering with isotropic Gaussian kernels.* We test our model across *isotropic* kernels by averaging metrics across 100 images on 4(+original signal) different isotropic kernels. Indicated by σ^2-cont., the NGSSF [Mujkanovic et al. 2024] and NFC [Nsampi et al. 2023] baselines are the only other methods that operate continuously in isotropic scale space. Other methods, namely BACON [Lindell et al. 2022], MINER [Saragadam et al. 2022], INSP [Xu et al. 2022], illustrate the tradeoff between changing the network architecture and filtering quality. Best and second best are **bold** and underlined. Our method outperforms these alternatives on filtering with changing scales. While MINER yields marginally higher PSNR at representing the original signal, it lacks continuous control over σ^2 scale space.

Method	σ^2-cont.	$\sigma^2 = 0$			$\sigma^2 = 10^{-4}$			$\sigma^2 = 10^{-3}$			$\sigma^2 = 10^{-2}$			$\sigma^2 = 10^{-1}$		
		PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
BACON [2022]	✗	32.89	0.308	0.823	38.95	0.235	0.955	36.48	0.123	0.953	30.59	0.086	0.895	25.36	0.100	0.601
MINER [2022]	✗	**41.19**	**0.088**	**0.963**	37.38	0.259	0.945	36.99	0.097	0.959	25.89	0.205	0.815	24.38	0.156	0.567
INSP [2022]	✗	30.57	0.454	0.770	30.14	0.420	0.838	23.77	0.546	0.725	20.75	0.546	0.627	23.37	0.381	0.633
NFC [2023]	✓	20.75	0.703	0.533	26.49	0.224	0.839	36.05	0.071	0.949	39.74	0.011	0.965	41.06	0.006	0.965
NGSSF [2024]	✓	33.85	0.305	0.854	35.05	0.207	0.942	34.74	0.077	0.954	35.06	0.023	0.949	34.99	0.020	0.878
Ours	✓	38.23	0.193	0.918	**43.83**	**0.192**	**0.971**	**48.91**	**0.064**	**0.991**	**53.09**	**0.009**	**0.997**	**53.81**	**0.005**	**0.997**

Table 2. *Image filtering with anisotropic Gaussian kernels.* We test our model across *anisotropic* kernels by averaging metrics across 100 images on 100 different anisotropic kernels with varying scales and orientations. Best scores are **bold**. We achieve greater accuracy on all metrics compared to prior work while remaining continuous in filter space.

	Σ-cont.	PSNR↑	LPIPS↓	SSIM↑
PNF [2022]	✗	24.15	0.571	0.704
NFC [2023]	✗	30.31	0.094	0.857
NGSSF [2024]	✓	34.82	0.069	0.940
Ours	✓	**49.33**	**0.054**	**0.991**

Neural Gaussian Scale-Space Fields (NGSSF) [Mujkanovic et al. 2024] learns a scale-conditioned MLP that is well suited to general low-pass filtering. However, it does not provide an explicit control to switch the *type* of smoothing at test time. To compare across families, we therefore recalibrate NGSSF's encoding with a Monte Carlo estimate of the filtered signal for the target family. While this calibration aligns the overall spectral shape, the nature of the smoothing (*e.g.*, Gaussian, Box, and Lanczos), it is not directly controllable and residual estimation errors remain (see Fig. 6).

Results on images. We assess generalization for the *image regression* experiments quantitatively in Table 5 and qualitatively in Figure 6. For this task, frequency characteristics are most visible in the frequency-domain insets (bottom right of each panel; see also Figs. 1 and 6). Single-family training already generalizes well to the other families; joint training provides only marginal gains. The Lanczos frequency response closely matches the reference spectrum with or without Lanczos-specific training (See supplemental material for more examples). However, errors in the Lanczos filter overall are higher than those in the other filters.

Results on SDFs. We present generalization capabilities of our model for SDFs in Figure 7. Our model matches overall smoothing shape and characteristics of each filter, and achieves perceptually convincing results. Fine-scale artifacts persist in some cases, which we attribute to training with higher MC variance on SDFs. Importance sampling high-variance regions such as sharp surface changes and better tuning Fourier basis scale to match signal frequency characteristics (see Sec. 4.3) help mitigate these artifacts.

Summary. Across both images and SDFs, our model trained with frequency-modulated features supports Gaussian, Box, and Lanczos filtering with anisotropic covariances in single forward pass, without architectural changes. Compared to NFC's Dirac-impulse parameterization (capacity-limited and not continuous in anisotropic scale space) and NGSSF's Gaussian-specific design (recalibration needed and no explicit kernel family control), our approach exposes the frequency response directly in the input encoding, which enables cross-family generalization and controllable smoothing at test time.

5.3 Ablations

In this section, we investigate design choices of our method and ablate them to observe their effectiveness. For each of these ablations, the model and the experiment parameters were the same as before.

Model parameters. To assess which components matter most for (i) reconstructing the original signal, (ii) isotropic filtering, and (iii) anisotropic filtering, we run ablations in Figure 4 by varying the Fourier-feature (embedding) size, the MLP hidden width, and the network depth. Although the model can reconstruct the original signal with relatively small embeddings (*e.g.*, NeRF uses m=10 [Mildenhall et al. 2021]), the embedding size is the most sensitive factor for filtering performance. Increasing hidden width also helps across all tests, while increasing depth beyond three layers yields only marginal gains. We perform a depth ablation with matched number of trainable parameters (see supplemental Fig.), varying network depth, and find that a three or four layer MLP yields the best reconstruction quality.

Model components. We ablate three components on the *Alien* image (Figure 5 and Table 6): (i) single-sample Monte Carlo supervision, (ii) exact Fourier down-weighting of the input features via $\mathcal{F}\{K_\Sigma\}$ (Sec. 4.1), and (iii) lifting the Lipschitz constraint used by NGSSF. MC alone yields only small gains in filtering quality (*e.g.*, isotropic PSNR 31.35 → 33.16). Exact Fourier down-weighting is effective only when we train with MC: without MC it changes little, while with MC it boosts filtering PSNR to 41.8–42.0 dB. Lifting the Lipschitz constraint further improves results under MC (cf. rows 4⟶5 and

Table 3. *SDF filtering with isotropic Gaussian kernels.* Metrics are averaged over *Lucy, Dragon, Thai Statue, Armadillo* SDFs. Columns correspond to $\sigma^2 \in \{0, 10^{-4}, 10^{-3}, 10^{-2}\}$; σ^2=0 denotes unfiltered reconstruction. "σ^2-cont." indicates continuity w.r.t. the scalar variance. Best and second best are **bold** and underlined. MINER reports the lowest MSE at $\sigma^2 \in \{0, 10^{-4}, 10^{-3}\}$ and NFC at $\sigma^2 = 10^{-2}$, whereas our method yields the best *geometry*—lowest Chamfer and highest IoU—at all non-zero blur levels while remaining continuous in σ^2.

