PREF: PHASORIAL EMBEDDING FIELDS FOR COMPACT NEURAL REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

We present an efficient frequency-based neural representation termed PREF: a shallow MLP augmented with a phasor volume that covers significant border spectra than previous Fourier feature mapping or Positional Encoding. At the core is our compact 3D phasor volume where frequencies distribute uniformly along a 2D plane and dilate along a 1D axis. To this end, we develop a tailored and efficient Fourier transform that combines both Fast Fourier transform and local interpolation to accelerate naïve Fourier mapping. We also introduce a Parsvel regularizer that stables frequency-based learning. In these ways, Our PREF reduces the costly MLP in the frequency-based representation, thereby significantly closing the efficiency gap between it and other hybrid representations, and improving its interpretability. Comprehensive experiments demonstrate that our PREF is able to capture high-frequency details while remaining compact and robust, including 2D image generalization, 3D signed distance function regression and 5D neural radiance field reconstruction.

1 INTRODUCTION

We recently witness considerable advances in implicit neural representations that remain fast and compact while being capable to render high-frequency details (Martel et al., 2021; Sun et al., 2021; Yu et al., 2021a; Müller et al., 2022; Chen et al., 2022a). To trade memory for speed, they are typified by a hybrid representation: a shallow but efficient MLP augmented with a luxury data structure like dense grid (Hedman et al., 2021; Sun et al., 2021), tree (Yu et al., 2021a; Takikawa et al., 2021), point cloud (Xu et al., 2022) or mesh (Yang et al., 2022; Munkberg et al., 2022). As usual, they encode the inputs x, *i.e.*, coordinates, into a high dimensional embedding field $\mathbf{f}(x)$:

$$\mathbf{f}(x) = \sum_{i}^{N_v} \phi(x - x_i) \cdot \mathbf{v}_i \tag{1}$$

where $\mathbf{v}_i$ is the learnable embedding vector associated with an often fixed and data structure-dependent location x_i. $\phi(x)$ is an interpolation kernel, usually local and linear, to achieve an efficient query. To recover high frequencies, they require discretizing the spatial location at a very fine level that results in a high memory footprint (Sun et al., 2021), or joint co-adapt the vertices (by splitting, pruning or subdivision) into a heuristic structure like tree or mesh to capture high-frequency locals (Martel et al., 2021). As a result, they may complicate the training process or limit the representation to be task-specific.

By contrast, frequency encoding does not impose any specific data structure to recover high frequencies (Mildenhall et al., 2020; Tancik et al., 2020). In fact, it facilitates easy queries and optimizations of specific frequencies of the input embedding field, despite relying on a mathematically simple (inverse) Fourier transform:

$$\mathbf{f}(x) = \sum_{i}^{N_k} e^{j2\pi x k_i} \cdot \mathbf{p}_i \tag{2}$$

where $\mathbf{p}_i$ is a complex-valued vector, associated with a frequency coordinate k_i. We call $\mathbf{p}_i$ as phasor because its entry implies a wave of a certain frequency, phase and magnitude. From the Nyquist-Shannon sampling perspective, a phasor volume should have the same expressiveness (bandwidth) as
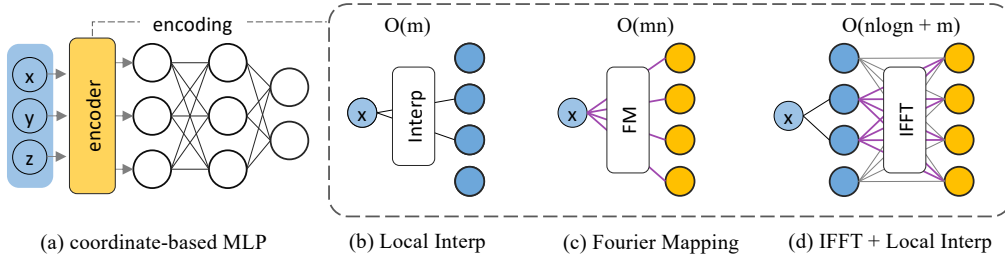
Figure 1: A conceptual overview of accelerating frequency-based implicit neural representation. Assume there are m coordinates in a single iteration and n embedding nodes in the encoding layer.

a uniform spatial volume with the same size, *i.e.*, $N_v = N_k$ and $\{k_i\}$ is a set of uniform-distributed Fourier basis. Therefore, frequency encoding should match the performance when having dense spectra. In fact, frequency representation turns out to be more compact than a dense parameterization scheme by directly accessing sparse frequencies. Tancik et al. (2020) demonstrate Fourier mapping with a set of exponential-increased or Gauss-sampled Fourier basis outperforms that of uniform-spaced (and therefore a dense grid). One explanation is that natural signals do exhibit some smoothness or correlations, so an MLP augmented with relatively sparse spectra is sufficient. Despite being compact, frequency representations seem less attractive compared to hybrid representations (Sun et al., 2021; Müller et al., 2022), mainly because of their inefficiency in domain transform (*i.e.*, Fourier transform). For example, NeRF's MLPs (Mildenhall et al., 2020) use only limited spectrum, *e.g.*, $N_k = 24$ as opposite to $N_v = 160^3$ used in a dense volume (Sun et al., 2021). As a result, they should equip a costly larger MLP that incurs longer training and inference time. To allow a better trade-off between memory and speed, it is desirable to scale up spectra and improve the transform efficiency.

To this end, we propose a PhasoRial Embedding Field: a shallow MLP augmented with a phasor volume that covers significant border spectra than previous frequency representations (Tancik et al., 2020). To retain compactness and efficiency, we devise a phasor volume where frequencies distribute uniformly along a 2D plane and dilate along a 1D axis. Then, we develop a tailored approximated algorithm that combines both Fast Fourier transform (FFT) and local interpolation to accelerate naïve Fourier mapping. To make PREF stable, we also propose a Parsvel regularizer tailored for the phasor layer, an analogy to the Total Variation (TV) regularizer in local parameterization schemes. Moreover, the design of PREF also improves interpretability by assigning the majority of network capacity to the phasor volume instead of the black-box MLPs. As a result, we can modulate the phasor embedding to edit the neural field, such as level-of-details filtering, which is either inconvenient or even not applicable for the recent rising hybrid encoders.

To summarize, the contributions of our work include: (1) A compact and efficient approach called PREF that is more cable of capturing high-frequency details, better than all previous frequency-based neural representations and comparative to existing hybrid representations, such as dense grid and Octree. Extensive experiments, ranging from 2D image generalization, and 3D signed distance function fitting to 5D radiance field reconstruction, illustrate this. (2) A Parsvel regularizer, tailored for frequency encoding, is proposed as an analogy to other regularizers used in fully connected layers or local interpolation layers.

