Neural Gaussian Scale-Space Fields

FELIX MUJKANOVIC, Max-Planck-Institut für Informatik, Germany
NTUMBA ELIE NSAMPI, Max-Planck-Institut für Informatik, Germany
CHRISTIAN THEOBALT, Max-Planck-Institut für Informatik, Germany
HANS-PETER SEIDEL, Max-Planck-Institut für Informatik, Germany
THOMAS LEIMKÜHLER, Max-Planck-Institut für Informatik, Germany

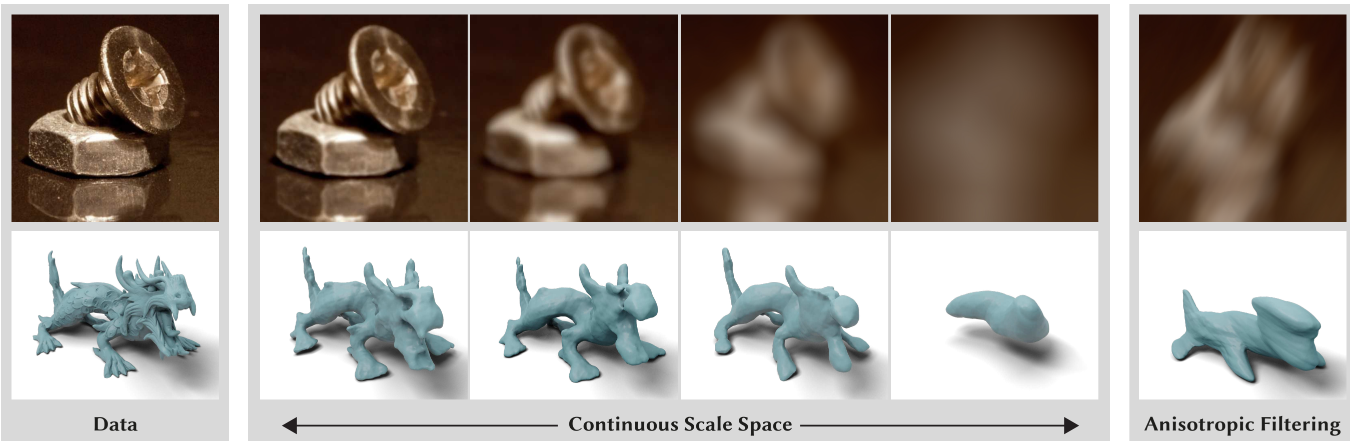

Fig. 1. We learn neural fields that capture continuous, anisotropic Gaussian scale spaces. Given a training signal, such as an image or geometry (left), we learn a neural field representation that allows continuous Gaussian smoothing (center). Crucially, this representation is learned self-supervised, i.e., without ever filtering the training signal. Our scale spaces are continuous in all parameters, including arbitrary covariance matrices that allow anisotropic filtering (right).

Gaussian scale spaces are a cornerstone of signal representation and processing, with applications in filtering, multiscale analysis, anti-aliasing, and many more. However, obtaining such a scale space is costly and cumbersome, in particular for continuous representations such as neural fields. We present an efficient and lightweight method to learn the fully continuous, anisotropic Gaussian scale space of an arbitrary signal. Based on Fourier feature modulation and Lipschitz bounding, our approach is trained self-supervised, i.e., training does not require any manual filtering. Our neural Gaussian scale-space fields faithfully capture multiscale representations across a broad range of modalities, and support a diverse set of applications. These include images, geometry, light-stage data, texture anti-aliasing, and multiscale optimization.

CCS Concepts: • **Computing methodologies → Neural networks**; **Computer graphics**.

Additional Key Words and Phrases: Gaussian filters, Scale spaces, Neural fields, Positional encoding, Lipschitz continuity, Matrix exponential, Signal processing, Image processing, Geometry processing

Authors' addresses: Felix Mujkanovic, Max-Planck-Institut für Informatik, Germany, felix.mujkanovic@mpi-inf.mpg.de; Ntumba Elie Nsampi, Max-Planck-Institut für Informatik, Germany, nnsampi@mpi-inf.mpg.de; Christian Theobalt, Max-Planck-Institut für Informatik, Germany, theobalt@mpi-inf.mpg.de; Hans-Peter Seidel, Max-Planck-Institut für Informatik, Germany, hpseidel@mpi-sb.mpg.de; Thomas Leimkühler, Max-Planck-Institut für Informatik, Germany, thomas.leimkuehler@mpi-inf.mpg.de.

ACM Reference Format:
Felix Mujkanovic, Ntumba Elie Nsampi, Christian Theobalt, Hans-Peter Seidel, and Thomas Leimkühler. 2024. Neural Gaussian Scale-Space Fields. *ACM Trans. Graph.* 43, 4, Article 134 (July 2024), 15 pages. https://doi.org/10.1145/3658163

1 INTRODUCTION

Continuous neural representations, so-called neural fields, are becoming ubiquitous in visual-computing research and applications [Tewari et al. 2022; Xie et al. 2022]. At their core, they map coordinates to signal values using a neural network. The generality, compactness, and malleability of this continuous data structure make them a popular choice for representing a broad variety of modalities. For example, neural fields have been used to represent geometry [Park et al. 2019], images [Stanley 2007], radiance fields [Mildenhall et al. 2020], flow [Park et al. 2021], reflectance [Gargan and Neelamkavil 1998], and much more.

In its basic form, a trained neural field allows querying the *original* signal value for a given coordinate. Oftentimes, however, this functionality is not sufficient: Many important use cases require *low-pass filtered versions* of the signal, arising from a custom band-limiting kernel that might be spatially-varying and even anisotropic. For example, a downstream application might require querying the signal at different scales [Marr and Hildreth 1980; Starck et al. 1998], or a subsequent discretization stage demands careful pre-filtering for anti-aliasing [Antoniou 2006; Greene and Heckbert 1986].

Scale-space theory [Iijima 1959; Koenderink 1984; Lindeberg 2013; Witkin 1987] provides a principled framework for tackling this problem. A linear scale-space representation is obtained by creating a family of progressively Gaussian-smoothed versions of a signal. This effectively leads to a progressive suppression of fine-scale structures. Once this representation is created, filtering boils down to merely querying the scale space at the required location. In this work, we set out to develop a lightweight and efficient method to learn a neural field that captures a fully continuous, anisotropic Gaussian scale space in a self-supervised manner.

Obtaining such a representation is non-trivial, as naïvely executing large-scale (anisotropic) Gaussian convolutions is highly inefficient. The task is even more challenging in neural fields, as they typically only allow point-wise function evaluations. Gaussian-weighted aggregation can be achieved using Monte Carlo estimation, but this requires trading high computational cost against high variance. Recently, convolutions have been rather efficiently executed in neural fields using differentiation strategies [Nsampi et al. 2023; Xu et al. 2022]. Yet, these solutions only support either fixed small-scale kernels and require costly repeated automatic differentiation [Xu et al. 2022], or are limited to axis-aligned kernels and demand multiple forward passes per filter location [Nsampi et al. 2023]. Neural fields can directly learn a continuous, isotropic scale space [Barron et al. 2021, 2022], but typically require dense supervision across scales. Specialized network architectures allow to learn an explicit decomposition of the signal into different frequency bands [Fathony et al. 2020; Lindell et al. 2022; Saragadam et al. 2022; Shekarforoush et al. 2022], but only support a coarse, discrete set of isotropic scales.

In this work, we present a novel approach for a neural field to learn a complete anisotropic Gaussian scale space that is applicable to arbitrary signals and modalities (Fig. 1). Different from previous works, our scale spaces are *fully continuous in all parameters*, i.e., both in signal coordinates and in arbitrary Gaussian covariance matrices. This allows fine-grained, spatially-varying (pre-)filtering using only a single forward pass. Crucially, training is self-supervised, i.e., we do not require supervision from filtered versions of the training signal, facilitating lightweight training and, consequently, a broad applicability of our method.

We observe that a positional encoding in the form of Fourier features [Hertz et al. 2021; Rahimi and Recht 2007; Tancik et al. 2020] provides a convenient means to modulate frequency content. However, the key ingredient to allow high-quality low-pass filtering of a signal is to pair this encoding with a Lipschitz-bounded multi-layer perceptron (MLP) [Gouk et al. 2021; Miyato et al. 2018; Szegedy et al. 2013]. We show that this regularization translates the dampening of encoding frequencies into a Gaussian smoothing of the signal. Training such an MLP with anisotropically modulated Fourier features on the *original* training signal forces the network to learn a Gaussian scale space, without requiring any manual filtering. After training, a calibration stage maps modulation parameters to Gaussian variance, based on ultra-lightweight, one-time Monte Carlo estimates of Gaussian convolutions. At inference, filtered versions of the learned signal can be synthesized in a single forward pass using arbitrary, continuous Gaussian covariance matrices.

We evaluate the accuracy and applicability of our approach on a broad variety of tasks and modalities. This includes anisotropic smoothing of images and geometry; (pre-)filtering of textures and light-stage data; spatially varying filtering; and multiscale optimization. We further analyze the interplay between Fourier features and Lipschitz-bounded MLPs to elucidate the combined effect of the central ingredients of our approach.

In summary, our contributions are:

- A novel approach for learning a fully continuous, anisotropic Gaussian scale space in a general-purpose neural field representation.
- An effective and efficient training methodology to achieve this goal self-supervised, i.e., without the requirement to filter the training data.
- The application and careful evaluation of our method on a spectrum of relevant modalities and tasks.

2 RELATED WORK

Here, we review related work on classical and neural multiscale representations (Sec. 2.1), the use of Fourier features in neural fields (Sec. 2.2), and Lipschitz bounds in deep learning (Sec. 2.3). For a comprehensive overview of neural fields in visual computing, we refer to recent surveys [Tewari et al. 2022; Xie et al. 2022].

2.1 Multiscale Signal Representations

Representing a signal at multiple scales has a long history in signal processing and visual computing, with the concept of scale spaces [Iijima 1959; Koenderink 1984; Lindeberg 2013; Witkin 1987] at the center of attention. Many different scale spaces can be constructed from a signal [Dorst and Van den Boomgaard 1994; Florack et al. 1995; Weickert 1998] using a rigorous axiomatic foundation [Lindeberg 1997]. Of special interest is the linear form, i.e., the convolution of the signal with a family of Gaussian kernels, as it exhibits a number of useful and well-studied properties, such as predictable behavior after differentiation [Babaud et al. 1986].

Scale spaces are typically constructed using various flavors of discretization. The convolution of a discrete signal with a discretized Gaussian kernel can be executed using cubature, but comes at high computational cost, especially in higher dimensions. Acceleration strategies involve the discrete Fourier transform [Brigham 1988] or exploiting the separability of Gaussians [Geusebroek et al. 2003]. In low dimensions, pyramidal structures (MIP mapping) [Burt 1981; Williams 1983] are even more efficient. Here, the isotropic scale parameter, i.e., the pyramid level, is also discrete. Anisotropic filtering using pyramids (RIP mapping) exists [Simoncelli and Freeman 1995], but comes at the cost of an additional coarse discretization of filter orientation, which can be hidden using carefully designed steerable filters [Freeman and Adelson 1991]. A different line of work has successfully explored signal representations using a discrete set of multiscale basis functions [Daubechies 1988; Guo et al. 2006; Mallat 1989]. In contrast to all these works, our approach is fully continuous in all dimensions.