Method	σ^2-cont.	$\sigma^2 = 0$			$\sigma^2 = 10^{-4}$			$\sigma^2 = 10^{-3}$			$\sigma^2 = 10^{-2}$		
		MSE↓	Cham.↓	IoU↑	MSE↓	Cham.↓	IoU↑	MSE↓	Cham.↓	IoU↑	MSE↓	Cham.↓	IoU↑
BACON [2022]	✗	2.5e-3	1.3e-3	**0.99**	4.0e-3	2.2e-3	0.97	8.3e-2	1.5e-2	0.84	2.6e-4	4.9e-2	0.53
MINER [2022]	✗	**1.6e-7**	**1.1e-3**	0.98	**3.3e-7**	1.4e-3	**0.98**	**4.1e-6**	8.0e-3	0.92	1.8e-4	6.1e-2	0.52
INSP [2022]	✗	1.2e-1	1.3e-3	**0.99**	4.3e-2	4.4e-3	0.95	3.6e-2	1.1e-2	0.88	3.1e-2	3.7e-2	0.64
NFC [2023]	✓	3.7e-3	5.7e-3	0.89	2.5e-5	4.8e-3	0.92	1.4e-5	2.2e-3	0.97	**1.0e-5**	2.3e-2	0.77
NGSSF [2024]	✓	8.3e-5	3.9e-3	0.94	6.0e-5	5.5e-3	0.92	6.5e-4	1.6e-2	0.83	1.1e-2	1.3e-1	0.32
Ours	✓	1.6e-5	1.7e-3	0.98	1.1e-5	**1.2e-3**	**0.98**	9.5e-6	**1.8e-3**	**0.98**	1.8e-5	**3.6e-3**	**0.95**

Table 4. *SDF filtering with anisotropic Gaussian kernels.* Metrics are averaged over covariances Σ on 100 different anisotropic kernels with varying scales and orientations. "Σ-cont." indicates continuity over the full covariance. Best scores are **bold**. Our method achieves greater accuracy on all metrics while remaining continuous in filter space.

	Σ-cont.	MSE↓	Cham.↓	IoU↑
NFC [2023]	✗	7.1e-2	4.6e-1	0.08
NGSSF [2024]	✓	2.8e-3	1.2e-1	0.42
Ours	✓	**2.2e-5**	**3.6e-3**	**0.83**

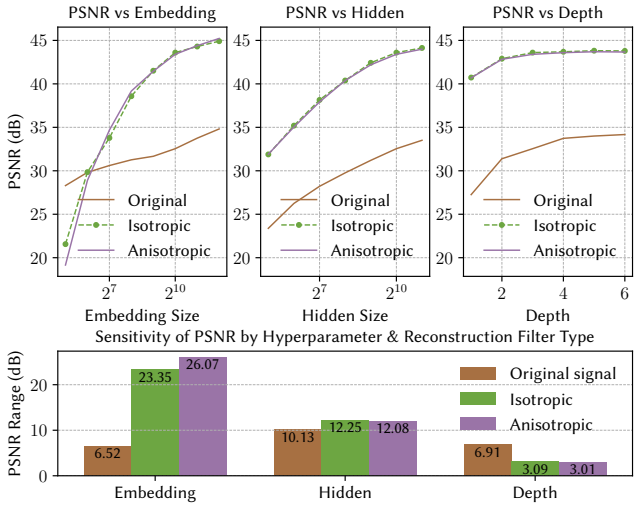

Fig. 4. *Sensitivity Analysis.* We evaluate how Fourier-feature embedding size and network architecture affect (1) reconstruction of the original signal, (2) isotropic smoothing, and (3) anisotropic smoothing. *Embedding Size.* While embedding size has relatively smaller effect on reconstructing the original signal, the model benefits substantially from larger embeddings in both isotropic and anisotropic smoothing tasks. *Hidden Dimension.* Increasing the hidden-layer width consistently yields higher PSNR across all tests. *Network Depth.* Adding more than three layers provides only marginal PSNR gains, indicating diminishing returns beyond a depth of three. The bottom row shows the PSNR range (highest minus lowest) observed over each hyperparameter sweep, illustrating the relative sensitivities.

6⟶7). With all three enabled, the model achieves the best scores on the original and filtered signals (32.54/43.40/43.60 dB PSNR with the lowest LPIPS and highest SSIM).

Table 7. *Effect of the number of MC samples.* We conduct an image regression experiment using lower-variance estimates of the filtered field, averaged over 100 anisotropic test-time Gaussian kernels and evaluated after 100,000 training iterations. We observe that, although increasing the number of Monte Carlo samples substantially reduces the variance of the signal estimate, this has a minimal impact on the overall quality of the reconstructed result at convergence.

	1 MC	4 MC	16 MC	64 MC	256 MC	1024 MC
PSNR ↑	43.64	44.32	44.57	44.70	44.68	44.73

Activation function. We integrate our exact Fourier encoding with a SIREN-style sine activation and achieve 43.43 dB PSNR on the Alien image under anisotropic smoothing, nearly identical to the 43.40 dB of our ReLU model.

Using a single Monte Carlo sample. Table 7 shows that presenting the network with a lower-variance estimate of the signal does not significantly improve the results, which validates our choice of using a single Monte Carlo sample.

6 Limitations

Training/Inference speed. Like all multilayer-perceptron-based neural fields, our method incurs higher per-sample evaluation cost compared to grid-based or other discrete representations (*e.g.*, hash grids or voxel grids). Compared to NGSSF, removing Lipschitz bounds reduces the training computation by 38.5% (130ms to 80ms forward + backward + optimization step). For inference time, our method is as fast or faster than NGSSF (1.8s), BACON (1.6s), PNF (5.7s), INSP (189.5s), NFC (63.9s). MINER (0.1s) remains faster at inference due to its multiple sparse tiny MLPs (0.1s vs 1.8s). This performance gap can limit deployment in real-time or latency-sensitive applications.

Only Symmetric filters. Our method relies on the filter's Fourier transform being real and even. Asymmetric filters introduce phase shifts that are not captured by our current encoding, so extending

Table 5. *Filter generalization.* We train four versions of our image-regression model—one each with Gaussian, Box, and Lanczos filters, plus a "Multiple" model trained on all three. At test time, we evaluate each model using Gaussian, Box, and Lanczos kernels. The results show that even models trained on single filters generalize well to unseen filters, nearly matching the quality of training on all filters simultaneously.

Training Filter	Gaussian			Box			Lanczos		
	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
Multiple	43.82	0.128	0.969	40.22	**0.111**	0.963	39.21	**0.104**	0.963
Gaussian	**44.71**	0.127	**0.972**	40.23	**0.111**	0.962	37.33	0.106	0.955
Box	43.81	**0.125**	**0.972**	**41.58**	0.113	**0.966**	36.09	0.108	0.948
Lanczos	38.03	0.128	0.952	35.24	0.120	0.938	**41.47**	0.107	**0.967**

Table 6. *Quantitative Ablation Study on Alien.* We analyze the impact of having one Monte Carlo sample, removing the Lipschitz constraint, and using our exact Fourier modulation. We observe that, while the Lipschitz-constrained MLP achieves better filtering without Monte Carlo, lifting it performs better once single Monte Carlo sampling is enabled. The exact Fourier modulation improves results in all cases. When the three components are enabled, the method achieves the best results in both original signal fitting and filtering.