2 RELATED WORK

Our PREF framework is in line with renewed interest in adopting implicit neural networks to represent continuous signals from low dimensional input. In computer vision and graphics, they include 2D images, 3D surfaces in the form of occupancy fields (Mescheder et al., 2019; Niemeyer et al., 2020; Oechsle et al., 2021; Peng et al., 2020) or signed distance fields (SDFs) (Atzmon & Lipman, 2020; Gropp et al., 2020; Wang et al., 2021a; Yariv et al., 2020; Martel et al., 2021; Takikawa et al., 2021; Yariv et al., 2021; Wang et al., 2021b; Park et al., 2019), concrete 3D volumes with a density field (Ji et al., 2017; Qi et al., 2016), 4D light fields (Levoy & Hanrahan, 1996; Wood et al., 2000) and 5D plenoptic functions for the radiance fields (Mildenhall et al., 2020; Zhang et al., 2020; Liu et al., 2020; Park et al., 2021; Yariv et al., 2020; Chen et al., 2022b).

Frequency encoding. To learn high frequencies, state-of-the-art implicit neural networks have adopted the Fourier encoding scheme by transforming the coordinates with periodic $\sin(x)$ and $\cos(x)$ functions or equivalently, under Euler's formula e^{jx}. Under Fourier encoding (Tancik et al., 2020; Zhong et al., 2021; Mildenhall et al., 2020; Liu et al., 2021), feature optimization through MLPs can be mapped to optimizing complex-valued matrices with complex-valued inputs in the linear layer. Specifically, Position Encoding (PE) and Fourier Feature Maps (FFM) both transform spatial coordinates to Fourier basis at earlier input layers whereas SIREN (Sitzmann et al., 2020) and MFN (Fathony et al., 2020) embed the process in the deeper layers by using periodic activation functions. In the computer graphics community, frequency representation has used Fourier coefficients to represent opacity (Jansen & Bavoil, 2010) and spherical harmonics (SH) (Müller, 2006) to represent view-dependent effect. In the context of neural fields for graphics, accessing frequency representations has also led to many other advances. For example, frequency representations allow level-of-detail smoothing and multi-scale anti-aliasing rendering (Lindell et al., 2022; Barron et al., 2021). Like BACON (Lindell et al., 2022) and FFM (Tancik et al., 2020), the output of our phasor encoder can be also expressed as a summation of sines. Whereas their frequencies are randomly sampled and the transformation is exactly computed, our frequency support is tailored so that allows a fast and approximated Fourier Transform. Fig. 1 briefly outlines the differences. Beyond neural fields, Fourier-CPNN (Tesfaldet et al., 2019) has proposed to learn Fourier coefficients to synthesize the image. However, the coefficients are learned instead of optimized from scratch so that they may incorrectly admit continuity in the spectra, and consequently, hinders the performance.

Hybrid representation. Improving the training and inference efficiency of MLP-based networks has been explored from the embedding perspective with smart data structures. Various schemes (Reiser et al., 2021; Hedman et al., 2021; Yu et al., 2021b;a; Sun et al., 2021; Müller et al., 2022; Chen et al., 2022a) replace the deep MLP architecture with voxel-based representations, to trade memory for speed. Early approaches bake an MLP to an Octree along with a kilo sub-NeRF or 3D texture atlas for real-time rendering (Reiser et al., 2021; Hedman et al., 2021). More smart and compact data structure includes point cloud (Xu et al., 2022) and triangle mesh (Munkberg et al., 2022; Yang et al., 2022). These approaches, however, rely on a pre-trained or half-trained MLP as prior and therefore still incur long per-scene training and complicate the training process. Plenoxel (Yu et al., 2021a) and DVGO (Sun et al., 2021) directly optimize the density values on discretized voxels and employ an auxiliary shading function represented by either spherical harmonics (SH) or a shallow MLP network to account for view dependency. They achieve orders of magnitude acceleration over the original NeRF on training but incur a very large memory footprint by storing per-voxel features. Besides, over parametrization can easily lead to noisy density estimation and subsequently inaccurate surface estimations and rendering. Seminal works have attempted to store and query voxel features in an easier and much more compact way. Instant-NGP (Müller et al., 2022) stores and looks up features with a hash table to achieve unprecedentedly fast training and very visual results. EG3D (Chan et al., 2022) queries feature from a compact tri-plane representation. In a similar vein, TensoRF (Chen et al., 2020) employs highly efficient tensor decomposition via vertical projections. Whereas these methods differ in how they look up the very local substitute, our method computes feature substitute that relates to the global parameter space through an approximated Fourier transform. The difference is that the frequency of the field is now directly related to the input phasor field, and the optimization is not local anymore so that PREF remains robust to recover high frequency details, reducing the risk of overfitting to certain locals.

3 PHASORIAL EMBEDDING FIELDS

PREF consists of a dense frequency encoder and a shallow MLP. We illustrate our design with a 3D signal for example. We denote the phasor volume $\mathbf{P}$ as a 3D complex-valued feature volume with embedding length k, and denote the encoder as $\mathcal{T}$ thus a local high dimensional embedding is obtained by $\mathbf{z}(\mathbf{x}) = \mathcal{T}(\mathbf{P}) \in \mathbb{R}^k$. Although other time-frequency transforms exist, this paper focus on the simple Fourier transforms so as to connect to previous frequency encoders (Mildenhall et al., 2020; Tancik et al., 2020). Similar to previous hybrid representation, the transformed embedding can be fed into an MLP to model task-specific fields such as signed distance fields or radiance fields. We first introduce the design of the phasor volume $\mathbf{P}$ (Sec. 3.1), and along the way, describe its tailored approximated Fourier transform $\mathcal{T}$ (Sec. 3.2) and regularizer $\mathcal{L}_{\text{reg}}$ (Sec. 3.3). We provide an overall visual depiction in Fig. 2. We also include the pseudo-code in the Supp. B.
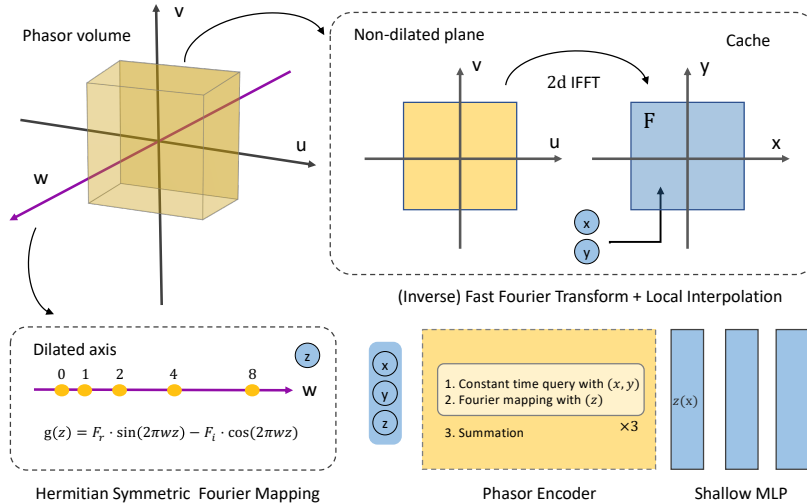
Figure 2: Visual depiction of our PREF. See text for the detail illustration of phasor volume (Sec. 3.1) and approximated inverse Fourier transform (Sec. 3.2).