Continuous representations impose significant challenges for multiscale techniques, and a dominant strategy for filtering is stochastic multi-sampling [Barron et al. 2023; Hermosilla et al. 2018; Ma et al. 2022; Shocher et al. 2020; Wang et al. 2018]. Such Monte Carlo approaches require a high number of samples to avoid objectionable noise. Sample count can be significantly reduced by relying on differentiation and integration properties of convolutions [Nsampi et al. 2023]. However, learning the required integral representation is costly, high-quality Gaussian filtering still requires a substantial number of network evaluations, and general anisotropic filtering requires pre-computing many kernel shapes. Our neural fields are easy to train and allow arbitrary anisotropic Gaussian filtering using just a single forward pass. A different strategy relies on approximating continuous filtering using a learned linear combination of derivatives of the signal obtained via automatic differentiation [Xu et al. 2022]. Different from our solution, this approach only supports small filter kernels.

Pre-filtering for anti-aliasing in continuous neural representations has recently received a lot of attention [Barron et al. 2021, 2022; Hu et al. 2023; Nam et al. 2023]. Similar to our approach, these solutions employ carefully crafted inductive biases that help learn a multiscale representation. However, they rely on supervision across scales, such as images of scene objects captured at different distances. In contrast, our method allows to learn a full scale space from a single-scale supervision signal, thereby significantly extending its applicability to a broad range of signals and modalities.

Strong architectural inductive biases allow training neural networks with intermediate activations that represent progressively band-limited versions of the learned signal [Fathony et al. 2020; Lindell et al. 2022; Shekarforoush et al. 2022], with applications in coarse-to-fine learning [Karras et al. 2018; Xiangli et al. 2022]. This leads to a discretization of scales and typically only allows isotropic filtering with a sinc-kernel. Extending this scheme to the anisotropic case is possible [Yang et al. 2022], but, similar to RIP mapping, it requires an additional coarse discretization of filter orientation, limiting this approach to low dimensions. The discretization of scales can be combined with neuroexplicit architectures, e.g., via spatial discretization or subdivision [Saragadam et al. 2022; Takikawa et al. 2022], and many corresponding domain-specific solutions exist [Chen et al. 2021, 2023a; Gauthier et al. 2022; Kuznetsov et al. 2021; Paz et al. 2022; Takikawa et al. 2021; Xu et al. 2021; Zhuang et al. 2023]. Yet, none of them allows fully continuous, arbitrary, anisotropic Gaussian filtering.

2.2 Fourier Features in Neural Fields

With early applications in time series analysis and representation learning [Kazemi et al. 2019; Vaswani et al. 2017; Xu et al. 2019], Fourier features [Rahimi and Recht 2007] are now a popular tool for learning neural-field representations of signals [Mildenhall et al. 2020]. Also referred to as positional encoding, their ability to map coordinates to latent features of different frequencies is an effective remedy for the spectral bias of neural networks [Rahaman et al. 2019]. Tancik et al. [2020] have analyzed the properties of such an encoding for neural fields using the neural tangent kernel [Jacot et al. 2018], and propose the use of normally distributed frequency vectors. Many applications rely on carefully dampened Fourier features to increase training stability [Hertz et al. 2021; Lin et al. 2021; Park et al. 2021; Yang et al. 2023], or to obtain a multiscale representation given a multiscale supervision signal [Barron et al. 2021, 2022]. Further, the analytical structure of Fourier features has been exploited for alias-free image synthesis [Karras et al. 2021]. We employ dampened Fourier features with carefully chosen frequency vectors as well, and combine this encoding with a Lipschitz-bounded network to obtain a Gaussian scale space.

Fourier features have been explored in different architectural variants. Examples of this scheme include periodic activation functions [Mehta et al. 2021; Sitzmann et al. 2020], Wavelet-style spatio-spectral encodings [Wu et al. 2023], or the modulation of unstructured representations based on radial basis functions [Chen et al. 2023b]. Different from our approach, the goal of these works is to improve single-scale reconstruction quality of complex signals.

2.3 Lipschitz Networks and Matrix Parameterizations

Neural networks with guaranteed Lipschitz bounds have numerous applications, such as robustness [Cisse et al. 2017; Hein and Andriushchenko 2017], smooth interpolation [Liu et al. 2022], and generative modeling [Arjovsky et al. 2017]. Technically, a desired Lipschitz constant can be enforced on the level of individual weight matrices. Corresponding methods can be divided into two classes.

The first class of methods relies on variants of projected gradient descent, where weight matrices are projected towards the closest feasible solution for each optimization step [Szegedy et al. 2013]. This can be done using spectral normalization [Behrmann et al. 2019; Gouk et al. 2021; Miyato et al. 2018; Yang et al. 2021; Yoshida and Miyato 2017], or using more sophisticated projections [Cisse et al. 2017] relying on orthonormalization [Björck and Bowie 1971]. The second class of methods reparameterize the weight matrices such that an unconstrained optimization can be applied [Anil et al. 2019; Liu et al. 2022]. Our method relies on this approach, as we observe that it leads to controllable training dynamics, but the choice of matrix parameterization is crucial for numerical stability.

The singular value decomposition is a convenient tool in this context [Mathiasen et al. 2020; Zhang et al. 2018b]. As it requires a parameterization of orthogonal matrices, ad-hoc parameterizations for special orthogonal matrices have been considered [Arjovsky et al. 2016; Helfrich et al. 2018; Huang et al. 2018; Jing et al. 2017]. More general solutions rely on Householder reflections [Mathiasen et al. 2020; Mhammedi et al. 2017; Zhang et al. 2018b], but they exhibit unfavourable properties when used within an optimization loop. We rely on matrix exponentials [Hyland and Rätsch 2017; Lezcano-Casado and Martınez-Rubio 2019], which have been shown to outperform other approaches [Golinski et al. 2019]. Yet, to the best of our knowledge, we are the first to use this set of techniques in the context of Lipschitz-bounded neural networks.

3 PRELIMINARIES

Here, we introduce concepts relevant to our approach. We first establish our notation for signals and fields, before reviewing technical background on Gaussian scale spaces, Fourier features, and Lipschitz continuity.

Signals and fields. We are concerned with arbitrary continuous signals $f \in \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{d_o}$, where d_i and d_o are typically rather small. This generic formulation captures many modalities in visual

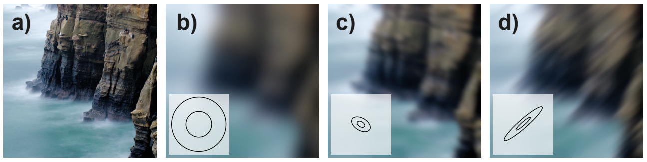

Fig. 2. An original 2D signal f (*a*) alongside samples from its Gaussian scale space f_Σ (*b*-*d*). In *b*), isotropic smoothing is applied, while *c*) and *d*) demonstrate examples of anisotropic filtering. Insets show isolines of the corresponding Gaussian kernels. We consider scale spaces that are continuous both in signal coordinates and in full Gaussian covariance.

computing, e.g., RGB images using $d_i = 2$, $d_o = 3$, or signed distance functions (SDFs) for geometry using $d_i = 3$, $d_o = 1$. Ultimately, we are interested in fitting a neural field $F \in \mathbb{R}^{d_i} \to \mathbb{R}^{d_o}$ to (versions of) f. We write continuous coordinates of signals and fields as $\mathbf{x} \in \mathbb{R}^{d_i}$.

Gaussian scale space. The linear (Gaussian) scale space [Iijima 1959; Koenderink 1984; Lindeberg 2013; Witkin 1987] of f is defined as the continuous convolution of f with a Gaussian kernel with positive definite covariance matrix $\Sigma \in \mathbb{R}^{d_i \times d_i}$:

$$f_\Sigma(\mathbf{x}) \coloneqq \frac{1}{\sqrt{(2\pi)^{d_i} \det(\Sigma)}} \int_{\mathbb{R}^{d_i}} f(\mathbf{x} - \boldsymbol{\tau}) \exp\left(-\frac{1}{2}\boldsymbol{\tau}^T \Sigma^{-1} \boldsymbol{\tau}\right) \mathrm{d}\boldsymbol{\tau}. \quad (1)$$

In such an augmented signal representation, continuously varying Σ gives rise to differently smoothed versions of the signal (Fig. 2). A very practical benefit of this structure is that it provides precise control over the frequency content of a signal. Yet, obtaining a scale-space representation is costly, as it requires executing a continuous d_i-dimensional integral for each combination of $\mathbf{x}$ and Σ – an operation for which, in general, no closed-form solution exists.

Fourier features. Neural networks exhibit an intrinsic spectral bias towards "simple" solutions [Rahaman et al. 2019], which makes it challenging for basic fully-connected architectures to learn high-frequency content. An established remedy is to first featurize the input coordinate $\mathbf{x}$ using a fixed mapping $\gamma \in \mathbb{R}^{d_i} \to \mathbb{R}^{2m}$, based on sinusoids of m different frequencies [Mildenhall et al. 2020; Rahimi and Recht 2007; Tancik et al. 2020]:

$$\gamma(\mathbf{x}) = \begin{pmatrix} \lambda_1 \cos\left(2\pi \mathbf{a}_1^T \mathbf{x}\right) \\ \lambda_1 \sin\left(2\pi \mathbf{a}_1^T \mathbf{x}\right) \\ \vdots \\ \lambda_m \cos\left(2\pi \mathbf{a}_m^T \mathbf{x}\right) \\ \lambda_m \sin\left(2\pi \mathbf{a}_m^T \mathbf{x}\right) \end{pmatrix}. \quad (2)$$

In this positional encoding, $\mathbf{a}_i \in \mathbb{R}^{d_i}$ are frequency vectors and $\lambda_i \in \mathbb{R}$ are weights of the corresponding Fourier feature dimensions. Small offsets in $\mathbf{x}$ lead to rapid changes of $\gamma(\mathbf{x})$ for high frequencies $\mathbf{a}_i$. Therefore, feeding $\gamma(\mathbf{x})$ instead of the raw $\mathbf{x}$ into a neural network effectively lifts the burden of creating high frequencies from the network, resulting in higher-quality fits of complex signals.

Lipschitz continuity. A Lipschitz-continuous function is limited in how fast it can change. Formally, for this class of functions, there exists a Lipschitz bound $c \geq 0$ such that

$$\|f(\mathbf{x}_1) - f(\mathbf{x}_2)\|_p \leq c\|\mathbf{x}_1 - \mathbf{x}_2\|_p \quad (3)$$

for all possible $\mathbf{x}_1$ and $\mathbf{x}_2$ and an arbitrary choice of p. Intuitively, moving a certain distance in the function's domain is guaranteed to result in a bounded change of function values.

If f is implemented using a fully-connected network (MLP) with l layers and 1-Lipschitz activation functions (e.g., ReLU), an upper Lipschitz bound is given by [Gouk et al. 2021]

$$c = \prod_{k=1}^{l} \|W_k\|_p, \quad (4)$$

where W_k is the (trainable) weight matrix of the k'th network layer. Enforcing bounded weight-matrix norms effectively imposes a *fixed*, *global* constraint on how rapidly f can change.