Monte Carlo	Lipschitz Lift	Exact Fourier	Original			Anisotropic			Isotropic		
			PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑	PSNR↑	LPIPS↓	SSIM↑
✗	✗	✗	26.81	0.157	0.899	32.01	0.205	0.957	31.35	0.183	0.945
✗	✓	✗	29.30	0.046	0.946	22.58	0.374	0.883	21.34	0.331	0.839
✗	✗	✓	26.06	0.169	0.894	31.24	0.197	0.953	30.56	0.172	0.941
✓	✗	✗	25.98	0.166	0.909	32.38	0.405	0.928	33.16	0.296	0.921
✓	✓	✗	31.89	0.032	0.972	36.01	0.350	0.933	37.14	0.183	0.934
✓	✗	✓	27.01	0.127	0.928	42.00	0.185	0.973	41.80	0.159	0.967
✓	✓	✓	32.54	0.025	0.977	43.40	0.180	0.974	43.60	0.150	0.970

to arbitrary non-symmetric kernels would require modeling and compensating for phase information.

High-pass filter support. Our results only indicate performance on low-pass filters. A high-pass filter such as difference-of-Gaussians is readily feasible with our method as the filtered response can be achieved as a linear combination of Gaussians with two different scales. Since our method operates continuously in scale space, any such parametric combination is supported. We leave the investigation of supporting arbitrary and asymmetric filters as future work.

Scope and applicability. In this work we evaluate only MLP-based neural fields with ReLU and SIREN-like sinusoidal activations. Our approach relies on continuous inputs and analytic frequency responses, and is therefore not directly applicable to grid-parameterized methods (*e.g.*, multi-resolution or voxel/tensor grids) without additional changes. Extending the method to such grids is outside the scope of this paper.

Scope of analysis. We focus on the empirical evaluation of our training scheme and its practical generalization to unseen filters. A theoretical investigation into why Fourier-feature modulation so effectively represents multi-scale signals, and establishing the conditions under which it provably converges represent a promising future direction.

7 Conclusion

We have presented a simple yet powerful framework for prefiltering neural fields in the frequency domain by analytically modulating Fourier–feature embeddings with a family of symmetric filter kernels. By integrating a closed-form expression for the filter's frequency response into the first layer of a multi-layer perceptron and supervising with single-sample Monte Carlo estimates, our method supports continuous Gaussian, Box, and Lanczos filters, even those unseen at training time, without imposing architectural constraints or relying on precomputed multiscale datasets. Extensive experiments on 2D images and 3D signed-distance fields demonstrate that our approach delivers higher fidelity and generality than prior scale-aware neural-field methods, while remaining memory-efficient and straightforward to implement.

Acknowledgments

We thank Jaiden Ekgasit for help with figure preparation; Baha Eren Yaldiz for running experiments; and Yash Belhe, Sina Nabizadeh, Zilu Li, Kaiwen Jiang, Wesley Chang, Alex Trevithick, and Sumanth Varambally for comments and discussions. We also thank the anonymous reviewers for their extensive comments that greatly improved the clarity and analysis in the final paper.

This work was supported in part by ONR grant N00014-23-1-2526; NSF grants (2110409, 2212085, 2042583, 2238839); the European Research Council (ERC) Advanced Grant NERPHYS (101141721, https://project.inria.fr/nerphys/); gifts from Adobe, Google, and Qualcomm; the Ronald L. Graham Chair; and the UC San Diego

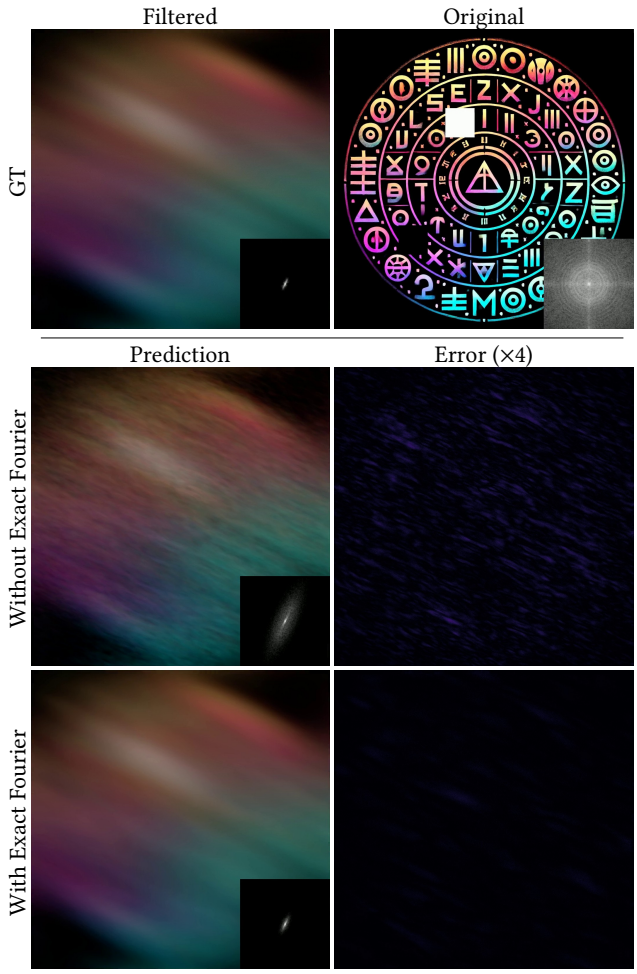

Fig. 5. *Effect of exact Fourier modulation.* We show the impact of exact Fourier modulation on the Alien image. The inset on the right bottom represents the frequency spectrum of the image. Our model produces much smoother filtering results when encoding is exactly modulated to match the filter's Fourier transform.

Center for Visual Computing. We also acknowledge NSF grants (2100237, 2120019) for the NRP Nautilus cluster.

References

E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. 1984. Pyramid methods in image processing. *RCA Engineer* 29, 6 (1984), 33–41.

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. 2021. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5855–5864.

Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. 2023. Zip-nerf: Anti-aliased grid-based neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19697–19705.

Yash Belhe, Michaël Gharbi, Matthew Fisher, Iliyan Georgiev, Ravi Ramamoorthi, and Tzu-Mao Li. 2023. Discontinuity-Aware 2D Neural Fields. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 42, 6, Article 217 (2023), 11 pages.

Sai Bi, Zexiang Xu, Pratul P. Srinivasan, Ben Mildenhall, Kalyan Sunkavalli, Milos Hasan, Yannick Hold-Geoffroy, David J. Kriegman, and Ravi Ramamoorthi. 2020. Neural Reflectance Fields for Appearance Acquisition. *CoRR* abs/2008.03824 (2020). arXiv:2008.03824 https://arxiv.org/abs/2008.03824

Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. 2011. Learning photographic global tonal adjustment with a database of input/output image pairs. In *CVPR 2011*. IEEE, 97–104.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 605–613.

Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J Zico Kolter. 2020. Multiplicative filter networks. In *International Conference on Learning Representations*.

Ned Greene and Paul S Heckbert. 1986. Creating raster Omnimax images from multiple perspective views using the elliptical weighted average filter. *IEEE Computer Graphics and Applications* 6, 6 (1986), 21–27.

Paul S. Heckbert. 1989. *Fundamentals of Texture Mapping and Image Warping*. Technical Report. EECS Department, University of California, Berkeley.

Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. 2018. Monte carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions on Graphics (tog)* 37, 6 (2018), 1–12.

Hugues Hoppe. 1996. Progressive meshes. In *SIGGRAPH*. 99–108.

Wenbo Hu, Yuling Wang, Lin Ma, Bangbang Yang, Lin Gao, Xiao Liu, and Yuewen Ma. 2023. Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19774–19783.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. 2022. BACON: Band-limited coordinate networks for multiscale scene representation. In *Computer Vision and Pattern Recognition*. 16252–16262.