3.1 DILATED PHASOR VOLUME

To preserve high frequencies, spatial embedders require discretizing the volume at a very fine level. By contrast, frequency modeling manages to query specific frequencies without resorting to large volume size and memory. To provide a better trade-off between speed and memory, we devise a tiny volume and dilate it to a large volume with certain frequencies skipped. In particular, we use a 3D tiny volume $\mathbf{P} \in \mathbb{C}^{n \times n \times d}$ (we omit the feature dimension k for clarity), with d being a relative small number compared to n. We then assign the frequency to be linear-increased in the first two dimension, *i.e.*, $\mathbf{u} = \mathbf{v} = \{\lfloor -n/2 \rfloor, \lfloor -n/2 \rfloor + 1, \cdots, \lfloor n/2 \rfloor\}$, and dilate that of the tiny dimension to be exponential increased $\mathbf{w} = \{0, 1, \cdots, 2^{d-2}\}^1$, following Mildenhall et al. (2020). The frequency represented by the phasor volume then covers $\mathbf{v} \times \mathbf{u} \times \mathbf{w}$. To avoid the representation bias on certain axes, we then sum the three separate phasor volumes by dilating along each dimension and by reusing the linearity of the Fourier transform,

$$\mathbf{f}(x, y, z) = \mathcal{T}(\mathbf{P^u}) + \mathcal{T}(\mathbf{P^v}) + \mathcal{T}(\mathbf{P^w}), \tag{3}$$

with the superscript denoting the dilated dimension. It is also noteworthy that if the length of the non-dilated dimensions equals 1, *i.e.*, only a few on-axis dilated frequencies comprise the support, our PREF degenerates to the Positional Encoding (Mildenhall et al., 2020). Similar to proceeding frequency encoders, the volume shapes serve as hyper-parameters and need tuning to achieve a better trade-off between memory and speed. Notice that the three volumes can be rearranged into a single larger volume $\mathbf{P} \in \mathbb{C}^{n \times n \times n}$ with many entries being zero. In the following, we discuss the transformation of $\mathbf{P}$ and drop the superscript for clarity.

3.2 APPROXIMATED INVERSE FOURIER TRANSFORM

Directly applying Fourier mapping (Tancik et al., 2020) from such a large phasor volume is prohibitively expensive. To make the computation tractable, we can approximate it via first computing and storing a transformed embedding grid, from which embedding at arbitrary locations can be efficiently locally interpolated. Since we assume our underlying embedding field is smooth, the approximation error is modest when the grid size is sufficiently large. To illustrate, assuming we have a full phasor volume N^3, we first compute a spatial grid with the same size N^3. This process equals the inverse discrete Fourier Transform (iDFT) that can be efficiently solved by well-established fast Fourier Transform (FFT) methods, e.g., the Cooley–Tukey algorithm (Cooley & Tukey, 1965). In each forward iteration, the inverse FFT is performed once for all coordinates so that $\mathcal{T}$ can be efficient. Fig. 1 presents the conceptual idea.

---

[1]We only keep the positive frequency because we constraint the phasor volume to be hermitian symmetric.
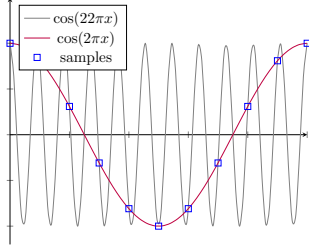
Figure 3: Illustration on overfitting.

(a) wo / $\mathcal{L}_{\text{reg}}$ (b) w / $\mathcal{L}_{\text{reg}}$
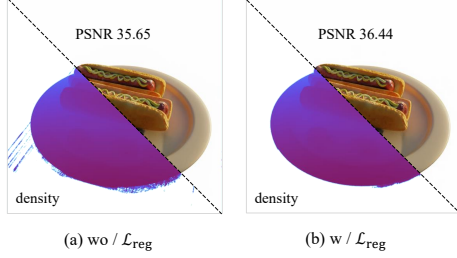
Figure 4: Effect of volume regularization.

To further tailor our dilated phasor volume, we instead employ both iFFT and Fourier mapping to achieve both high accuracy and low complexity. Specifically, we perform 2D iFFT along the non-dilated axes $\mathbf{u}, \mathbf{v}$ to obtain an intermediate volume, a partially transformed volume,

$$\mathbf{F}[\mathbf{x}^*, \mathbf{y}^*, :]^2 = \sum_{\mathbf{i}=0}^{M} \sum_{\mathbf{j}=0}^{N} e^{j2\pi(\mathbf{x}^* \mathbf{u_i} + \mathbf{y}^* \mathbf{v_j})} \mathbf{P}[\mathbf{i}, \mathbf{j}, :] = \text{iFFT2D}(\mathbf{P}[\mathbf{i}, \mathbf{j}, :]) \tag{4}$$

Afterward, linear interpolation and a per-sample hermitian-symmetric Fourier mapping along the dilated axis are conducted:

$$\mathbf{f}(\mathbf{x}, \mathbf{y}, \mathbf{z}) \approx \sum_{k=0}^{d} \underbrace{e^{j2\pi \mathbf{w}_k \mathbf{z}} \widetilde{\mathbf{F}}[\mathbf{x}, \mathbf{y}, \mathbf{k}] + e^{-j2\pi \mathbf{w}_k \mathbf{z}} \widetilde{\mathbf{F}}^*[\mathbf{x}, \mathbf{y}, \mathbf{k}]}_{\text{Hermitian-symmetric Fourier mapping}}$$

$$= \sum_{k=0}^{d} 2\text{Re}(\widetilde{\mathbf{F}})[\mathbf{x}, \mathbf{y}, \mathbf{k}] \sin(2\pi \mathbf{w}_k \mathbf{z}) - 2\text{Im}(\widetilde{\mathbf{F}})[\mathbf{x}, \mathbf{y}, \mathbf{k}] \cos(2\pi \mathbf{w}_k \mathbf{z}) \tag{5}$$

where $\widetilde{\mathbf{F}}$ is a continuous phasor field bilateral-interpolated from the phasor grid $\mathbf{F}$. We force the transformed feature to be real quantity by assuming the phasor field to be hermitian-symmetric, *i.e.*, $\widetilde{\mathbf{F}}[-\mathbf{x}, -\mathbf{y}, -\mathbf{k}]$ is $\widetilde{\mathbf{F}}^*[\mathbf{x}, \mathbf{y}, \mathbf{k}]$, the conjugate of $\widetilde{\mathbf{F}}[\mathbf{x}, \mathbf{y}, \mathbf{k}]$. Note that, the length d in the dilated dimension is generally small (as described in Sec. 3.1). Therefore, per-sample Fourier mapping is very efficient, significantly reducing the training cost.