4 METHOD

We seek to learn a neural field $F(\mathbf{x}, \Sigma)$ that captures the full anisotropic Gaussian scale space $f_\Sigma(\mathbf{x})$ of a signal f, i.e., a family of Gaussian-smoothed signals with arbitray, anisotropic covariance Σ. We consider, both, coordinates $\mathbf{x}$ and covariance matrix Σ, continuous parameters, so that the field can be queried at *any* location using *any* Gaussian filter. Since computing f_Σ via Eq. 1 is intractable for all but the simplest f, we learn $F(\mathbf{x}, \Sigma)$ self-supervised, i.e., we only rely on the *original* signal f.

To achieve this goal, we make a simple but far-reaching observation: Careful dampening of high-frequency Fourier features produces a low-pass filtered signal of high quality if the neural network representing the signal is Lipschitz-bounded. Based on this observation, we develop a novel paradigm that leverages the combined properties of modulated Fourier features and Lipschitz-continuous networks (Sec. 4.1). Our approach relies on a neural architecture with carefully designed constraints (Sec. 4.2), such that training can be performed using supervision from raw, unfiltered signal samples (Sec. 4.3). The emerging continuous filter parameters are uncalibrated, since we employ an efficient method that does not explicitly execute any Gaussian smoothing during training. Therefore, after training, we perform a lightweight calibration to enable precise filtering (Sec. 4.4).

4.1 Self-supervised Learning of Gaussian-smoothed Neural Fields

We consider an established neural architecture that consists of the composition of a positional encoding using Fourier features γ (Eq. 2) with a multi-layer perceptron (MLP) Ψ_θ:

$$F(\mathbf{x}) = \Psi_\theta\left(\gamma(\mathbf{x})\right). \quad (5)$$

Here, θ represents the trainable network parameters, consisting of weight matrices W_k and bias vectors $\mathbf{b}_k$. Based on this setup, our approach fuses two techniques that are well-known in isolation, but, to the best of our knowledge, have not yet been systematically considered in combination: First, we employ Fourier feature modulation, i.e., we dampen high-frequency components of the positional encoding in Eq. 2 using custom, frequency-dependent

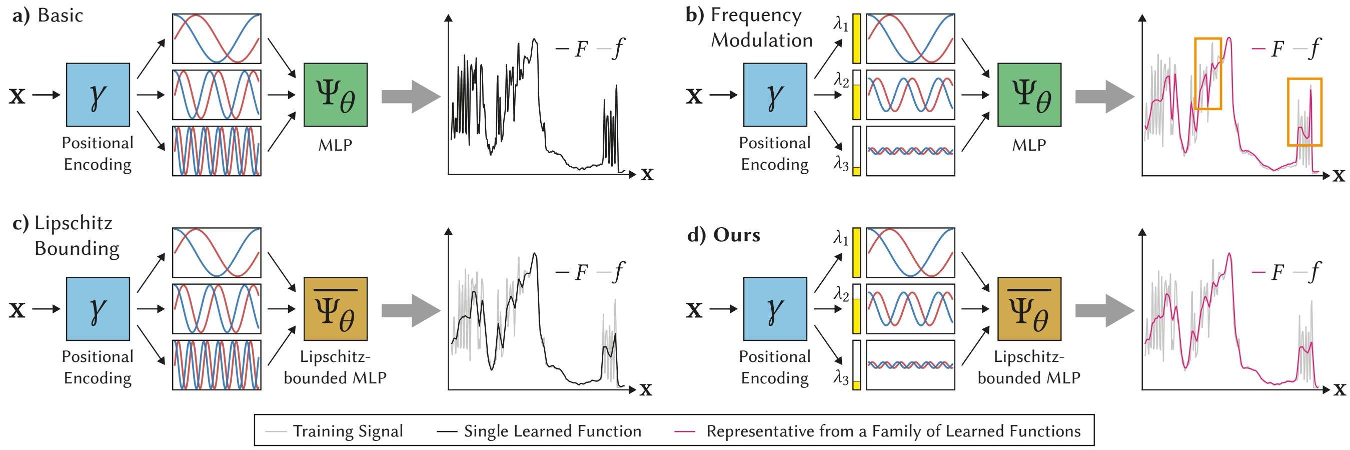

Fig. 3. Four different strategies to learn a neural field F from a signal f. F takes a continuous coordinate $\mathbf{x}$ as input, which is fed into a positional encoding γ (Eq. 2) that produces a set of Fourier features using cosine (blue curves) and sine (red curves) functions of different frequencies. The resulting features serve as input to an MLP Ψ_θ that regresses f. (*a*) The basic setup learns a faithful reconstruction of f (the curves for F and f overlay completely), but does not allow any smoothing. (*b*) Modulating the Fourier features using custom weights λ_i (yellow bars) tends to remove some high frequencies, but distorts the reconstruction in an unpredictable way (orange rectangles mark incoherent spikes in F). (*c*) Employing a Lipschitz-bounded MLP $\overline{\Psi_\theta}$ leads to smoothing, but it requires choosing a single fixed bound for training, lacking flexibility. (*d*) Our approach combines Fourier feature modulation with Lipschitz bounding to enable controllable smoothing.

weights λ_i [Barron et al. 2021; Hertz et al. 2021; Lin et al. 2021; Park et al. 2021]. Second, we enforce an upper Lipschitz bound of Ψ_θ [Gouk et al. 2021; Miyato et al. 2018; Szegedy et al. 2013]. We refer to such a bounded network as $\overline{\Psi_\theta}$. To understand how this construction can help learn a controllably smooth function from a raw signal, consider four different strategies for learning a signal in Fig. 3.

In Fig. 3a, we illustrate the basic setup of Eq. 5 without any modifications, i.e., with $\lambda_i = 1\ \forall i$ and an unbounded MLP Ψ_θ. Unsurprisingly, we observe that training F on f results in a faithful fit. Yet, we do not have any handle for creating smooth network responses here.

As a potential remedy, consider the setup in Fig. 3b, where we dampen the higher-frequency Fourier features of γ. Consistent with established findings in the literature [Mildenhall et al. 2020; Müller et al. 2022; Tancik et al. 2020], we observe that fitting quality degrades. Yet, this happens in an unpredictable way, leading to incoherent high-frequency spikes in F. This is because Ψ_θ – depending on factors such as signal complexity and network capacity – can compensate for missing input frequencies by forging a function with high gradients w.r.t. its inputs. Fourier feature modulation can help learn signals robustly when used progressively [Hertz et al. 2021; Lin et al. 2021], or facilitate learning of a multiscale representation when supervision across scales is available [Barron et al. 2021, 2022]. Yet, on its own, it is not a viable strategy for learning a smooth function from a raw supervision signal.

We now turn to an architecture with a Lipschitz-bounded MLP $\overline{\Psi_\theta}$, yet with unmodified Fourier features, depicted in Fig. 3c. We observe that the trained F now indeed captures a smoother version of f. We seem to have achieved our goal; however, the Lipschitz bound c needs to be fixed for training and is baked into the MLP. While this is a useful property for robust training [Cisse et al. 2017; Hein and Andriushchenko 2017] or smooth interpolation [Liu et al. 2022], this strategy does not provide any control over the smoothing once the network is trained.

The above considerations motivate us to develop a new approach that combines frequency modulation with Lipschitz bounding, as shown in Fig. 3d. When dampening high-frequency Fourier features in this setup, the Lipschitz-bounded $\overline{\Psi_\theta}$ cannot compensate for the missing frequency content, since to turn the now low-frequency encoding into a high-frequency output, it would need to produce large gradient magnitudes w.r.t. the positional encoding. Instead, *it is forced to learn an F that matches the raw f as closely as possible given frequency and gradient constraints.* This form of "parameterized gradient limiting" through modulated Fourier features facilitates controllable smoothing (dashed colored lines Fig. 4).

While we intuitively expect *some* form of smoothing, the exact reconstruction qualities emerging from our strategy are not obvious. However, examining reconstructions on a broad variety of real-world signals and modalities, we make a surprising, yet crucial empirical observation: *The emerging smoothing is a remarkably faithful approximation of Gaussian filtering.* We extensively validate this claim in Sec. 5, but consider a rigorous theoretical justification beyond the scope of this work. In Fig. 4, we visually compare results F obtained from our approach against the best-fitting Gaussian-smoothed versions f_Σ of f (solid grey curves).

The insights developed above suggest an effective and efficient procedure for learning a Gaussian scale space in a self-supervised fashion: First, we build an architecture following Eq. 5 with carefully sampled Fourier frequency vectors and a robustly Lipschitz-bounded neural network (Sec. 4.2). Second, we train the architecture with strategically dampened Fourier features using the original signal f

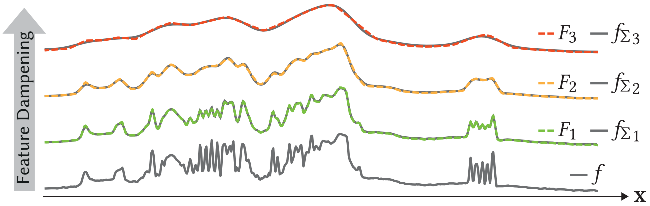

Fig. 4. Given a training signal f (bottom row), progressive dampening of Fourier features in combination with a Lipschitz-bounded MLP allows a neural field F learn Gaussian-smoothed versions f_Σ of f. In the three upper rows, differently smoothed F_i (dashed colored curves) and their respective closest f_{Σ_i} (solid grey curves) are overlayed, revealing that our solution provides a faithful approximation of Gaussian filtering.

for supervision to learn an entire continuous anisotropic Gaussian scale space (Sec. 4.3). Finally, we map dampening weights λ_i to Gaussian covariance Σ to enable precise filtering (Sec. 4.4).

4.2 Architecture

Following the reasoning developed in the previous section, we design our neural field as

$$F(\mathbf{x}, \hat{\Sigma}) = \overline{\Psi_\theta}\left(\gamma(\mathbf{x}, \hat{\Sigma})\right), \tag{6}$$

which implements two modifications to the basic setup of Eq. 5. First, we extend the positional encoding γ to incorporate a pseudo-covariance matrix $\hat{\Sigma}$ as an additional parameter, as detailed in Sec. 4.2.1. Second, we use a Lipschitz-bounded MLP $\overline{\Psi_\theta}$, the construction of which is explained in Sec. 4.2.2.

We emphasize that our formulation in Eq. 6 naturally supports spatially varying filtering, as $\mathbf{x}$ and $\hat{\Sigma}$ are independent inputs to the field.

4.2.1 Fourier Features for Filtering. We are concerned with designing a variant of the positional encoding in Eq. 2 that facilitates high-quality filtering through feature dampening.

We observe that the distribution of frequencies $\mathbf{a}_i$ plays a crucial role in the process. Several strategies have been explored in the literature on neural field design. A popular approach relies on axis-aligned frequencies [Barron et al. 2021, 2022; Mildenhall et al. 2020] (Fig. 5a), but the lack of angular coverage does not allow arbitrary anisotropies. Tancik et al. [2020] propose to distribute frequencies following a normal distribution (Fig. 5b). This leads to denser coverage, but the uncorrelated samples introduce clusters and holes. We find this uneven coverage problematic for highly selective dampening and opt for a strategy that involves stratification [Niederreiter 1992]. Specifically, we use a Sobol [1967] sequence and map it to the hyperball using the method of Griepentrog et al. [2008] (Fig. 5c). We then radially warp the samples such that radially averaged sample density follows a zero-mean Gaussian distribution with variance $\sigma_\mathbf{a}^2$ (Fig. 5d). We find this shifting of sample budget towards the low frequencies a good trade-off between high-quality filtering with small-scale and large-scale kernels.