Selena Ling, Merlin Nimier-David, Alec Jacobson, and Nicholas Sharp. 2025. Stochastic Preconditioning for Neural Field Optimization. *ACM Trans. Graph.* 44, 4 (2025).

Junchen Liu, Wenbo Hu, Zhuo Yang, Jianteng Chen, Guoliang Wang, Xiaoxue Chen, Yantong Cai, Huan-ang Gao, and Hao Zhao. 2024. Rip-NeRF: Anti-aliasing radiance fields with ripmap-encoded platonic solids. In *ACM SIGGRAPH 2024 Conference Papers*. 1–11.

G Lowe. 2004. Sift-the scale invariant feature transform. *Int. J* 2, 91-110 (2004), 2.

Youssef Mansour and Reinhard Heckel. 2023. Zero-shot noise2noise: Efficient image denoising without any data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14018–14027.

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

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

Felix Mujkanovic, Ntumba Elie Nsampi, Christian Theobalt, Hans-Peter Seidel, and Thomas Leimkühler. 2024. Neural Gaussian Scale-Space Fields. *ACM Transactions on Graphics* 43, 4 (2024).

Ntumba Elie Nsampi, Adarsh Djeacoumar, Hans-Peter Seidel, Tobias Ritschel, and Thomas Leimkühler. 2023. Neural Field Convolutions by Repeated Differentiation. *ACM Trans. Graph.* 42, 6, Article 206 (Dec 2023), 11 pages. doi:10.1145/3618340

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

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

Ali Rahimi and Benjamin Recht. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems* 20 (2007).

Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. 2019. Neural BTF Compression and Interpolation. *Comput. Graph. Forum (Proc. Eurographics)* 38, 2 (2019).

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

Dilip Sarwate. 2023. Convolution of squares / boxcars in functional form. Signal Processing Stack Exchange (QA). https://dsp.stackexchange.com/questions/83340/convolution-of-squares-boxcars-in-functional-form Licensed under CC BY-SA 4.0.

Shayan Shekarforoush, David Lindell, David J Fleet, and Marcus A Brubaker. 2022. Residual multiplicative filter networks for multiscale reconstruction. *Advances in Neural Information Processing Systems* 35 (2022), 8550–8563.

Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. 2021. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems* 34 (2021), 19313–19325.

Venkataram Sivaram, Tzu-Mao Li, and Ravi Ramamoorthi. 2024. Neural Geometry Fields For Meshes. In *SIGGRAPH Conference Proceedings*. Article 29, 11 pages.

Ying Song, Jiaping Wang, Li-Yi Wei, and Wencheng Wang. 2015. Vector Regression Functions for Texture Compression. *ACM Trans. Graph.* 35, 1, Article 5 (2015).

Kenneth O Stanley. 2007. Compositional pattern producing networks: A novel abstraction of development. *Genetic programming and evolvable machines* 8 (2007), 131–162.

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

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

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

Lance Williams. 1983. Pyramidal parametrics. In *Proceedings of the 10th annual conference on Computer graphics and interactive techniques*. 1–11.

Andrew P Witkin. 1987. Scale-space filtering. In *Readings in computer vision*. Elsevier, 329–332.

Liwen Wu, Sai Bi, Zexiang Xu, Fujun Luan, Kai Zhang, Iliyan Georgiev, Kalyan Sunkavalli, and Ravi Ramamoorthi. 2024. Neural Directional Encoding for Efficient and Accurate View-Dependent Appearance Modeling. In *CVPR*.

Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. 2022. Neural fields in visual computing and beyond. *Comput. Graph. Forum (Proc. Eurographics STAR)* 41, 2 (2022), 641–676.

Dejia Xu, Peihao Wang, Yifan Jiang, Zhiwen Fan, and Zhangyang Wang. 2022. Signal processing for implicit neural representations. *Advances in Neural Information Processing Systems* 35 (2022), 13404–13418.

Guandao Yang, Sagie Benaim, Varun Jampani, Kyle Genova, Jonathan Barron, Thomas Funkhouser, Bharath Hariharan, and Serge Belongie. 2022. Polynomial neural fields for subband decomposition and manipulation. *Advances in Neural Information Processing Systems* 35 (2022), 4401–4415.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

Denis Zorin, Peter Schröder, and Wim Sweldens. 1997. Interactive multiresolution mesh editing. In *SIGGRAPH*. 259–268.