3.3   Volume Regularization

In many reconstruction tasks, we confront a dilemma between overfitting and generalization due to limited supervision. Overfitting can occur especially when the frequency bandwidth exceeds the minimal sampling rate of the observations (training samples). For example, training samples at a 10 Hz sampling rate from a continuous function $\cos(2\pi x)$ can be also fitted exactly by $\cos(22\pi x)$, where the latter can be treated as overfitting in the lack of prior knowledge. See Fig. 3 for illustration. Therefore, when increasing the sampling rate (*i.e.*, training samples) is not applicable, it is desirable to regularize the representations. Despite plenty of regularizers existing, for example, $L2$ norm (weight decay) for MLP and TV regularizer for discrete grid embedders, they are not directly applicable to frequency encoders. To make the underlying embedding fields smooth, the Fourier coefficient should fall quickly to zero as frequency increases. To this end, we propose a simple, yet effective Parsvel regularizer:

$$\mathcal{L}_{\text{reg}} = ||2\pi \mathbf{u} \mathbf{P}||_2 + ||2\pi \mathbf{v} \mathbf{P}||_2 + ||2\pi \mathbf{w} \mathbf{P}||_2$$

$$= \sqrt{\sum_{\mathbf{i},\mathbf{j},\mathbf{k}} ||2\pi \mathbf{u_i} \mathbf{P}[\mathbf{i}, \mathbf{j}, \mathbf{k}]||^2} + \sqrt{\sum_{\mathbf{i},\mathbf{j},\mathbf{k}} ||2\pi \mathbf{v_j} \mathbf{P}[\mathbf{i}, \mathbf{j}, \mathbf{k}]||^2} + \sqrt{\sum_{\mathbf{i},\mathbf{j},\mathbf{k}} ||2\pi \mathbf{w_k} \mathbf{P}[\mathbf{i}, \mathbf{j}, \mathbf{k}]||^2} \tag{6}$$

This regularizer resembles L_2 norm used in MLP, except that each entry is multiplied by its spatial frequency. Lemma 3.1 illustrates that the Parseval regularizer equals the anisotropic TV regularizer in the spatial domain. A proof based on Parseval's theorem is included in the Supp. A. Implementation-wise, the regularizer can be integrated into modern optimizers without extra cost.

---

[2]We slightly abuse the matrix indexing, where $\mathbf{x}^*$ and $\mathbf{y}^*$ should be mapped to integers.

Table 1: Fast radiance fields reconstruction on the NeRF-synthetic dataset (Mildenhall et al., 2020).

Method	BatchSize	Steps	Time $\downarrow$	Size(MB)$\downarrow$	PSNR$\uparrow$	SSIM$\uparrow$
SRN (Sitzmann et al., 2019)	-	-	>10h	-	22.26	0.846
NeRF (Mildenhall et al., 2020)	4096	300k	~35h	5.0	31.01	0.947
SNeRG (Hedman et al., 2021)	8192	250k	~15h	1771.5	30.38	0.950
NSVF (Liu et al., 2020)	8192	150k	>48h	-	31.75	0.950
PlenOctrees† (Yu et al., 2021b)	1024	200k	~15h	1976.3	31.71	0.958
Plenoxels† (Yu et al., 2021a)	5000	128k	11.4m	778.1	31.71	0.958
DVGO (Sun et al., 2021)	5000	30k	15.0m	612.1	31.95	0.957
NGP† (Müller et al., 2022)	256k	30k	4.6m	87.3	32.63	0.921
TensoRF (Chen et al., 2022a)	4096	30k	17.4m	71.8	33.14	0.963
Ours PREF-128	4096	30k	~ 18m	34.4	32.08	0.952
Ours PREF-64	4096	30k	~18m	9.84	31.18	0.950
Ours PREF-32	4096	30k	~18m	2.28	29.83	0.940

† denotes using customized Cuda kernels.

Lemma 3.1 *Let* $f(x, y, z)$ *be integrable, and* $\mathbf{P}(u, v, w)$ *be its Fourier transform. The anisotropic TV loss of* $f(x, y, z)$ *can be represented by* $||2\pi u\mathbf{P}||_2 + ||2\pi v\mathbf{P}||_2 + ||2\pi w\mathbf{P}||_2$

4 EXPERIMENTS

4.1 NEURAL RADIANCE FIELDS RECONSTRUCTION

For the neural radiance field, we focus on rendering novel views from a set of images with known camera poses. Each RGB value in each pixel corresponds to a ray cast from the image plane. We adopt the volume rendering model used by Mildenhall et al. (2020):

$$\hat{C}(r) = \sum_{i}^{N} T_i(1 - \exp(-\sigma_i\delta_i))c_i, \text{where } T_i = \exp(-\sum_{j=1}^{i-1} \sigma_j\delta_j) \tag{7}$$

where σ_i and c_i are corresponding density and color at location x_i, δ_i is the interval between adjacent samples. Then we optimize the rendered color with the ground truth color with the following loss:

$$\mathcal{L} = \frac{1}{M} \sum_{i=0}^{M} ||C(r) - \hat{C}(r)||^2 + \lambda\mathcal{L}_{\text{Reg}}. \tag{8}$$

PREF Setting. We describe how PREF models the density σ and the radiance c. We use three phasor volume of $256 \times 256 \times 1$ width 16d feature length augmented with a two-layer MLP with hidden dimension 64d. We then use Softplus to map the raw output to the positive-valued density. For the view-dependent radiance branch, we use a volume of $256 \times 256 \times 1$ with 32d feature length followed by a linear layer to output 27d feature embedding. The feature embedding therefore has an approximated Nyquist frequency of 128. To render 5D view-dependent radiance, we concatenate the result with the positional encoded view directions and feed them into a 2-layer MLP with 128d hidden dimension and a linear layer to map the feature to color with Sigmoid activation. All linear layers except the final use ReLU activation, following common practices.

Rendering. To compare with SOTAs (Sun et al., 2021; Chen et al., 2020), we train each scene using 30k iterations with a batch size of 4096 rays. We adopt a progressive training scheme (Hertz et al., 2021): from the resolution of 128 to 256. Specifically, we gradually unlock the higher frequencies at the training step $[2000, 3000, 4000, 5500, 7000]$. Accordingly, the number of samples per ray progressively increases from about 384 to about 1024. This allows us to achieve faster and more stable optimization by first covering the lower frequencies and later high-frequency details. During training, we maintain an alpha mask to skip empty space to avoid unnecessary evaluations.

Optimization. As mentioned in the paper, our PREF uses the Parsvel regularizer $\mathcal{L}_{\text{Reg}}$ on the density phasor volume to avoid overfitting. Our objective is then set to $\mathcal{L} = \mathcal{L}_{\text{RGB}} + \lambda\mathcal{L}_{\text{Reg}}$ with $\lambda = 1e^{-2}$. Without regularization, PREF may overfit specific frequencies, as shown in Fig. 4. On the NeRF synthetic dataset, PREF converges on average 18 minutes with 30k iterations on a single RTX 3090, with a learning rate gradually decaying from 0.02 to 0.002 during the training. The Adam optimizer is used with $\beta_1 = 0.9$ and $\beta_2 = 0.99$ by default.

Table 2: Quantitative results of SDF regression.