Our positional encoding needs to support anisotropically modulated Fourier features. To this end, we use a positive semi-definite pseudo-covariance matrix $\hat{\Sigma} \in \mathbb{R}^{d_i \times d_i}$ to obtain frequency-dependent dampening factors λ_i of the individual components in Eq. 2 (Fig. 5e):

$$\lambda_i(\hat{\Sigma}) = \exp\left(-\sqrt{\mathbf{a}_i^T \hat{\Sigma} \mathbf{a}_i}\right). \tag{7}$$

Notice how $\hat{\Sigma}$ is used without inversion here, in contrast to Σ in Eq. 1. This is a direct consequence of the reciprocal relationship of covariance in the primal and the Fourier domain [Brigham 1988]. Consider filtering a 2D signal with stronger horizontal than vertical smoothing. The convolution in the primal domain requires a kernel with higher variance in the horizontal direction, while the corresponding multiplication in the Fourier domain needs to dampen horizontal frequencies more strongly, leading to vertically elongated covariance.

Different from very similar existing techniques for supervised anti-aliasing based on axis-aligned frequencies [Barron et al. 2021, 2022], our dampening operates on Fourier frequencies that occupy the entire d_i-dimensional space, enabling arbitrary anisotropic filtering. In Sec. 4.4, we describe how to obtain filtering results with Gaussian covariance Σ from pseudo-covariance $\hat{\Sigma}$.

4.2.2 Robust Lipschitz Bounding. We require the MLP $\overline{\Psi_\theta}$ in Eq. 6 to be Lipschitz-bounded. Eq. 3 gives us the freedom to use any p-norm, but we choose $p = 2$, because only this choice retains spatially invariant bounding after adding the positional encoding. To understand this connection, consider a setting with a one-dimensional input coordinate $\mathbf{x}$. Now consider a coordinate pair $(\mathbf{x}_1, \mathbf{x}_2)$ and a shifted version of it $(\mathbf{x}_3, \mathbf{x}_4)$ (Fig. 6, top). If we applied $\overline{\Psi_\theta}$ directly to these coordinates, the right-hand side of Eq. 3 tells us that because $\mathbf{x}_1$ and $\mathbf{x}_2$ have the same distance as $\mathbf{x}_3$ and $\mathbf{x}_4$, a fixed Lipschitz bound c results in the same smoothing, i.e., the bounding is spatially invariant. However, our situation is different: $\overline{\Psi_\theta}$ does not operate on raw coordinates, but on their positional encoding $\gamma(\mathbf{x})$, where each $\mathbf{x}$ is mapped to a location on a circle (Fig. 6, bottom). Only if equidistant coordinates remain equidistant after positional encoding, formally

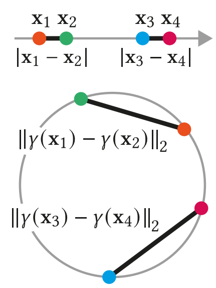

Fig. 6. Distances (black lines) before (top) and after (bottom) positional encoding.

$$\|\gamma(\mathbf{x}_1) - \gamma(\mathbf{x}_2)\|_p \overset{!}{=} \|\gamma(\mathbf{x}_3) - \gamma(\mathbf{x}_4)\|_p, \tag{8}$$

the property that a specific c has the same effect across the entire domain is maintained. Eq. 8 is only fulfilled by the isotropic 2-norm.

Following Eq. 4, choosing $p = 2$ translates into an MLP $\overline{\Psi_\theta}$ whose weight matrices have bounded *spectral* norms $\|W_k\|_2$. We pick a Lipschitz bound of $c = 1$, which can be satisfied by individually constraining the spectral norm of each weight matrix to at most 1.

We parameterize each (arbitrarily-shaped) weight matrix using the singular value decomposition (SVD) $W_k = U_k S_k V_k^T$, where U_k and V_k are orthogonal matrices, and S_k is a diagonal matrix containing the non-negative singular values of W_k. Using this decomposition, $\|W_k\|_2 \leq 1$ can be achieved by constraining the trainable parameters on the diagonal of S_k using a sigmoid function. To parameterize U_k and V_k, we capitalize on the fact that the matrix

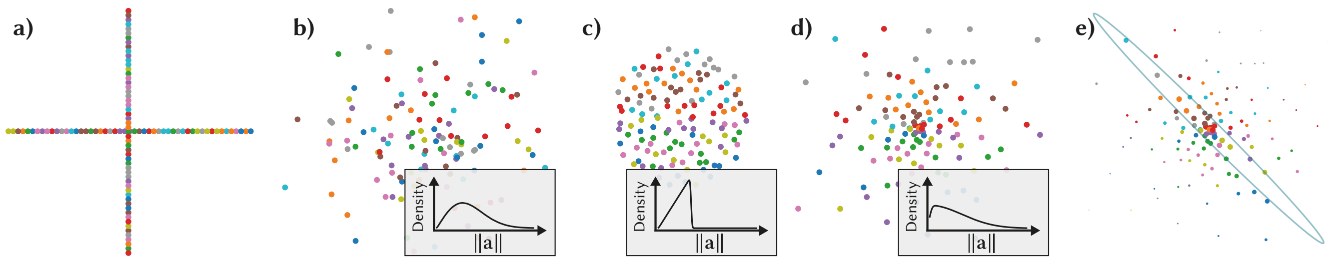

Fig. 5. Distribution of Fourier frequencies $\mathbf{a}$ (colored dots). Axis-aligned frequencies (*a*) cannot capture anisotropies. Uncorrelated sampling from a Gaussian distribution (*b*) leads to clusters and holes, impeding filtering quality. Our approach starts with a low-discrepancy sequence (*c*) and warps samples radially such that radial density follows a zero-mean Gaussian distribution (*d*). The grey insets in *b*)-*d*) show the radial density distributions of the respective point sets. In *e*), we visualize an example of dampening the frequencies in *d*) with a matrix $\hat{\Sigma}$ (an isoline of its inverse is shown). Here, point size corresponds to dampening factors $\lambda_i(\hat{\Sigma})$. Our carefully distributed Fourier frequencies facilitate highly selective anisotropic filtering.

exponential of a skew-symmetric matrix results in an orthogonal matrix [Hyland and Rätsch 2017; Lezcano-Casado and Martınez-Rubio 2019]. Concretely, for each U_k and V_k, we arrange a suitable number of trainable parameters into skew symmetric matrices $A_{U,k} = -A_{U,k}^T$ and $A_{V,k} = -A_{V,k}^T$ and compute

$$U_k = \exp(A_{U,k}) \quad \text{and} \quad V_k = \exp(A_{V,k}). \tag{9}$$

Using this parameterization of weight matrices (Fig. 7), all trainable parameters of $\overline{\Psi_\theta}$ can be optimized robustly and in an unconstrained fashion, while always resulting in a Lipschitz bound $c \leq 1$.

$$W_k = \underbrace{\exp\left[\overbrace{\begin{pmatrix} 0 & \theta_1 & \theta_2 \\ -\theta_1 & 0 & \theta_3 \\ -\theta_2 & -\theta_3 & 0 \end{pmatrix}}^{A_{U,k}}\right]}_{U_k} \underbrace{\begin{pmatrix} \mathrm{sig}(\theta_4) & 0 & 0 \\ 0 & \mathrm{sig}(\theta_5) & 0 \\ 0 & 0 & \mathrm{sig}(\theta_6) \end{pmatrix}}_{S_k} \underbrace{\exp\left[\overbrace{\begin{pmatrix} 0 & \theta_7 & \theta_8 \\ -\theta_7 & 0 & \theta_9 \\ -\theta_8 & -\theta_9 & 0 \end{pmatrix}}^{A_{V,k}}\right]^T}_{V_k}$$

Fig. 7. Our parameterization for a Lipschitz-bounded $W_k \in \mathbb{R}^{3\times 3}$, containing nine trainable parameters $\{\theta_1, \ldots, \theta_9\}$ that can be freely optimized.

Special care has to be taken in the case of $d_o > 1$, i.e., when $\overline{\Psi_\theta}$ has multiple output channels. Recall that the definition of Lipschitz continuity in Eq. 3 relies on a norm of differences between function values. One undesired way to more easily satisfy Eq. 3 for $p = 2$ is to lower the value of the norm on the left-hand side by producing similar function values across output channels. This is because, in expectation, Euclidean distances of point pairs on the main diagonal of $\mathbb{R}^{d_o}$ are shorter than distances of point pairs in full d_o-dimensional space. In practice, we observe that, without accounting for this effect, our models tend to decrease variance between output channels, e.g., they produce washed-out colors in RGB images. Fortunately, there is a simple remedy for this problem: We treat the rows of the last weight matrix W_l in $\overline{\Psi_\theta}$ individually. Specifically, we replace the SVD-based parameterization of W_l by a simple row-wise ℓ_2-normalization. This straightforward modification eliminates all undesired cross-channel contamination.

4.3 Training

We train our neural Gaussian scale-space fields using the loss

$$\mathcal{L} = \mathbb{E}_{\mathbf{x},\hat{\Sigma}} \left\| F(\mathbf{x}, \hat{\Sigma}) - f(\mathbf{x}) \right\|_2^2 . \tag{10}$$

It merely requires stochastically sampling coordinates $\mathbf{x}$ and pseudo-covariances $\hat{\Sigma}$ to produce a filtered network output and comparing it against *original* signal samples $f(\mathbf{x})$. We emphasize that our training does not require f_Σ, thereby completely avoiding costly manual filtering per Eq. 1 or any approximation thereof.

As all $\hat{\Sigma}$ need to be positive semi-definite, we sample them using the eigendecomposition $\hat{\Sigma} = Q\Lambda Q^T$, where $Q, \Lambda \in \mathbb{R}^{d_i \times d_i}$. Specifically, we uniformly sample an orthonormal set of eigenvectors Q, and log-uniformly sample corresponding non-negative eigenvalues, arranged into the diagonal matrix Λ.

The network parameters θ are optimized using Adam [Kingma and Ba 2015] with default parameters.

4.4 Variance Calibration

Once trained, our neural field $F(\mathbf{x}, \hat{\Sigma})$ in Eq. 6 captures a scale space, where the degree of smoothing is steered by modulation of Fourier features via pseudo-covariance $\hat{\Sigma}$. However, there is no guarantee at all that a particular choice of $\hat{\Sigma}$ results in a Gaussian-filtered function with covariance $\Sigma = \hat{\Sigma}$.

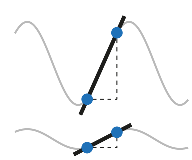

Fig. 8. Lipschitz bounds (see text).

Importantly, we find that the relationship between Σ and $\hat{\Sigma}$ depends on the signal f itself. Consider two sine waves of the same frequency but with different amplitudes (Fig. 8). Gaussian filtering of these signals per Eq. 1 gives two identical smoothed signals up to the original difference in amplitude. Our approach does not exhibit this kind of invariance. The original low-amplitude signal has a lower Lipschitz bound (slope of the black lines in Fig. 8) and, thus, requires more aggressive bounding to achieve the same degree of smoothing as the high-amplitude signal. Therefore, the same $\hat{\Sigma}$ in Eq. 7 will have different effects when learning scale spaces of the two waves.