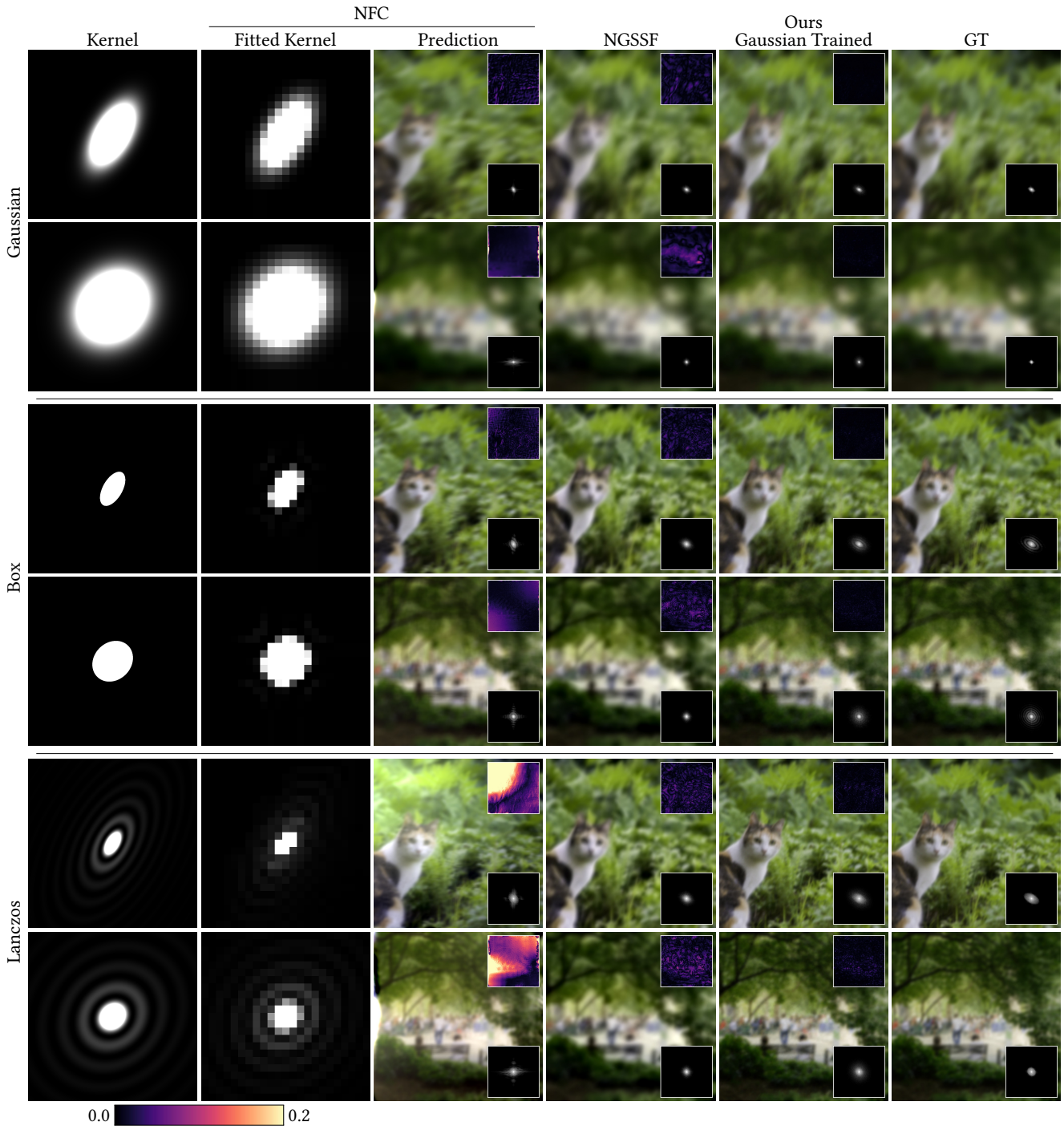

Fig. 6. Comparisons against Neural Field Convolutions (NFC) [Nsampi et al. 2023] and Neural Gaussian Scale-Space Fields (NGSSF) [Mujkanovic et al. 2024] for image filtering across Gaussian, Box, and Lanczos kernels. Our model supports controllable smoothing across families and anisotropic covariances in a single forward pass. NFC parameterizes filters with Dirac impulses; it is reliable for isotropic/mild kernels but is capacity-limited for anisotropic and non-polynomial families. NGSSF is tuned for Gaussian smoothing; we recalibrated its encoding for each family, but the filter family cannot be switched explicitly at test time. Bottom-right insets show frequency spectra; top-right insets show mean error. See the supplemental and website for additional results. Images from Adobe FiveK; © original photographers/Adobe.

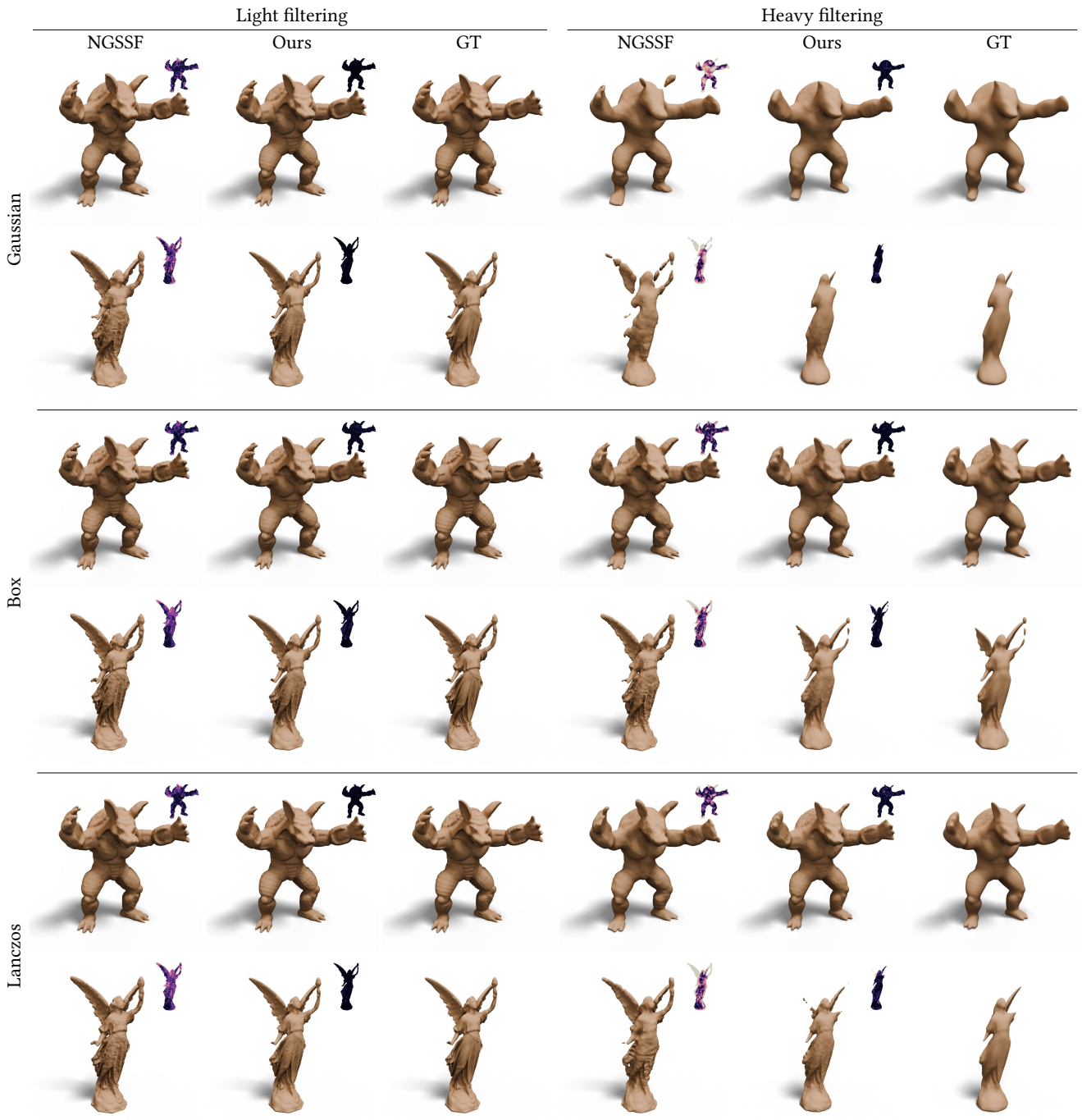

Fig. 7. *Filter generalization on SDFs.* Comparisons against recalibrated Neural Gaussian Scale-Space Fields [Mujkanovic et al. 2024] (NGSSF) for SDF filtering with different kernels. Insets at the top-right represent the distance to the ground truth mesh, darker is better. Our model shows closer geometry to the ground-truth compared to NGSSF. Mesh models from the Stanford 3D Scanning Repository; © Stanford Computer Graphics Laboratory.

A Convolution via Fourier Feature Encoding

A.1 Fourier transform of a Gaussian

Here we calculate the Fourier transform of the multivariate Gaussian with covariance matrix Σ^{-1} and Fourier basis vector $\mathbf{b}^\top$:

$$\mathcal{F}\{p_X\}(\mathbf{b}) = \int_{\mathbb{R}^d} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\tfrac{1}{2}\mathbf{x}^\top\Sigma^{-1}\mathbf{x}\right) e^{-2\pi i\mathbf{b}^\top\mathbf{x}}\, d\mathbf{x} \quad (16)$$

Once we combine the exponentials

$$= \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} \int_{\mathbb{R}^d} \exp\left(-\tfrac{1}{2}\mathbf{x}^\top\Sigma^{-1}\mathbf{x} - 2\pi i\mathbf{b}^\top\mathbf{x}\right)\, d\mathbf{x}. \quad (17)$$