Method	Size(MB)	Time	Armadillo		Gargoyle	
			IOU↑	Chamfer$-L_1$↓	IOU↑	Chamfer$-L_1$↓
DV (Sun et al., 2021)	128.0	∼21m	98.81	5.67e-6	97.99	1.19e-5
NGP (Müller et al., 2022)	46.7	∼17m	99.34	5.54e-6	99.42	1.03e-5
PE (Tancik et al., 2020)	6.0	∼26m	96.65	1.21e-5	80.46	1.03e-4
BACON (Lindell et al., 2022)	2.1	∼1.5h	98.33	2.20e-6	98.76	3.65e-5
Ours	36.0	∼21m	99.02	5.57e-6	99.05	1.06e-5

Comparisons. We use the blender dataset (Mildenhall et al., 2020) for training and evaluation since plenty of public benchmarks with speed are available. We report the quantitative and qualitative comparisons in Tab. 1, Fig. 7 respectively. We show that our PREF outperforms its dense volume counterparts and many proceeding designs, whereas NGP (Müller et al., 2022) and TensoRF (Chen et al., 2022a) still present the best image quality. TensoRF, the concurrent work, which factorizes the 3D volume with Tensor decomposition (Chen et al., 2020), seems to show a more effective inductive bias to exploit the correlation within scenes. To match their results, one promising attempt is to combine PREF and Tensor decomposition by reusing the separability of Fourier transform. NGP is also excellent to recover high frequency locals by distributing coordinates to different buckets with customized hash functions. Despite this, the choice of the hash function can be sensitive to the performance and the hashed features may not be directly applicable in generalization tasks.

Ablations. We also include results of using smaller phasor volumes that have 128 and 64 resolution, respectively. We denote them as PREF-64 and PREF-32 according to their Nyquist frequency. We find that PREF with a higher volume size does not yield a better trade-off between memory and speed. Following (Yu et al., 2021a; Sun et al., 2021; Chen et al., 2022a), we can also model the density field with phasor volume without any MLP, denoted as PREF-vanilla. In this case, the density field is the summation of sinusoids, similar to FOM (Jansen & Bavoil, 2010). We tune phasor volumes with size $256 \times 256 \times 10$ and achieve a similar PSNR 31.79, comparable to that augmented with MLP. We advocate the lateral design since it provides a unified form for both density and radiance.

4.2 3D SHAPE REPRESENTATION

Next, we conduct the task of signed distance field (SDF) regression to evaluate its fitting capacity. SDFs describe the shape in terms of a function $f(x)$ as the signed Euclidean distance from the point x to the surface. Our goal is to recover a continuous SDF $f(x)$ given a set of sampled $f(x^*)$.

Data preparation and training. We adopt two widely used models: gargoyle (50k vertices) and armadillo (49k vertices). We scale the model within a bounding box of $[-1, 1]$ and samples $N = 2^{18}$ points for each training epoch: $4/8N$ points on the surface, $3/8N$ points around the surface by adding Gaussian noise to the surface point with $scale = 0.01$, and the last $1/8N$ points uniformly sampled within the bounding box. In each training epoch, we sample a batch size of $N = 2^{18}$ to regress the SDF values. The MAPE loss is used for error back-propagation. To optimize the networks, we use the Adam optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\epsilon = 1e^{-5}$. We use an initial learning rate $1e^{-4}$ and reduce the learning rate to $1e^{-5}$ at the 10th epoch. We adopt a batch size of $N/100$ to optimize all baselines whereas for our method 20 epochs.

Metrics. We report the IOU of the ground truth mesh and the regressed signed distance field by discretizing them into two 128^3 volumes. We also report the Chamfer distance metric by sampling 30k surface points from the extracted mesh with Marching Cube (Lorensen & Cline, 1987).

Baseline implementation. For our PREF, we use three $16 \times 128 \times 128 \times 6$ complex-valued volumes to nonlinearly transform the spatial coordinates to a 16d feature embedding. For the embedding-based baselines, we use our implementation of the dense volume technique. It contains $16 \times 128 \times 128 \times 128$ learnable parameters that transform the input coordinates to their feature embedding at a length 16 by trilinear-interpolation. We also implement NGP (Müller et al., 2022), where they maintain a multi-level hash function to transform the spatial coordinate into feature embedding. We use a 16 num-of-level hash function with dimension 2. Consequently, the output feature embedding is of length 32. All the embedding-based baselines and our PREF adopt the same MLP structure that consists of 3 layers that progressively map the input embedding to the 64 hidden dimension features and finally to a scalar, with ReLU as the intermediate activation. For Position Encoding
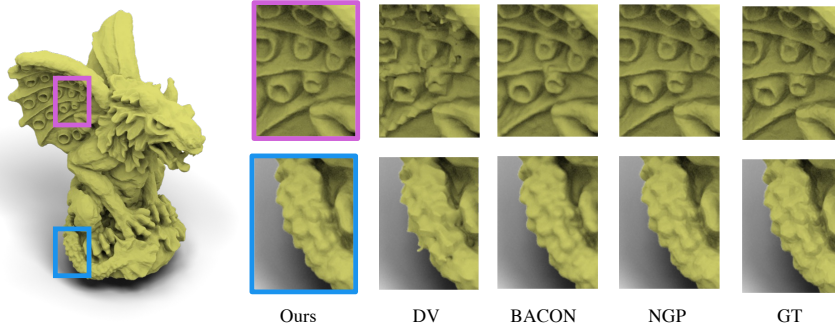
Figure 6: Qualitative visualizations of regressed SDF. The left image shows our regressed result. On the right, we compare our method to a dense volume encoder, BACON (Lindell et al., 2022) and NGP (Müller et al., 2022).

(Mildenhall et al., 2020), we use 6 log-sampled frequencies that roughly achieve 128 resolution. To compensate for such pure implicit models, we use a wider and deeper MLP configuration: 8 linear layers with 512 hidden dimensions. For BACON (Lindell et al., 2022), we direct follow their training configuration to obtain the results. We conduct the experiments on a single RTX 2080Ti.

Comparisons. Tab. 2 lists model size and performance of the baselines vs. PREF. Our method manages to match the SOTA NGP (Müller et al., 2022) with compact model size, and outperform its spatial counterpart dense volume and other frequency-based proceedings. We owe the improvement of PREF to its globally continuous nature that allows for preserving details. NGP also successfully capture high-frequency details without distortion, potentially owing to their multi-scale coordinate inputs that capture global information. PREF only requires a single coordinate input because the multi-scale signals have been encoded in the frequency space.

Level of detail filtering. In line with frequency representations (Lindell et al., 2022; Barron et al., 2021), our PREF also allows implicit surface editing (Yang et al., 2021) such as level of detail filtering. But different from them, our method does not need multi-scale supervision, since we have assigned a major capability of the network to the interpretable phasor embedding fields. Therefore, we can manipulate the trained phasor embedding directly such as multiplying a kernel, which will result in the convolution of transformed fields and that kernel. For example, point-wise multiplying with Gauss kernels of different variations σ yields neural fields of different level-of-details, as Fig. 5 shows. Details are included in Supp D. Such multi-scale representation can benefit many computer graphics tasks like texture removal. Moreover, other predefined or leaned kernels may be further used to deform the neural fields in different styles. To some extent, this experiment also explains why PREF is robust to recovering high frequencies, since a slight change in phasor magnitude will result in global details recovery or suppressing.
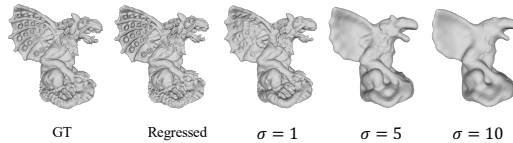

Figure 5: Neural field smoothing.