Since our ultimate goal is to produce filtering results with control over covariance that is as precise as possible, we seek to find a signal-specific calibration function $h(\Sigma) = \hat{\Sigma}$ that allows us to obtain our final Gaussian scale-space field:

$$F(\mathbf{x}, \Sigma) = \overline{\Psi_\theta} \left(\gamma(\mathbf{x}, h(\Sigma)) \right) . \tag{11}$$

Notice that h is injected into the pipeline *after* training.

Calibration. We design a lightweight calibration scheme for determining h that is applicable to any signal modality. First, we empirically observe that the discrepancy between Σ and $\hat{\Sigma}$ stems from a difference in *isotropic* scale, while anisotropies are captured faithfully. Consequently, our calibration only considers matrices of the form $\Sigma = \sigma^2\mathbf{I}$ and $\hat{\Sigma} = \hat{\sigma}^2\mathbf{I}$, where $\mathbf{I}$ is the $d_i \times d_i$ identity matrix, and $\sigma^2, \hat{\sigma}^2 \in \mathbb{R}_{\geq 0}$ are variance and pseudo-variance, respectively.

We rely on computing a small number of Monte Carlo estimates of Gaussian smoothing that serve as ground truth and can be matched against our trained field. Specifically, we consider a set of $n_{\mathbf{x}} = 64$ random pilot coordinates $\mathbf{x}_i$, and a set of $n_{\sigma^2} = 16$ log-uniformly spaced variances σ_j^2. For each combination of $\mathbf{x}_i$ and σ_j^2, we compute a Monte Carlo estimate of Gaussian smoothing based on $N = 2000$ samples from $F(\mathbf{x}, \mathbf{0})$, i.e., our trained field without any feature dampening (Fig. 9a):

$$F_{i,j} = \frac{1}{N} \sum_{\boldsymbol{\tau} \sim \mathcal{N}(\mathbf{0}, \sigma_j^2\mathbf{I})} F(\mathbf{x}_i - \boldsymbol{\tau}, \mathbf{0}). \tag{12}$$

In addition, we consider a set of $n_{\hat{\sigma}^2} = 256$ log-uniformly spaced pseudo-variances $\hat{\sigma}_k^2$ and compute, for each $\mathbf{x}_i$ (Fig. 9b),

$$\hat{F}_{i,k} = F(\mathbf{x}_i, \hat{\sigma}_k^2\mathbf{I}). \tag{13}$$

For each variance σ_j^2, we now find the pseudo-variance $\hat{\sigma}_{k_j}^2$ that results in the lowest error across pilot coordinates $\mathbf{x}_i$:

$$k_j = \arg\min_k \sum_{i=1}^{n_{\mathbf{x}}} \|F_{i,j} - \hat{F}_{i,k}\|_2^2. \tag{14}$$

Our final task is to regress the transformation h that maps variances σ_j^2 to their corresponding pseudo-variances $\hat{\sigma}_{k_j}^2$. We observe a strong linear relationship (Fig. 9c), so we choose $\hat{\sigma}^2 = h(\sigma^2) = \mu\sigma^2$, where $\mu \in \mathbb{R}$. Taking the logarithmic spacing of our samples into account, we regress

$$\mu = \left(\prod_{j=1}^{n_{\sigma^2}} \frac{\hat{\sigma}_{k_j}^2}{\sigma_j^2} \right)^{\frac{1}{n_{\sigma^2}}}. \tag{15}$$

Application to the full-covariance setting gives our final calibration:

$$\hat{\Sigma} = h(\Sigma) = \mu\Sigma. \tag{16}$$

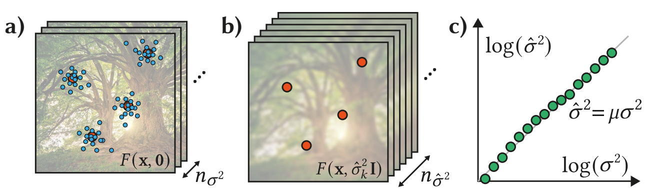

Fig. 9. Our variance calibration is based on pilot coordinates $\mathbf{x}_i$ (red points). (*a*) We use Monte Carlo samples (blue points) to estimate ground-truth smoothing based on our field when no feature dampening is applied. (*b*) We compute a sequence of differently smoothed network responses to be matched against the ground-truth from *a*). (*c*) The obtained variance–pseudo-variance pairs (green points) exhibit a linear relationship.

Discussion. The one-time estimations in Eq. 12 and Eq. 13 require a total of $n_{\mathbf{x}} \times (n_{\sigma^2} \times N + n_{\hat{\sigma}^2}) \approx 2M$ forward passes through our trained network, the total computation of which is instantaneous. Therefore, the entire calibration procedure imposes negligible cost compared to network training.

5 EVALUATION

We demonstrate the performance of our approach on different modalities (Sec. 5.1) and applications (Sec. 5.2), before analyzing individual components of our pipeline (Sec. 5.3). Our source code and supplementary materials are available on our project page at https://neural-gaussian-scale-space-fields.mpi-inf.mpg.de.

Implementation Details. All our signals are scaled to cover the unit domain $[-1, 1]^{d_i}$. Our positional encoding γ uses 1024 Fourier features ($m = 512$). Networks $\overline{\Psi_\theta}$ consist of four layers with 1024 features each, and are trained with a learning rate of 5e-4 (1e-4 for light stage data) until convergence. The variances for radial Fourier feature warping are $\sigma_{\mathbf{a}}^2 = 2000$ for images, $\sigma_{\mathbf{a}}^2 = 100$ for SDFs, $\sigma_{\mathbf{a}}^2 = 500$ for light stage data, and $\sigma_{\mathbf{a}}^2 = 50$ for optimization. Eigenvalues for $\hat{\Sigma}$ during training are log-uniformly sampled in $[10^{-12}, 10^2]$. We have implemented our method in PyTorch [Paszke et al. 2017].

Baselines. We quantitatively and qualitatively compare our filtering results against several baselines, while a converged Monte Carlo estimate of Eq. 1 serves as ground truth.

BACON [Lindell et al. 2022], MINER [Saragadam et al. 2022], and PNF [Yang et al. 2022] learn neural multiscale representations, where intermediate network outputs constitute a *discrete* set of low-pass filtered versions of the original signal. Following Nsampi et al. [2023], we linearly combine these intermediate outputs using coefficients that we optimize per signal to best match the filtered ground truth. Only PNF supports anisotropic filtering using a discretization of orientation. MINER requires prefiltered input during training.

We further consider INSP [Xu et al. 2022], which performs signal processing of a trained neural field via a dedicated filtering network. Each filter kernel requires training a separate filtering network, while our approach supports a continuous family of filter kernels.

Finally, we compare against NFC [Nsampi et al. 2023], which allows filtering based on a learned integral field that needs to be queried hundreds of times per output coordinate. While this method supports continuous axis-aligned scaling of filter kernels, general anisotropic kernels require individual optimizations, leading to a discretization of kernels in the anisotropic setting. In contrast, our approach handles arbitrary anisotropic Gaussian kernels and produces filtered results using a single forward pass. We obtain best results for NFC when using piecewise linear models for 2D isotropic filtering, and piecewise constant models in all other cases.

All methods differ in their (implicit) treatment of signal boundaries. To facilitate a meaningful quantitative comparison, we crop all results such that the boundary does not influence the evaluation. Qualitative results always show uncropped signals.

Regardless of whether we evaluate isotropic or anisotropic filtering capabilities, our scale-space fields are always trained using the complete anisotropic pipeline as described in Sec. 4.

Table 1. **Image** quality of filtering with different **isotropic** kernels (columns) for different methods (rows). "$\mathbf{x}$-cont." and "σ^2-cont." indicate, whether the method is continuous in the spatial and the kernel domain, respectively. Bold and underlined numbers denote the best and second-best method, respectively.

Method	$\mathbf{x}$-cont.	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	✓	×	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	✓	×	**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	✓	×	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	✓	✓	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**
Ours	✓	✓	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

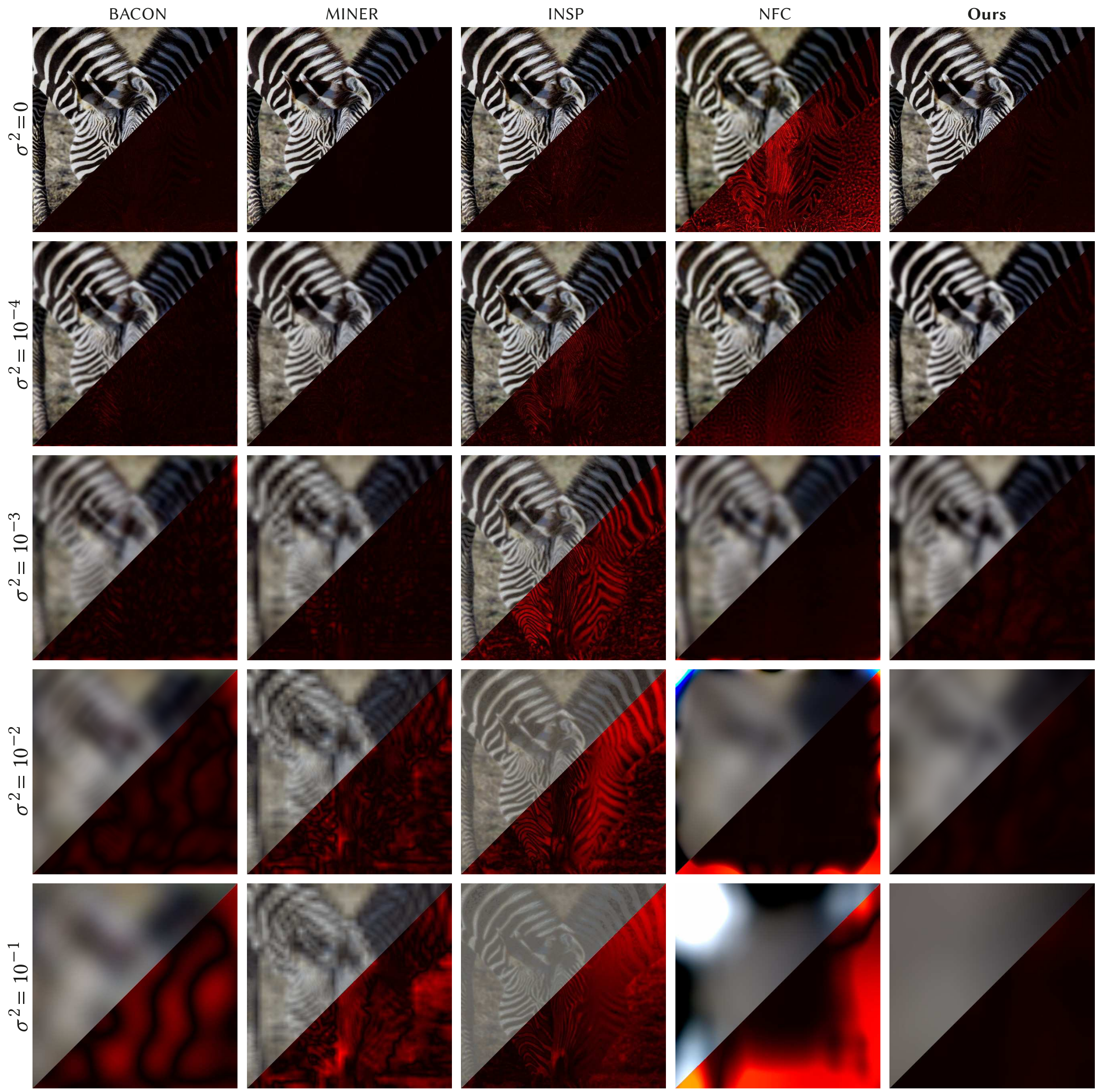

Fig. 10. Qualitative results for isotropic image filtering. We show results (upper left triangles) next to error visualizations (lower right triangles). Our supplementary materials contain more visual comparisons.