We can define the quadratic form in the exponent:

$$-\tfrac{1}{2}\mathbf{x}^\top\Sigma^{-1}\mathbf{x} - 2\pi i\mathbf{b}^\top\mathbf{x} = -\tfrac{1}{2}\left(\mathbf{x} + i2\pi\Sigma\mathbf{b}\right)^\top\Sigma^{-1}\left(\mathbf{x} + i2\pi\Sigma\mathbf{b}\right) - 2\pi^2\mathbf{b}^\top\Sigma\mathbf{b}. \quad (18)$$

The integral becomes:

$$\exp\left(-2\pi^2\mathbf{b}^\top\Sigma\mathbf{b}\right) \int_{\mathbb{R}^d} \exp\left(-\tfrac{1}{2}\left(\mathbf{x} + i2\pi\Sigma\mathbf{b}\right)^\top\Sigma^{-1}\left(\mathbf{x} + i2\pi\Sigma\mathbf{b}\right)\right) d\mathbf{x}. \quad (19)$$

The resulting is another shifted Gaussian integral, which evaluates to the same normalization factor $(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}$, yielding:

$$\boxed{\mathcal{F}\{p_X\}(\mathbf{b}) = \exp\left(-2\pi^2\mathbf{b}^\top\Sigma\mathbf{b}\right).} \quad (20)$$

A.2 Fourier transform of normalized ellipsoid-shaped box filter

Consider an ellipsoid $E = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x}^\top\Sigma^{-1}\mathbf{x} \le 1\}$, where Σ is a symmetric positive-definite matrix. The Fourier transform of its characteristic function $\chi_E(\mathbf{x})$ is:

$$\mathcal{F}\{\chi_E\}(\mathbf{b}) = \int_E e^{-2\pi i\mathbf{b}^\top\mathbf{x}} d\mathbf{x}. \quad (21)$$

Diagonalize Σ as $\Sigma = Q\Lambda Q^\top$, where Q is orthogonal and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$. Let $y = Q^\top\mathbf{x}$, transforming the ellipsoid into a scaled unit ball:

$$y^\top\Lambda^{-1}y \le 1 \quad \Rightarrow \quad \sum_{i=1}^n \frac{y_i^2}{\lambda_i} \le 1. \quad (22)$$

We apply change of variables $z_i = \frac{y_i}{\sqrt{\lambda_i}}$, so $\sum_{i=1}^n z_i^2 \le 1$ (unit ball). The Jacobian determinant, which is product of square root of eigenvalues, is $|\Sigma|^{1/2}$, giving:

$$d\mathbf{x} = dy, \quad d\mathbf{x} = |\Sigma|^{1/2} dz. \quad (23)$$

Substitute $x = Qy = Q\sqrt{\Lambda}z$, and let $\eta = Q^\top\mathbf{b}$:

$$\mathcal{F}\{\chi_E\}(\mathbf{b}) = |\Sigma|^{1/2} \int_{B_1} e^{-2\pi i(\sqrt{\Lambda}z)^\top\eta} dz = |\Sigma|^{1/2} \int_{B_1} e^{-2\pi i z^\top\sqrt{\Lambda}\eta} dz. \quad (24)$$

The Fourier transform of the unit ball B_1 in $\mathbb{R}^n$ is known:

$$\int_{B_1} e^{-2\pi i z^\top\omega} dz = \frac{J_{n/2}(2\pi\|\omega\|)}{\|\omega\|^{n/2}}, \quad (25)$$

where $J_{n/2}$ is the Bessel function.

Let $\omega = \sqrt{\Lambda}\eta$, so $\|\omega\| = \sqrt{\eta^\top\Lambda\eta} = \sqrt{\mathbf{b}^\top\Sigma\mathbf{b}}$. We substitute variables and simplify further:

$$\mathcal{F}\{\chi_E\}(\mathbf{b}) = |\Sigma|^{1/2} \cdot \frac{J_{n/2}\left(2\pi\sqrt{\mathbf{b}^\top\Sigma\mathbf{b}}\right)}{\left(\sqrt{\mathbf{b}^\top\Sigma\mathbf{b}}\right)^{n/2}}. \quad (26)$$

In addition, let us calculate the **volume-normalized ellipsoid** $\tilde{\chi}_E(x) = \frac{1}{\mathrm{Vol}(E)}\chi_E(x)$, where $E = \{x \in \mathbb{R}^n : x^\top\Sigma^{-1}x \le 1\}$.

The volume of E is:

$$\mathrm{Vol}(E) = V_n|\Sigma|^{1/2}, \quad \text{where } V_n = \frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)}, \quad (27)$$

and V_n is the volume of the unit n-ball.

Substituting and simplifying the constants:

$$\boxed{\mathcal{F}\{\tilde{\chi}_E\}(\mathbf{b}) = \frac{\Gamma\left(\frac{n}{2}+1\right)}{\pi^{n/2}} \frac{J_{n/2}\left(2\pi\sqrt{\mathbf{b}^\top\Sigma\mathbf{b}}\right)}{\left(\sqrt{\mathbf{b}^\top\Sigma\mathbf{b}}\right)^{n/2}}} \quad (28)$$

The Fourier transform of the ellipsoid's characteristic function involves Bessel functions, reflecting its sharp boundary. Unlike Gaussians, which remain Gaussian under Fourier transforms, the ellipsoid's oscillatory frequency response encodes its geometric shape through Σ. The eigenvalues of Σ scale the frequencies, while the Bessel function $J_{n/2}$ captures radial symmetry through $\mathbf{b}^\top\Sigma\mathbf{b}$.

Let's substitute and derive normalized Fourier transform expression of an ellipsoid into 2D and 3D cases.

Case 1: 2D (n=2) $\Gamma\left(\frac{2}{2}+1\right) = \Gamma(2) = 1$:

$$\boxed{\mathcal{F}\{\tilde{\chi}_E\}(\mathbf{b}) = \frac{1}{\pi\sqrt{\mathbf{b}^\top\Sigma\mathbf{b}}} J_1\left(2\pi\sqrt{\mathbf{b}^\top\Sigma\mathbf{b}}\right)} \quad (29)$$

Case 2: 3D (n=3) $\Gamma\left(\frac{3}{2}+1\right) = \Gamma\left(\frac{5}{2}\right) = \frac{3}{4}\sqrt{\pi}$, and the Bessel function $J_{3/2}(x)$ can be expressed as $J_{3/2}(x) = \sqrt{\frac{2}{\pi x}}\left(\frac{\sin x}{x} - \cos x\right)$.

$$\boxed{\mathcal{F}\{\tilde{\chi}_E\}(\mathbf{b}) = \frac{3}{4\pi\left(\sqrt{\mathbf{b}^\top\Sigma\mathbf{b}}\right)^{3/2}} J_{3/2}\left(2\pi\sqrt{\mathbf{b}^\top\Sigma\mathbf{b}}\right)} \quad (30)$$

A.3 Fourier transform of the Lanczos kernel

For a single-dimensional variable x, we write the (untruncated) Lanczos kernel of order a, $a > 0$ with normalized sinc functions as:

$$L(x) = \mathrm{sinc}(x)\, sinc(x/a) = \frac{a\sin\pi x \sin\pi x/a}{\pi^2x^2} \quad (31)$$

Note that this is the pointwise multiplication of two sinc functions. It is known that the Fourier transform of this function is a scaled rectangle function, which for a positive constant c is as follows:

$$\mathrm{sinc}(cx) = \frac{1}{c}\mathrm{rect}\left(\frac{b}{c}\right) \quad (32)$$

where

$$\mathrm{rect}(x) = \begin{cases} 1 & : \ |x| \le \frac{1}{2} \\ 0 & : \ \text{else} \end{cases} \quad (33)$$

The Lanczos kernel is the primal-domain multiplication of two sinc functions; it is therefore equivalently the frequency-domain convolution of two rectangle functions

$$a \int_{-\infty}^{\infty} \mathrm{rect}(b - b'/\pi)\, \mathrm{rect}(ab'/\pi)\; db' \tag{34}$$

which is a trapezoid (or a triangle wave if both rectangles coincide exactly).