4.3 UNDERSTANDING THE INDUCTIVE BIAS OF FREQUENCY REPRESENTATION

Despite there are theoretical analyses of the inductive bias of Fourier encoding from the perspective of NTK (Tancik et al., 2020) or dictionary learning (Yüce et al., 2022), we still get less insight on how frequency representation compares to dense grid representation. Theoretically, PREF with linear frequency distribution has the same expressiveness as a dense volume with the same volume size. Despite this, we notice that their expertise is different: PREF excels at recovering textures while dense grid excels at recovering high-fidelity local regions. For example, Fig. 8 shows that PREF better captures the fur while Dense Grid better captures the eye. Despite both two encoders having the same Nyquist frequency and volume size, PREF achieves slightly better PSNR (31.86) than Dense Grid (31.02) due to the large portion of fur. We owe this difference to the optimization behaviors: PREF synthesize a position with periodic sinusoidal while Dense Grid synthesise it with local parameters. For further validation and comparisons to more advanced approaches, we conduct the benchmark image painting task (Huang et al., 2021) which aims to regress the whole image only given 25% pixels. Results are shown in Tab. 3. Implementation detail is in the supplementary.

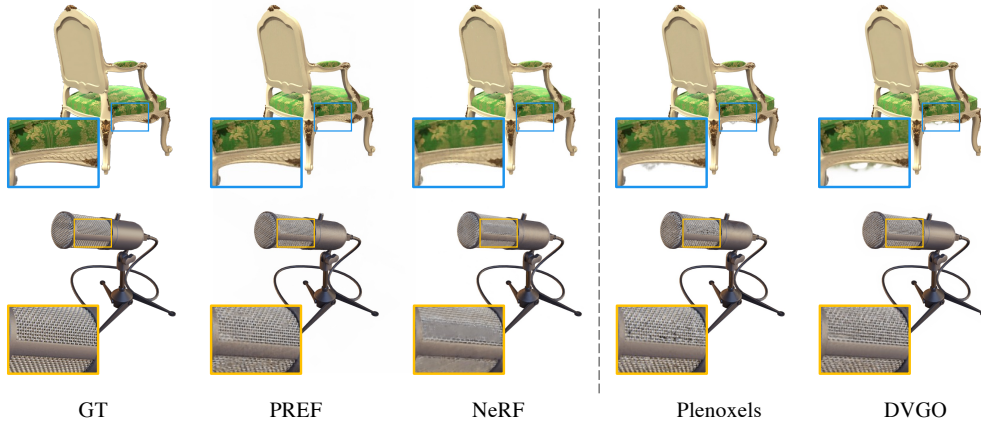GT   PREF   NeRF   Plenoxels   DVGO

Figure 7: Qualitative comparisons. We compare to NeRF (Mildenhall et al., 2020) which encode coordinates with fewer frequencies and larger MLP. We show by covering border spectra, PREF not only reduces the costly MLP but also significantly improves the visual quality. We also compare frequency modelings to dense volume methods (Yu et al., 2021a; Sun et al., 2021), which tend to produce outliers in local regions.

(a) Ground truth   (b) Dense Grid   (c) Phasor Grid (PREF)
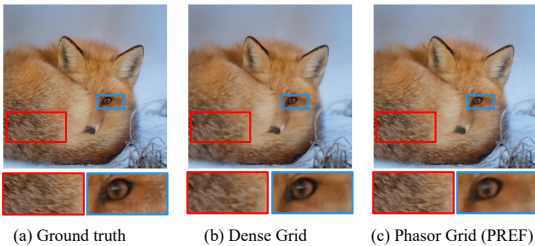
Figure 8: Image Regression (from Natural dataset).

Table 3: Image generalization (PSNR). We compare to PE (Mildenhall et al., 2020), SIREN (Sitzmann et al., 2020) and NGP (Müller et al., 2022).

	Memory (KB)	Natural	Text
Dense Grid	838	23.627 ± 4.085	27.561 ± 2.400
NGP	696	24.137 ± 3.509	27.641 ± 1.852
PE	1318	22.294 ± 3.245	26.852 ± 1.860
SIREN	777	22.394 ± 2.347	25.114 ± 6.680
Ours	545	24.113 ± 3.464	28.329 ± 2.373

5 LIMITATIONS

Speed-wise, PREF incurs additional training time caused by FFT but not in inference as we can pre-store the transformed features. Consequently, it is slightly slower in training but comparably fast in inference to SOTA. As opposed to a global FFT that transforms the entire phasor volume, a local window-based FFT, also known as short-time FFT, may leverage both frequency sparsity and spatial sparsity to further save memory and potentially accelerate training. Quality-wise, to maintain tiny volumes, PREF dilates along the reduced dimensions to cover the full spectra. As a result, it may introduce axial-align bias, similar to Mildenhall et al. (2020). To mitigate the problem, one may use NuFFT (Non-uniform Fast Fourier Transform) (Fessler & Sutton, 2003) to achieve non-uniform frequency sampling (Tancik et al., 2020) as well as fast Fourier transform.

6 CONCLUSION

The main motivation behind implicit neural representations is to search for *fast* algorithms to compute *compact* representation of functions and datasets. In this work, we extend the family of frequency representations by presenting PREF, a novel approach that combines neural networks and Fourier serials for compact representations. PREF remains *fast* by an approximated fast Fourier transform and retains *compact* by a tailored dilated phasor volume. PREF produces high-quality images, shapes, and radiance fields from given limited data and shows comparative results to the SOTAs. We also show our representation is more robust, potentially owing to its smooth and periodic properties that explore the correlations of data. In the future, we may apply our representation to efficient 3D-aware generative tasks, which may provide alternative frequency-based latent space for optimizations or controls.

Reproducibility. We provided the implementation details of each part in the main text and appendix, including data processing, parameter choices and devices. Source codes are available in supplementary materials.

REFERENCES

Matan Atzmon and Yaron Lipman. SAL: sign agnostic learning of shapes from raw data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

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

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. Tensorf: Tensorial radiance fields. *arXiv preprint arXiv:2203.09517*, 2022a.

Wanli Chen, Xinge Zhu, Ruoqi Sun, Junjun He, Ruiyu Li, Xiaoyong Shen, and Bei Yu. Tensor low-rank reconstruction for semantic segmentation. In *European Conference on Computer Vision*. Springer, 2020.

Zhiqin Chen, Thomas Funkhouser, Peter Hedman, and Andrea Tagliasacchi. Mobilenerf: Exploiting the polygon rasterization pipeline for efficient neural field rendering on mobile architectures. *arXiv preprint arXiv:2208.00277*, 2022b.

James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90), 1965.

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

J.A. Fessler and B.P. Sutton. Nonuniform fast fourier transforms using min-max interpolation. *IEEE Transactions on Signal Processing*, 51(2), 2003.

Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 2020.