Table 2. **Image** quality for methods that support **anisotropic** kernels. Refer to to the caption Tab. 1 for details on individual columns and highlighting.

	$\mathbf{x}$-cont.	Σ-cont.	PSNR↑	LPIPS↓	SSIM↑
PNF	✓	×	24.15	0.571	0.704
NFC	✓	×	<u>30.31</u>	<u>0.094</u>	<u>0.857</u>
Ours	✓	✓	**34.82**	**0.069**	**0.940**

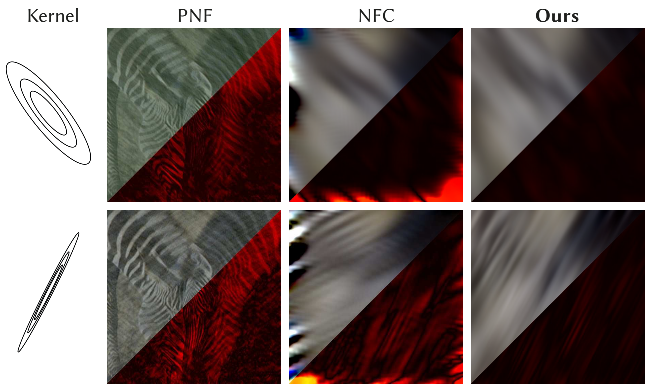

Fig. 11. Qualitative results for anisotropic image filtering. Our supplementary materials contain more visual comparisons.

5.1 Modalities

We extensively evaluate our method on two signal modalities relevant for visual computing: images and signed distance fields.

5.1.1 Images. We consider a corpus of 100 RGB images ($d_i = 2$, $d_o = 3$) at a resolution of 2048×2048 pixels, randomly selected from the Adobe FiveK dataset [Bychkovsky et al. 2011] and treated as continuous signals using bilinear interpolation. In a first step, we investigate *isotropic* filtering on a set of five Gaussian kernels with variances $\sigma^2 \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, where the first configuration measures fitting quality of the original signal without any filtering. In Tab. 1, we evaluate results using the image quality metrics PSNR, LPIPS [Zhang et al. 2018a], and SSIM [Wang et al. 2004]. Fig. 10 provides a corresponding qualitative comparison. We see that our approach is highly competitive across all filter sizes. While NFC provides the best results across all filter kernels of significant size, it introduces severe boundary artifacts, the effect of which we purposefully exclude in our numerical evaluations.

To evaluate performance for *anisotropic* filtering, we sample 100 full covariance matrices Σ using the scheme described in Sec. 4.3, where each Σ is evaluated on all 100 test images. We report corresponding results in Tab. 2 and Fig. 11 and observe that our approach outperforms all baselines on this task.

In Fig. 12, we show spatially varying filtering for foveated rendering. Here, the size of the filter kernel is modulated by the distance to a fixation point in the image. Our approach naturally supports such spatially varying kernels, since evaluation location $\mathbf{x}$ and filter covariance matrix Σ are independent inputs to our model.

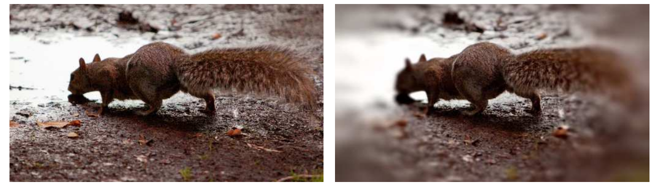

Fig. 12. An original image (left) and our foveated result (right).

5.1.2 Signed Distance Fields. Encoding surfaces as the zero-level-set of an SDF ($d_i = 3$, $d_o = 1$) is a popular way to represent geometry [Park et al. 2019]. Our evaluation is based on four 3D models, following a similar protocol as for images. In Tab. 3 and Tab. 4, we list quantitative evaluations for isotropic and anisotropic filtering, respectively, using MSE and intersection over union (IoU) across the SDF, as well as the Chamfer distance of the reconstructed surfaces. Fig. 13 and Fig. 14 show corresponding qualitative results. While results appear mostly inconclusive for the isotropic case, we outperform the only other baseline that can handle anisotropic filtering in this domain – NFC – by a large margin.

5.2 Applications

Here, we present three applications that utilize the properties of neural Gaussian scale-space fields. First, we demonstrate anti-aliasing with texture fields, before filtering a 4D light-stage capture and showing a proof-of-concept application in the domain of continuous multiscale optimization.

5.2.1 Texture Anti-aliasing. Texturing a 3D mesh is a fundamental building block in many rendering and reconstruction pipelines. It requires re-sampling of a texture into screen space, which must account for spatially-varying, anisotropic minification and magnification to avoid aliasing [Heckbert 1986]. Our method enables this functionality for textures that are represented as continuous neural fields.

In Fig. 15, we show a result using a scale-space field for texturing an object. We first learn the scale space of the texture in *uv*-coordinates, and determine the optimal anisotropic Gaussian kernel for a given camera view that results in alias-free re-sampling [Heckbert 1989]. We see that our approach is successful in removing aliasing artifacts from the rendering. The supplementary materials contain a video that demonstrates view-coherent texturing.

5.2.2 Light Stage. A light stage allows to capture an object using different controlled illumination conditions. A structured capture, e.g., using one light at a time, thus enables high-quality relighting using arbitrary environment maps in a post-process.

We apply our method to the 4D product space of 2D pixel coordinates and 2D spherical light directions. As demonstrated in Fig. 16, sampling a single light direction from our model at the finest scale produces hard shadows, while moving to coarser scales in light direction introduces soft shadows.

5.2.3 Multiscale Optimization. As a final application, we show that our scale-space fields can be used for continuous multiscale optimization. In our proof-of-concept setup, we assume that we have

Table 3. Quality of filtered **SDFs** with different **isotropic** kernels. Refer to to the caption Tab. 1 for details on individual columns and highlighting.

Method	**x**-cont.	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	✓	×	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	✓	×	**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	✓	×	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	✓	✓	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**
Ours	✓	✓	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

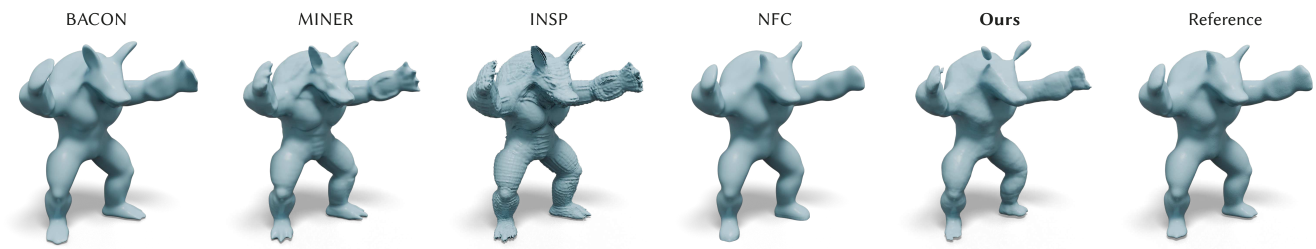

Fig. 13. Qualitative results for isotropic SDF filtering on a kernel with $\sigma^2 = 10^{-3}$. Our supplementary materials contain more visual comparisons.

Table 4. **SDF** quality for methods that support **anisotropic** kernels. Refer to the caption of Tab. 1 for details on individual columns and highlighting.

	x-cont.	Σ-cont.	MSE↓	Cham.↓	IoU↑
NFC	✓	×	7.1e-2	4.6e-1	0.08
Ours	✓	✓	**2.8e-3**	**1.2e-1**	**0.42**

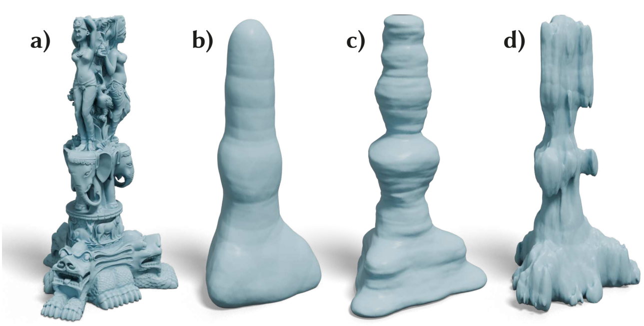

Fig. 14. (*a*) An original SDF. (*b*) Our smoothing of *a*) with an isotropic kernel, suppressing detail. (*c*) Our smoothing of *a*) with an anisotropic kernel, where the strength of the smoothing is weaker in the vertical direction, but strong in all other directions, allowing anisotropic structure suppression. (*d*) Strong smoothing is only applied in the vertical direction.

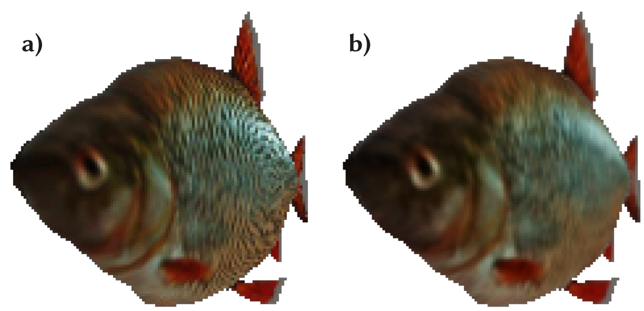

Fig. 15. (*a*) Rendering a textured mesh is prone to aliasing artifacts. (*b*) Our method learns the continuous scale space of a texture and allows spatially varying, anisotropic pre-filtering, eliminating aliasing. Please consult our supplementary materials for video results.

Fig. 16. Filtering of 4D light stage-data leads to a smoothing of the illumination condition. Notice how the hard shadows in the original signal (left) are smoothed out in the filtered version (right).

Table 5. Fitting and evaluation time as well as model size on disk for different methods. "Pref. Fit Time" measures fitting a prefiltered image/SDF whose highest frequencies were removed by a Gaussian filter with $\sigma^2 = 10^{-4}$. "#Evaluations" denotes the number of model evaluations necessary to obtain a filtered output.

Method	Disk Size	Image				SDF			
		Fit Time	Pref. Fit Time	Eval. Time	#Evaluations	Fit Time	Pref. Fit Time	Eval. Time	#Evaluations
BACON	5 MB	89 s	62 s	1.6 s	1	2021 s	1538 s	6.2 s	1
PNF	7 MB	2491 s	1338 s	5.7 s	1	—	—	—	—
MINER	17 MB	3 s	2 s	0.1 s	1	18 s	15 s	0.1 s	1
INSP	4 MB	83 s	11 s	189.5 s	32[a]	270 s	255 s	575.7 s	22[a]
NFC	1 MB	—[b]	—[b]	63.9 s	145-169	—[b]	—[b]	505.9 s	216-343
MLP	24 MB	15 s	12 s	1.8 s	1	107.6 s	75 s	7.3 s	1
Ours	24 MB	74 s	36 s	1.8 s	1	1294.1 s	413 s	7.3 s	1

[a] Includes evaluations of derivative networks obtained using automatic differentiation.
[b] The method did not reach the PSNR/Chamfer distance threshold.