Since a piecewise definition of this trapezoid is unwieldy, we refer the reader to the description of the trapezoid given in [Sarwate 2023] for more detail. In particular, for rectangle functions of widths A and B, this trapezoidal function begins from 0 at $b = -\frac{1}{2}(A+B)$, rising to a maximum of $\min(A,B)$ with a slope of 1 at $-\frac{1}{2}|A-B|$. This is even in b, so the construction of the positive-domain part of this function is analogous. We identified the following function which matches the above description:

$$\mathcal{F}(L)(b) = a \max\left(\min\left(\frac{(a+1)}{2a} \quad -|b|, \min\left(1, \frac{1}{a}\right)\right), 0\right) \tag{35}$$

For higher-dimensional cases, we consider the Lanczos kernel as a radial kernel in $\|\mathbf{x}\| = \sqrt{\mathbf{x}^\top \mathbf{x}}$, which is then shifted by a covariance Σ with determinant $|\Sigma|$. To find the Fourier transform, we make the change-of-variable $\mathbf{x} \to \Sigma^{1/2}\mathbf{u}$ as before so that $d\mathbf{x} = |\Sigma|^{1/2} d\mathbf{u}$ and use the property that the Fourier transform of a function radial in $\mathbf{x}$ is also radial in $\mathbf{b}$:

$$\mathcal{F}(L(\|\Sigma^{-1/2}\mathbf{x}\|)) = \int_{\mathbb{R}^2} L(\|\Sigma^{-1/2}\mathbf{x}\|) \exp(-2\pi i \mathbf{b}^\top \mathbf{x})\; d\mathbf{x} \tag{36}$$

$$= |\Sigma|^{1/2} \int_{\mathbb{R}^2} L(\|\mathbf{u}\|) \exp(-2\pi i (\Sigma^{1/2}\mathbf{b})^\top \mathbf{u})\; d\mathbf{u} \tag{37}$$

$$= |\Sigma|^{1/2} \mathcal{F}(L)(\|\Sigma^{1/2}\mathbf{b}\|) \tag{38}$$

We combine this with the previous result to obtain the Fourier transform of our multidimensional Lanczos kernel:

$$\mathcal{F}(L)(\mathbf{b}) = a|\Sigma|^{1/2} \max\left(\min\left(\frac{(a+1)}{2a} \quad -\|\Sigma^{1/2}\mathbf{b}\|, \min\left(1, \frac{1}{a}\right)\right), 0\right) \tag{39}$$

During rejection sampling, we draw samples from the distribution proportional to the *normalized absolute value* of the Lanczos kernel, i.e., $p(x) \propto |L(x)|$. Therefore, we need the total mass of normalization in our encoding: $\int_{\mathbb{R}} |L(x)|\, dx$. To balance accuracy and cost, we restrict sampling to $x \in [-2\sqrt{a}, 2\sqrt{a}]$, which captures most of the kernel's mass. For our Fourier encoding, however, we still assume infinite support: truncating in space would convolve the ideal trapezoidal spectrum with a sinc induced by the window, an effect that is negligible at this truncation radius. Given area can be defined as:

$$\int_{\mathbb{R}^n} \left|\mathrm{sinc}(\pi\|\Sigma^{-1/2}\mathbf{x}\|)\, \mathrm{sinc}(\pi\|\Sigma^{-1/2}\mathbf{x}\|/a)\right| \; d\mathbf{x} \tag{40}$$

Radial reduction (general n). With the change of variables $\mathbf{x} = \Sigma^{1/2}\mathbf{u}$ so that $d\mathbf{x} = |\Sigma|^{1/2}\, d\mathbf{u}$ and $\|\Sigma^{-1/2}\mathbf{x}\| = \|\mathbf{u}\|$, and writing $r = \|\mathbf{u}\|$, we switch to n-D spherical coordinates:

$$= \int_{\mathbb{R}^n} \left|\mathrm{sinc}(\|\Sigma^{-1/2}\mathbf{x}\|)\; \mathrm{sinc}(\|\Sigma^{-1/2}\mathbf{x}\|/a)\right| d\mathbf{x} \tag{41}$$

$$= |\Sigma|^{1/2}\, S_{n-1} \int_0^{2\sqrt{a}} \left|\mathrm{sinc}(r)\; \mathrm{sinc}(r/a)\right| r^{n-1}\, dr, \tag{42}$$

where $S_{n-1} = \frac{2\pi^{n/2}}{\Gamma(n/2)}$ is the surface area of the unit $(n-1)$-sphere. Finally we define the Fourier transform of our Lanczos kernel as:

$$\boxed{\mathcal{F}\{L\}(\mathbf{b}_i) = \frac{a}{z_n} \max\left(\min\left(\frac{a+1}{2a} - \sqrt{\mathbf{b}_i^\top \Sigma \mathbf{b}_i},\; \min\left(1, \frac{1}{a}\right)\right),\; 0\right)} \tag{43}$$

We numerically estimate $z_1 = 0.9499393398$, $z_2 = 0.9913304793$, $z_3 = 1.2732395447$ following: $z_n = S_{n-1}\int_0^{2\sqrt{a}} \left|\mathrm{sinc}(r)\; \mathrm{sinc}(r/a)\right| r^{n-1}\, dr$.

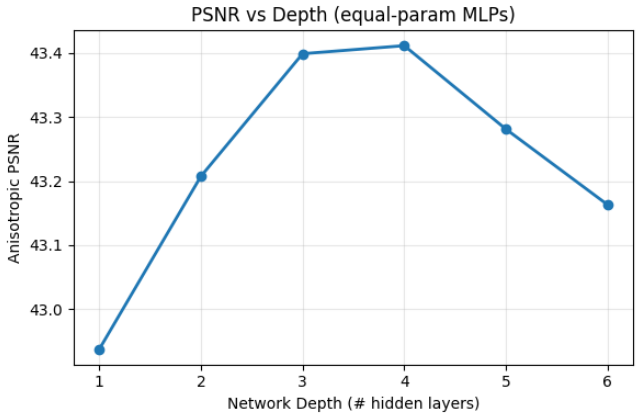

Fig. 8. *Depth ablation at matched parameter counts.* We vary MLP depth while holding total parameters constant (adjusting width accordingly); a three-layer MLP achieves the optimal reconstruction quality.