Peter Hedman, Pratul P Srinivasan, Ben Mildenhall, Jonathan T Barron, and Paul Debevec. Baking neural radiance fields for real-time view synthesis. *arXiv preprint arXiv:2103.14645*, 2021.

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

Zhichun Huang, Shaojie Bai, and J Zico Kolter. Implicit^2: Implicit layers for implicit representations. *Advances in Neural Information Processing Systems*, 34, 2021.

Jon Jansen and Louis Bavoil. Fourier opacity mapping. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, 2010.

Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996.

David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020.

Renhao Liu, Yu Sun, Jiabei Zhu, Lei Tian, and Ulugbek Kamilov. Zero-shot learning of continuous 3d refractive index maps from discrete intensity-only measurements. *arXiv preprint arXiv:2112.00002*, 2021.

William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. *SIGGRAPH Computer Graphics*, 21(4), 1987.

Julien NP Martel, David B Lindell, Connor Z Lin, Eric R Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *arXiv preprint arXiv:2105.02788*, 2021.

Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*. Springer, 2020.

Claus Müller. *Spherical harmonics*, volume 17. 2006.

Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *arXiv preprint arXiv:2201.05989*, 2022.

Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 2020.

Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. *arXiv preprint arXiv:2104.10078*, 2021.

Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019.

Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*. Springer, 2020.

Charles Ruizhongtai Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, 2016.

Christian Reiser, Songyou Peng, Yiyi Liao, and Andreas Geiger. Kilonerf: Speeding up neural radiance fields with thousands of tiny mlps. *arXiv preprint arXiv:2103.13744*, 2021.

Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *arXiv*, 2020.

Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. *arXiv preprint arXiv:2111.11215*, 2021.

Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020.

Mattie Tesfaldet, Xavier Snelgrove, and David Vazquez. Fourier-cppns for image synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021a.

Peng-Shuai Wang, Yang Liu, Yu-Qi Yang, and Xin Tong. Spline positional encoding for learning 3d implicit signed distance fields. *arXiv preprint arXiv:2106.01553*, 2021b.

Daniel N Wood, Daniel I Azuma, Ken Aldinger, Brian Curless, Tom Duchamp, David H Salesin, and Werner Stuetzle. Surface light fields for 3d photography. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000.

Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. *arXiv preprint arXiv:2201.08845*, 2022.

Bangbang Yang, Chong Bao, Junyi Zeng, Hujun Bao, Yinda Zhang, Zhaopeng Cui, and Guofeng Zhang. Neumesh: Learning disentangled neural mesh-based implicit field for geometry and texture editing. *arXiv preprint arXiv:2207.11911*, 2022.

Guandao Yang, Serge Belongie, Bharath Hariharan, and Vladlen Koltun. Geometry processing with neural fields. *Advances in Neural Information Processing Systems*, 34, 2021.

Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Basri Ronen, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *Proc. NeurIPS*, 2020.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Alex Yu, Sara Fridovich-Keil, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. *arXiv preprint arXiv:2112.05131*, 2021a.

Alex Yu, Ruilong Li, Matthew Tancik, Hao Li, Ren Ng, and Angjoo Kanazawa. Plenoctrees for real-time rendering of neural radiance fields. *arXiv preprint arXiv:2103.14024*, 2021b.

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

Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020.

Ellen D Zhong, Tristan Bepler, Bonnie Berger, and Joseph H Davis. Cryodrgn: reconstruction of heterogeneous cryo-em structures using neural networks. *Nature methods*, 18(2), 2021.

Supplementary

A   Parsvel Regularization

Theorem A.1 (Differentiation Theorem) *Let* $\mathbf{F}(\mathrm{u}, \mathrm{v})$ *be an absolutely continuous differentiable function, and* $\mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})$ *be its inverse Fourier transform, we have*

$$\frac{\partial \mathrm{f}^n(\mathrm{x}, \mathrm{y}, \mathrm{z})}{\partial \mathrm{x}^n} = \mathcal{T}((\mathrm{j}2\pi\mathrm{u})^n \mathbf{F}(\mathrm{u}, \mathrm{v}, \mathrm{w})) \tag{9}$$

Theorem A.2 (Parsvel Theorem) *Let* $\mathbf{F}(\mathrm{u}, \mathrm{v})$ *be absolutely continuous differentiable function, and* $\mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})$ *be its inverse Fourier transform, we have*

$$\iint ||\mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})||^2 \mathrm{dxdy} = \sum_{\mathrm{u}, \mathrm{v}, \mathrm{w}} ||\mathbf{F}(\mathrm{u}, \mathrm{v}, \mathrm{w})||^2. \tag{10}$$

Lemma A.3 *Let* $\mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})$ *be integrable, and* $\mathbf{P}(\mathrm{u}, \mathrm{v}, \mathrm{w})$ *be its Fourier transform. The anisotropic TV loss of* $\mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})$ *can be represented by* $||2\pi\mathrm{u}\mathbf{P}||_2 + ||2\pi\mathrm{v}\mathbf{P}||_2 + ||2\pi\mathrm{w}\mathbf{P}||_2$

Proof: Recall the TV loss can be computed as $||\nabla_x \mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})||_2 + ||\nabla_y \mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})||_2 + ||\nabla_z \mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})||_2$. Since $\mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})$ and $\mathbf{P}(\mathrm{u}, \mathrm{v}, \mathrm{w})$ are Fourier pairs, we have Fourier transform preserves the energy of original quantity based on Parseval's theorem (theorem A.2), *i.e.*,

$$\iint ||\mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})||^2 \mathrm{dxdydz} = \sum_{\mathrm{i}, \mathrm{j}, \mathrm{k}} ||\mathbf{P}[\mathrm{i}, \mathrm{j}, \mathrm{k}]||^2. \tag{11}$$

According to theorem A.1, $\nabla_x \mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})$ and $\mathrm{j}2\pi\mathrm{u}\mathbf{P}(\mathrm{u}, \mathrm{v}, \mathrm{w})$ are also Fourier pairs. The integration derivative along axis x is defined as,

$$\iint ||\nabla_x \mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})||^2 \mathrm{dxdydz} = \sum_{\mathrm{i}, \mathrm{j}, \mathrm{k}} ||\mathrm{j}2\pi\mathrm{u}_\mathrm{i} \mathbf{P}[\mathrm{i}, \mathrm{j}, \mathrm{k}]||^2 \tag{12}$$

By taking square root on both sides, we have $||\nabla_x \mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})||_2 = ||2\pi\mathrm{u}\mathbf{P}(\mathrm{u}, \mathrm{v}, \mathrm{w})||_2$. And $||\nabla_y \mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})||_2 = ||2\pi\mathrm{v}\mathbf{P}(\mathrm{u}, \mathrm{v}, \mathrm{w})||_2$, $||\nabla_z \mathrm{f}(\mathrm{x}, \mathrm{y}, \mathrm{z})||_2 = ||2\pi\mathrm{w}\mathbf{P}(\mathrm{u}, \mathrm{v}, \mathrm{w})||_2$ can be derived similarly.