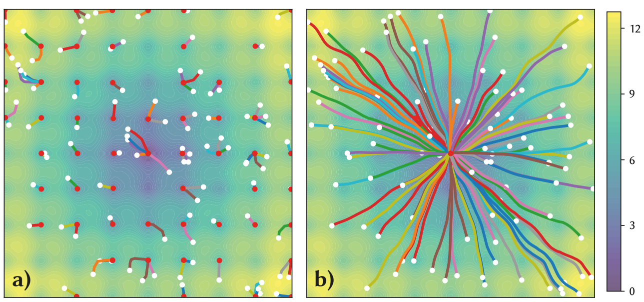

Fig. 17. Our approach applied to a continuous optimization problem involving an energy landscape with multiple local minima. (*a*) Performing gradient descent starting from random initializations (white dots) is prone to converging to the closest local minimum (red dots). (*b*) Our scale-space field allows almost all initializations to converge to the global minimum.

access to the continuous energy landscape of an optimization problem. We learn this landscape using our approach, which enables a coarse-to-fine optimization, successfully preventing the routine from getting stuck in local minima.

We demonstrate this capability using the 2D Ackley [1987] function, which consists of a global minimum surrounded by multiple local minima. A random initialization in the domain followed by gradient descent is prone to converging to one of the local minima (Fig. 17a). As a remedy, we first learn the continuous scale space of the energy landscape. Then, starting from a coarse-scale landscape, we optimize the continuous location of a point using gradient descent based on automatic differentiation through our field, while progressively transitioning to finer scales. We observe that 99% of randomly initialized points converge to the global minimum (Fig. 17b), while only 5% of points reach their goal with the single-scale baseline.

5.3 Ablations

In this section, we analyze individual components of our method using ablational studies. We use the full anisotropic image filtering setting (Sec. 5.1.1) for this investigation and report filtering quality for different configurations in Tab. 6.

Table 6. Ablations.

Configuration	PSNR↑	LPIPS↓	SSIM↑
w/o Sobol	34.07	0.072	0.930
w/o Freq. Warping	33.37	0.077	0.916
Freq. Scaling Only	20.66	0.541	0.563
w/o Lipschitz	21.37	0.216	0.634
10-Lipschitz	32.73	0.076	0.915
Spectral Norm.	29.08	0.127	0.868
ℓ_1-Loss	29.73	0.081	0.884
Ours	**34.82**	**0.069**	**0.940**

First, we are concerned with our Fourier feature dampening. We consider uncorrelated random sampling of frequencies (w/o Sobol) and removal of the frequency warping (w/o Freq. Warping). Second, we look into the Lipschitz-related components. Specifically, we consider the setup in Fig. 3a, where we learn a neural field and simply dampen the Fourier features after training (Freq. Scaling Only), before investigating configurations in which we plainly remove the Lipschitz bounding (w/o Lipschitz), use a looser bound (10-Lipschitz), or spectral normalization instead of our reparameterization scheme (Spectral Norm.). We also train our fields using an ℓ_1-loss instead of using the ℓ_2-norm in Eq. 10 (ℓ_1-Loss).

We observe that our full method outperforms all alternative configurations.

5.4 Timings and Model Size

In Tab. 5, we list performance statistics across different methods. We report training times needed to achieve 30 PSNR and 0.004 Chamfer distance on unfiltered images or SDFs, respectively. We additionally measure the time and number of network evaluations required to produce a filtered output. Finally, we report model sizes when stored to disk. All experiments utilize a single NVIDIA A40 GPU.

We observe that our method is generally on par with or faster than BACON, PNF, INSP, and also NFC, which requires orders-of-magnitude more network evaluations than our approach. While MINER is faster, it is supervised on prefiltered data. A vanilla multi-layer perceptron in the form of Eq. 5 (MLP) is also faster, but does not produce a scale space.

5.5 Discussion

While a Gaussian filter directly dampens amplitudes of output frequencies, our method dampens amplitudes of encoding frequencies and then relies on the Lipschitz bound to carry this dampening through to the output. We demonstrate consequences of this difference in Fig. 18. In the top row, observe that our method reduces the amplitude of the original sine wave like a Gaussian filter. However, the peaks of our wave are sharper, approaching a sawtooth wave that one would obtain from just limiting the slope of the original wave. In the spectrum, this manifests in the emergence of harmonic frequencies. Fortunately, this effect is barely visible in more complex signals as seen in the bottom row.

Neural networks exhibit an inductive bias against learning a high-frequency output when only low Fourier encoding frequencies are present [Rahaman et al. 2019]. The Lipschitz bound turns this bias into a hard constraint. Thus, it becomes even more important to include sufficiently high encoding frequencies, or else the unfiltered reconstruction is inadvertently bandlimited. In addition, the Lipschitz bound restricts the freedom of the neural network to learn arbitrary functions. We find increasing the network width to be an effective countermeasure.

Neural Radiance Fields [Mildenhall et al. 2020] are a popular application of neural fields, and combining them with our method could enable cheap anti-aliasing. Unfortunately, they exhibit a very high dynamic range in volumetric density, which poses a significant challenge for a Lipschitz-bounded network to fit. Our preliminary experiments indicate that more work is needed to accommodate this specific modality.

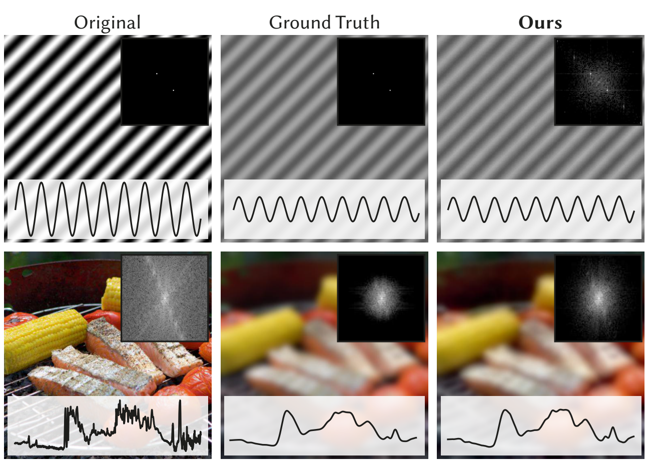

Fig. 18. Original images (left) are smoothed by both a Gaussian filter (middle) and our method (right). The 1D plots show a diagonal slice through the respective image. The upper-right insets depict logarithmic spectra. While both approaches produce similar results, our method introduces slight harmonics on simple signals like the sine wave in the top row. Best viewed digitally and zoomed-in.

Many of the baseline methods we consider do not generate a continuous scale space. Instead, they output a discrete set of filtered signals, which can be linearly combined to approximate all scales that lie in between. Our method similarly combines a finite set of Fourier frequencies. In contrast to these baselines, however, our combination is performed by a highly non-linear MLP, which we observe to eliminate all traces of discretization.

6 CONCLUSION

We have introduced neural Gaussian scale-space fields, a novel paradigm that allows to learn a scale space from raw data. Crucially, we have shown that a faithful approximation of a continuous, anisotropic scale space can be obtained without computing convolutions of a signal with Gaussian kernels. Our idea relies on a careful fusion of strategically dampened Fourier features in a positional encoding and a Lipschitz-bounded neural network. The approach is lightweight, efficient, and versatile, which we have demonstrated on a range of modalities and applications.

We see plenty opportunity for future work. From a theoretical perspective, it would be interesting (and ultimately necessary) to obtain a deeper understanding of *why* dampened Fourier features fed into a Lipschitz-bounded network result in a good approximation of Gaussian filtering. In terms of applications, we think that our approach can potentially be a useful tool for ill-posed inverse problems in neuro-explicit frameworks, such as inverse rendering or surface reconstruction.

In light of recent efforts in continuous modeling of the real world, we hope that our neural Gaussian scale-space fields contribute a useful component to the toolbox of researchers and practitioners.

REFERENCES

David H. Ackley. 1987. *A Connectionist Machine for Genetic Hillclimbing.* Kluwer Academic Publishers.

Cem Anil, James Lucas, and Roger Grosse. 2019. Sorting Out Lipschitz Function Approximation. In *International Conference on Machine Learning (ICML)*.

Andreas Antoniou. 2006. *Digital Signal Processing.* McGraw-Hill.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning (ICML)*.

Martin Arjovsky, Amar Shah, and Yoshua Bengio. 2016. Unitary Evolution Recurrent Neural Networks. In *International Conference on Machine Learning (ICML)*.

Jean Babaud, Andrew P. Witkin, Michel Baudin, and Richard O. Duda. 1986. Uniqueness of the Gaussian Kernel for Scale-Space Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8, 1 (1986), 26–33.

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 *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2023. Zip-NeRF: Anti-Aliased Grid-Based Neural Radiance Fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. 2019. Invertible Residual Networks. In *International Conference on Machine Learning (ICML)*.

Åke Björck and Clazett Bowie. 1971. An Iterative Algorithm for Computing the Best Estimate of an Orthogonal Matrix. *SIAM J. Numer. Anal.* 8, 2 (1971), 358–364.

E. Oran Brigham. 1988. *The fast Fourier transform and its applications.* Prentice-Hall.

Peter J. Burt. 1981. Fast filter transform for image processing. *Computer Graphics and Image Processing* 16, 1 (1981), 20–51.

Vladimir Bychkovsky, Sylvain Paris, Eric Chan, and Frédo Durand. 2011. Learning Photographic Global Tonal Adjustment with a Database of Input/Output Image Pairs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yinbo Chen, Sifei Liu, and Xiaolong Wang. 2021. Learning Continuous Image Representation With Local Implicit Image Function. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yun-Chun Chen, Vladimir Kim, Noam Aigerman, and Alec Jacobson. 2023a. Neural Progressive Meshes. In *ACM SIGGRAPH Conference*.

Zhang Chen, Zhong Li, Liangchen Song, Lele Chen, Jingyi Yu, Junsong Yuan, and Yi Xu. 2023b. NeuRBF: A Neural Fields Representation with Adaptive Radial Basis Function. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. 2017. Parseval Networks: Improving Robustness to Adversarial Examples. In *International Conference on Machine Learning (ICML)*.

Ingrid Daubechies. 1988. Orthonormal Bases of Compactly Supported Wavelets. *Communications on Pure and Applied Mathematics* 41, 7 (1988), 909–996.

Leo Dorst and Rein Van den Boomgaard. 1994. Morphological signal processing and the slope transform. *Signal Processing* 38, 1 (1994), 79–98.

Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. 2020. Multiplicative Filter Networks. In *International Conference on Learning Representations (ICLR)*.

LMJ Florack, Alfons H Salden, Bart M ter Haar Romeny, Jan J Koenderink, and Max A Viergever. 1995. Nonlinear scale-space. *Image and Vision Computing* 13, 4 (1995), 279–294.

William T. Freeman and Edward H. Adelson. 1991. The Design and Use of Steerable Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 9 (1991), 891–906.