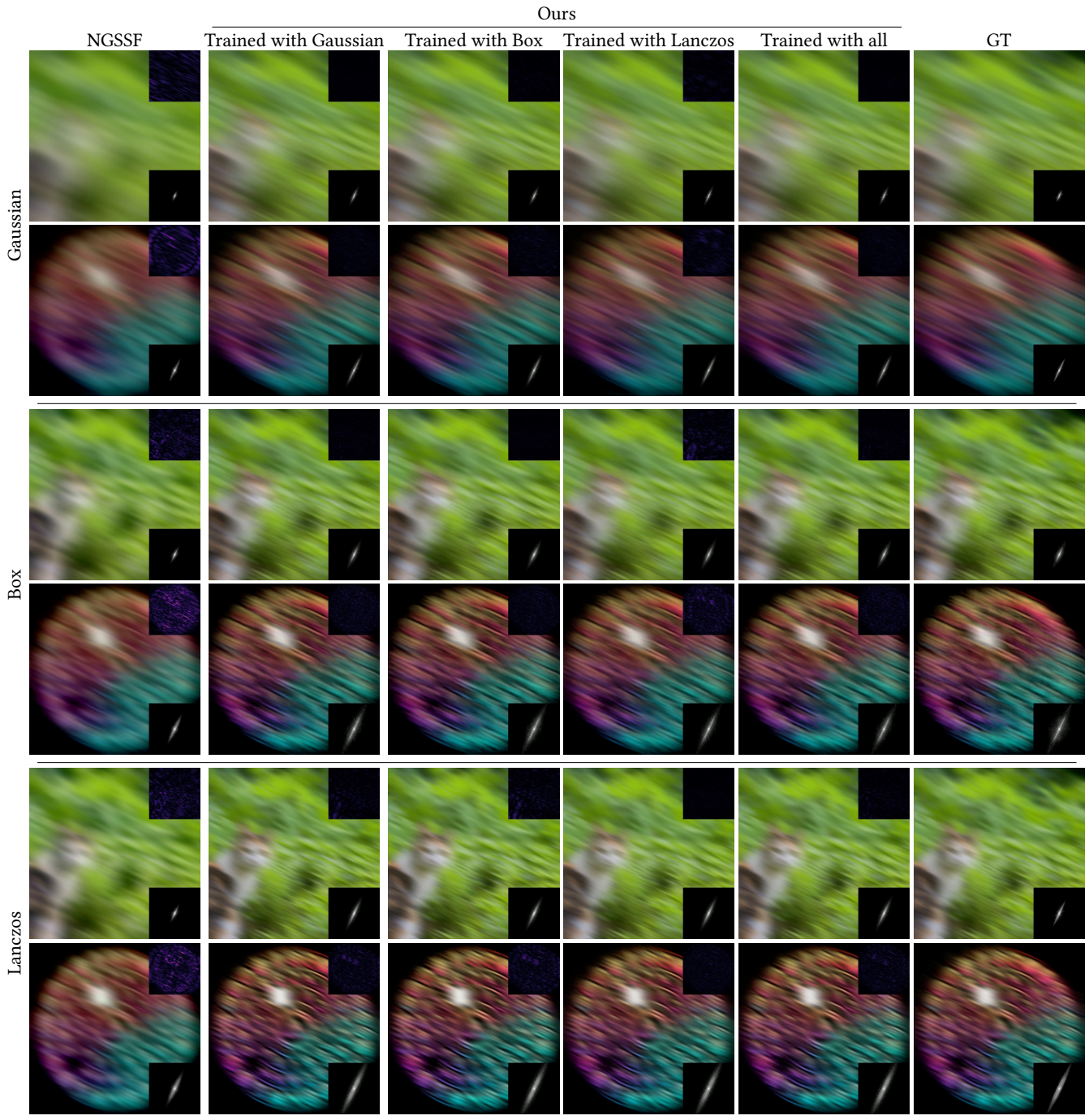

Fig. 9. *Filter generalization on Images.* Comparisons between different training/testing pairs and recalibrated Neural Gaussian Scale-Space Fields [Mujkanovic et al. 2024] (NGSSF) for image filtering with different kernels. Insets at the bottom-right of each image represent the frequency spectrum, and insets at the top-right represent the absolute mean error map. Our model can generalize to unseen filters at testing time with less error compared to NGSSF. Images from Adobe FiveK; © original photographers/Adobe.

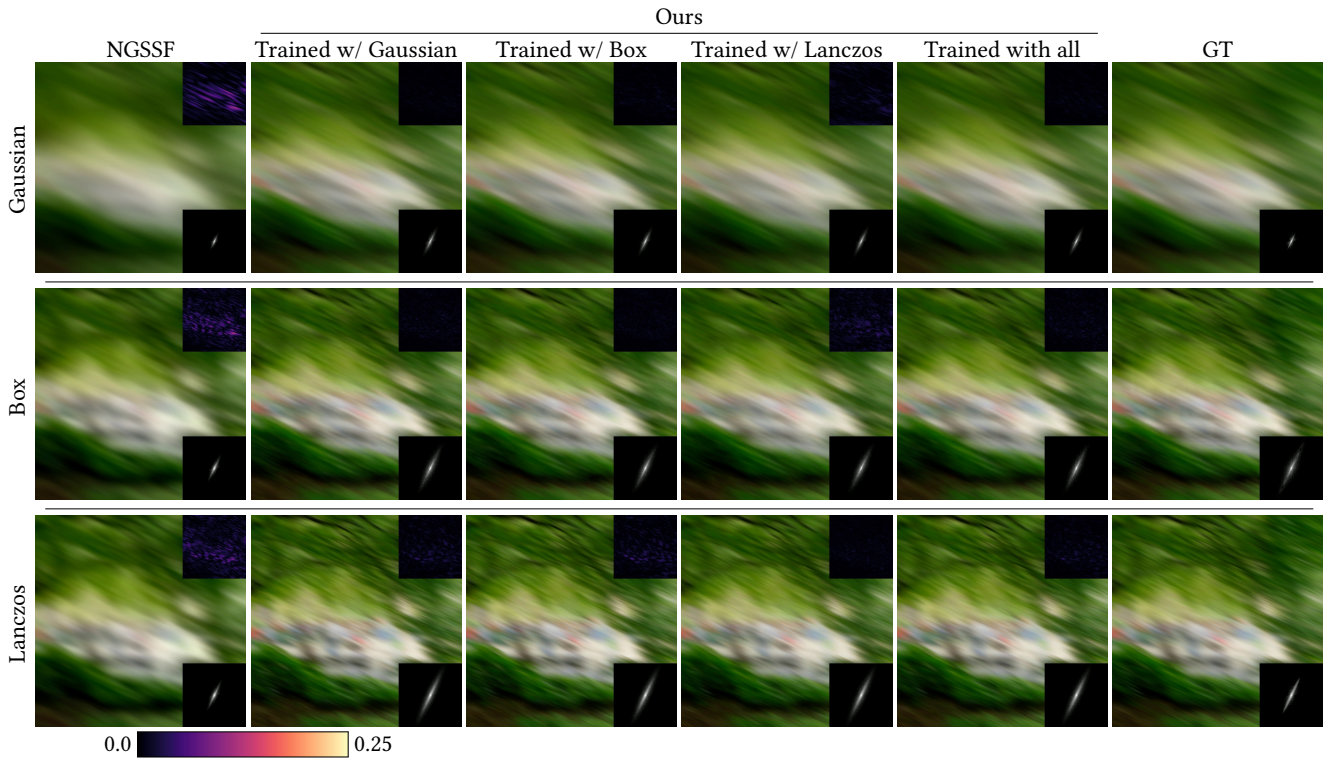

Fig. 10. *Additional filter generalization comparisons.* Images from Adobe FiveK; © original photographers/Adobe.