David Gargan and Francis Neelamkavil. 1998. Approximating Reflectance Functions Using Neural Networks. In *Eurographics Workshop on Rendering Techniques*.

Alban Gauthier, Robin Faury, Jeremy Levallois, Theo Thonat, Jean-Marc Thiery, and Tamy Boubekeur. 2022. MIPNet: Neural Normal-to-Anisotropic-Roughness MIP Mapping. *ACM Transactions on Graphics* 41, 6 (2022).

J.-M. Geusebroek, Arnold W.M. Smeulders, and Joost Van De Weijer. 2003. Fast anisotropic Gauss filtering. *IEEE Transactions on Image Processing* 12, 8 (2003), 938–943.

Adam Golinski, Mario Lezcano-Casado, and Tom Rainforth. 2019. Improving Normalizing Flows via Better Orthogonal Parameterizations. In *ICML Workshop on Invertible Neural Networks and Normalizing Flows*.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J. Cree. 2021. Regularisation of neural networks by enforcing Lipschitz continuity. *Machine Learning* 110 (2021), 393–416.

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.

Jens André Griepentrog, Wolfgang Höppner, Hans-Christoph Kaiser, and Joachim Rehberg. 2008. A bi-Lipschitz continuous, volume preserving map from the unit ball onto a cube. *Note di Matematica* 28, 1 (2008), 177–193.

Kanghui Guo, Gitta Kutyniok, and Demetrio Labate. 2006. Sparse Multidimensional Representations using Anisotropic Dilation and Shear Operators. *Wavelets and Splines* 14 (2006), 189–201.

Paul S. Heckbert. 1986. Survey of Texture Mapping. *IEEE Computer Graphics and Applications* 6, 11 (1986), 56–67.

Paul S. Heckbert. 1989. *Fundamentals of Texture Mapping and Image Warping*. Master's thesis. University of California, Berkeley.

Matthias Hein and Maksym Andriushchenko. 2017. Formal Guarantees on the Robustness of a Classifier against Adversarial Manipulation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kyle Helfrich, Devin Willmott, and Qiang Ye. 2018. Orthogonal Recurrent Neural Networks with Scaled Cayley Transform. In *International Conference on Machine Learning (ICML)*.

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* 37, 6 (2018).

Amir Hertz, Or Perel, Raja Giryes, Olga Sorkine-Hornung, and Daniel Cohen-Or. 2021. SAPE: Spatially-Adaptive Progressive Encoding for Neural Optimization. (2021).

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 *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Lei Huang, Xianglong Liu, Bo Lang, Adams Yu, Yongliang Wang, and Bo Li. 2018. Orthogonal Weight Normalization: Solution to Optimization Over Multiple Dependent Stiefel Manifolds in Deep Neural Networks. In *AAAI Conference on Artificial Intelligence*.

Stephanie Hyland and Gunnar Rätsch. 2017. Learning Unitary Operators with Help From u(n). In *AAAI Conference on Artificial Intelligence*.

Taizo Iijima. 1959. Basic theory of pattern observation. *Technical Group on Automata and Automatic Control* (1959), 3–32.

Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Li Jing, Yichen Shen, Tena Dubcek, John Peurifoy, Scott Skirlo, Yann LeCun, Max Tegmark, and Marin Soljačić. 2017. Tunable Efficient Unitary Neural Networks (EUNN) and their application to RNNs. In *International Conference on Machine Learning (ICML)*.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive Growing of GANs for Improved Quality, Stability, and Variation. In *International Conference on Learning Representations (ICLR)*.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. 2019. Time2Vec: Learning a Vector Representation of Time. *arXiv preprint arXiv:1907.05321* (2019).

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*.

Jan J. Koenderink. 1984. The structure of images. *Biological Cybernetics* 50, 5 (1984), 363–370.

Alexandr Kuznetsov, Krishna Mullia, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. 2021. NeuMIP: Multi-Resolution Neural Materials. *ACM Transactions on Graphics* 40, 4 (2021).

Mario Lezcano-Casado and David Martınez-Rubio. 2019. Cheap Orthogonal Constraints in Neural Networks: A Simple Parametrization of the Orthogonal and Unitary Group. In *International Conference on Machine Learning (ICML)*.

Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. 2021. BARF: Bundle-Adjusting Neural Radiance Fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Tony Lindeberg. 1997. On the Axiomatic Foundations of Linear Scale-Space. In *Gaussian Scale-Space Theory*. Springer, 75–97.

Tony Lindeberg. 2013. *Scale-Space Theory in Computer Vision*. Vol. 256. Springer Science & Business Media.

David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. 2022. BACON: Band-limited Coordinate Networks for Multiscale Scene Representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hsueh-Ti Derek Liu, Francis Williams, Alec Jacobson, Sanja Fidler, and Or Litany. 2022. Learning Smooth Neural Functions via Lipschitz Regularization. In *ACM SIGGRAPH Conference*.

Li Ma, Xiaoyu Li, Jing Liao, Qi Zhang, Xuan Wang, Jue Wang, and Pedro V Sander. 2022. Deblur-NeRF: Neural Radiance Fields From Blurry Images. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Stephane G. Mallat. 1989. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11, 7 (1989), 674–693.

David Marr and Ellen Hildreth. 1980. Theory of Edge Detection. *Proceedings of the Royal Society of London, Biological Sciences* 207, 1167 (1980), 187–217.

Alexander Mathiasen, Frederik Hvilshøj, Jakob Rødsgaard Jørgensen, Anshul Nasery, and Davide Mottin. 2020. What if Neural Networks had SVDs?. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. 2021. Modulated Periodic Activations for Generalizable Local Functional Representations. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zakaria Mhammedi, Andrew Hellicar, Ashfaqur Rahman, and James Bailey. 2017. Efficient Orthogonal Parametrisation of Recurrent Neural Networks Using Householder Reflections. In *International Conference on Machine Learning (ICML)*.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *ECCV*.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. 2018. Spectral Normalization for Generative Adversarial Networks. In *International Conference on Learning Representations (ICLR)*.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. 2022. Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics* 41, 4 (2022).

Seungtae Nam, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. 2023. Mip-Grid: Anti-aliased Grid Representations for Neural Radiance Fields. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Harald Niederreiter. 1992. Low-discrepancy point sets obtained by digital constructions over finite fields. *Czechoslovak Mathematical Journal* 42, 1 (1992), 143–166.

Ntumba Elie Nsampi, Adarsh Djeacoumar, Hans-Peter Seidel, Tobias Ritschel, and Thomas Leimkühler. 2023. Neural Field Convolutions by Repeated Differentiation. *ACM Transactions on Graphics* 42, 6 (2023).

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable Neural Radiance Fields. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *NeurIPS Workshop on Autodiff*.

Hallison Paz, Tiago Novello, Vinicius Silva, Luiz Schirmer, Guilherme Schardong, and Luiz Velho. 2022. Multiresolution Neural Networks for Imaging. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*.

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 (ICML)*.

Ali Rahimi and Benjamin Recht. 2007. Random Features for Large-Scale Kernel Machines. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Vishwanath Saragadam, Jasper Tan, Guha Balakrishnan, Richard G. Baraniuk, and Ashok Veeraraghavan. 2022. MINER: Multiscale Implicit Neural Representation. In *European Conference on Computer Vision (ECCV)*.

Shayan Shekarforoush, David Lindell, David J. Fleet, and Marcus A. Brubaker. 2022. Residual Multiplicative Filter Networks for Multiscale Reconstruction. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Assaf Shocher, Ben Feinstein, Niv Haim, and Michal Irani. 2020. From Discrete to Continuous Convolution Layers. *arXiv preprint arXiv:2006.11120* (2020).

Eero P. Simoncelli and William T. Freeman. 1995. The steerable pyramid: a flexible architecture for multi-scale derivative computation. In *IEEE International Conference on Image Processing (ICIP)*, Vol. 3. 444–447.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. 2020. Implicit Neural Representations with Periodic Activation Functions. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ilya Meerovich Sobol. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki* 7, 4 (1967), 784–802.

Kenneth O. Stanley. 2007. Compositional Pattern Producing Networks: A Novel Abstraction of Development. *Genetic Programming and Evolvable Machines* 8 (2007), 131–162.

Jean-Luc Starck, Fionn D. Murtagh, and Albert Bijaoui. 1998. *Image Processing and Data Analysis: The Multiscale Approach.* Cambridge University Press.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*.

Towaki Takikawa, Alex Evans, Jonathan Tremblay, Thomas Müller, Morgan McGuire, Alec Jacobson, and Sanja Fidler. 2022. Variable Bitrate Neural Fields. In *ACM SIGGRAPH Conference*.

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. 2021. Neural Geometric Level of Detail: Real-time Rendering with Implicit 3D Shapes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. 2020. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. 2022. Advances in Neural Rendering. *Computer Graphics Forum* 41, 2 (2022), 703–735.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Shenlong Wang, Simon Suo, Wei-Chiu Ma, Andrei Pokrovsky, and Raquel Urtasun. 2018. Deep Parametric Continuous Convolutional Neural Networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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.

Joachim Weickert. 1998. *Anisotropic Diffusion in Image Processing.* Teubner Stuttgart.

Lance Williams. 1983. Pyramidal Parametrics. In *ACM SIGGRAPH Conference*.

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

Zhijie Wu, Yuhe Jin, and Kwang Moo Yi. 2023. Neural Fourier Filter Bank. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yuanbo Xiangli, Linning Xu, Xingang Pan, Nanxuan Zhao, Anyi Rao, Christian Theobalt, Bo Dai, and Dahua Lin. 2022. BungeeNeRF: Progressive Neural Radiance Field for Extreme Multi-scale Scene Rendering. In *European Conference on Computer Vision (ECCV)*.

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. *Computer Graphics Forum* 41, 2 (2022), 641–676.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2019. Self-attention with Functional Time Representation Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Dejia Xu, Peihao Wang, Yifan Jiang, Zhiwen Fan, and Zhangyang Wang. 2022. Signal Processing for Implicit Neural Representations. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Xingqian Xu, Zhangyang Wang, and Humphrey Shi. 2021. UltraSR: Spatial Encoding is a Missing Key for Implicit Image Function-based Arbitrary-Scale Super-Resolution. *arXiv preprint arXiv:2103.12716* (2021).

Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. 2021. Geometry Processing with Neural Fields. *Advances in Neural Information Processing Systems (NeurIPS)* (2021).

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. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jiawei Yang, Marco Pavone, and Yue Wang. 2023. FreeNeRF: Improving Few-shot Neural Rendering with Free Frequency Regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yuichi Yoshida and Takeru Miyato. 2017. Spectral Norm Regularization for Improving the Generalizability of Deep Learning. *arXiv preprint arXiv:1705.10941* (2017).

Jiong Zhang, Qi Lei, and Inderjit Dhillon. 2018b. Stabilizing Gradients for Deep Neural Networks via Efficient SVD Parameterization. In *International Conference on Machine Learning (ICML)*.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018a. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yiyu Zhuang, Qi Zhang, Ying Feng, Hao Zhu, Yao Yao, Xiaoyu Li, Yan-Pei Cao, Ying Shan, and Xun Cao. 2023. Anti-Aliased Neural Implicit Surfaces with Encoding Level of Detail. In *ACM SIGGRAPH Conference*.