B   Phasorial Embedding Fields Implementation Details

Dilated Phasor Volume. Recall that PREF is a continuous embedding field corresponding to a multi-channel multi-dimensional square Fourier volume. We elaborate on implementation details. Let $\mathbf{P}[u, v, w]$ be a 3D phasor volume representing an embedding field $f(x, y, z)$.

Note that the $\mathbf{P}[u, v, w]$ is Hermitian symmetric when the $f(x, y, z)$ is a real-valued feature embedding, *i.e.*, $\mathbf{P}[u, v, w] = \mathbf{P}^*[-u, -v, -w]$ (i.e., its complex conjugate). Further, based on the observation that natural signals are generally band-limited, we model their corresponding fields with band-limited phasor volumes $\mathbf{P}[u, v, w]$ where we are able to partially mask out some entries and factor the volume along respective dimensions, as shown in Fig 1. Thus we factorize the full spectrum into tri-thin embeddings by reusing the linearity of the Fourier Transform, $f(x, y, z) = \mathcal{T}(\mathbf{P}_u) + \mathcal{T}(\mathbf{P}_v) + \mathcal{T}(\mathbf{P}_w)$.

IFT Implementation. Recall that for a 3D phasor volume, we approximate $\mathcal{T}$ via sub-procedures of applying 2D Fast Fourier Transforms (FFTs) and 1D numerical integration (NI) to achieve high efficiency. Therefore, given a batch of spatial coordinates, our PREF representation transforms them into a batch of feature embeddings in parallel where PREF can serve as a plug-and-play module. Such a module can be applied to many existing implicit neural representations to conduct task-specific neural field reconstructions. We present a sketchy PyTorch pseudo-code in Algorithm 1.
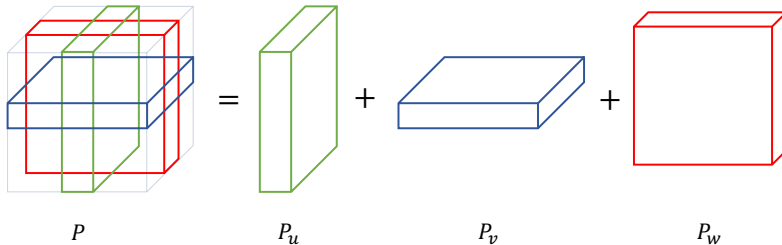
$$P \qquad P_u \qquad P_v \qquad P_w$$

Figure 9: Phasor volume decomposition. Let the phasor volume with zero frequency centered. We selectively mask out some entries of the phasor volume as zero, then approximate the spatial feature embedding of a large phasor volume in terms of the sum of the embedding from three smaller ones, each of which dilates along an axis. The complete transformation from phasor volume to spatial feature embeddings is shown in the PyTorch pseudo-code in Algorithm 1.

Algorithm 1 PREF Encoder in a PyTorch-like style.

```
import torch
import torch.nn as nn

class PREF(nn.Module):
    def _init_(self, res, d, ks):
        """
        res: resolution size
        d: reduced dim size
        ks: output kernel size
        """
        Nx, Ny, Nz = res
        # log sampling freq in reduced dimension
        self.freq = torch.tensor([0]+[2**i for i in torch.arange(d-1)])
        self.Pu = nn.Parameter(torch.zeros(1, ks, d, Ny, Nz).to(torch.complex))
        self.Pv = nn.Parameter(torch.zeros(1, ks, Nx, d, Nz).to(torch.complex))
        self.Pw = nn.Parameter(torch.zeros(1, ks, Nx, Ny, d).to(torch.complex))
    def forward(self, xyz):
        # 2D Fast Fourier Transform
        Pu = torch.fft.ifftn(self.Pu, dim=(3,4))
        Pv = torch.fft.ifftn(self.Pv, dim=(2,4))
        Pw = torch.fft.ifftn(self.Pw, dim=(2,3))
        # 2D Linear interpolation
        xs, ys, zs = xyz.chunk(3, dim=-1)
        Px = grid_sample_cmplx(Pu.transpose(3,3).flatten(1,2), torch.stack([zs, ys], dim=-1)[
            None]).reshape(Pu.shape[1], Pu.shape[2], -1)
        Py = grid_sample_cmplx(Pv.transpose(2,3).flatten(1,2), torch.stack([zs, xs], dim=-1)[
            None]).reshape(Pv.shape[1], Pv.shape[3], -1)
        Pz = grid_sample_cmplx(Pw.transpose(2,4).flatten(1,2), torch.stack([xs, ys], dim=-1)[
            None]).reshape(Pw.shape[1], Pw.shape[4], -1)
        # 1D Fourier Mapping
        fx = batch_NI(Px, xs, self.freq)
        fy = batch_NI(Py, ys, self.freq)
        fz = batch_NI(Pz, zs, self.freq)
        # Summation
        return fx+fy+fz
```

Phasor Volume Initialization. Our PREF approach can be alternatively viewed as a frequency space learning scheme for existing spatial coordinate-based MLPs. In our experiments, we found zero initialization works well for applications ranging from 2D image regression to 5D radiance field reconstruction while certain applications require more tailored initialization, e.g., geometric initialization in (Atzmon & Lipman, 2020). This is because $\mathbf{P}(k)$ (with k being the frequency coordinate) needs to satisfy the unique constraints of $f(x)$. We thus initialize the phasor volume as follows: Let $f^{\circ}(x)$ be the initialization of $f(x)$. We have $\mathbf{P}^{\circ}(k) = \mathcal{T}(f^{\circ}(x))$, with $\mathcal{T}$ as the Fourier transform. We then transform $\mathbf{P}^{\circ}(k)$ via the inverse Fourier transform $\mathcal{T}$ (due to duality between $f(x)$ and $\mathbf{P}(k)$) as the *approximation* to $f^{\circ}(x)$. We found such a strategy enhances stability and efficiency.

Computation Time. One of the key benefits of PREF is its efficiency. We conduct frequency-based neural field reconstruction by employing IFT, which is computationally low cost and at the same time effective. When the input batch is sufficiently large (e.g., 4096×1024 samples per batch in

	Chair	Drums	Ficus	Hotdog	Lego	Materials	Mic	Ship	Mean	Size (MB)↓
PlenOctrees (Yu et al., 2021b)	34.66	25.37	30.79	36.79	32.95	29.76	33.97	29.62	31.71	1976.3
Plenoxels (Yu et al., 2021a)	33.98	25.35	31.83	36.43	34.10	29.14	33.26	29.62	31.71	778.1
DVGO (Sun et al., 2021)	34.09	25.44	32.78	36.74	34.46	29.57	33.20	29.12	31.95	612.1
Ours	34.95	25.00	33.08	36.44	35.27	29.33	33.25	29.23	32.08	34.4

Table 4: PSNR results on each scene from the Synthetic-NeRF dataset (Mildenhall et al., 2020). We show the comparisons of the dense volume variants with our PREF (frequency-based scheme).

radiance field reconstruction), the per-sample numerical evaluation will dominate the computational cost. Since such per-sample evaluation can be efficiently implemented using matrix products, it is essentially equivalent to adding a tiny linear layer. The overall implementation makes PREF nearly as fast as the state-of-the-art, e.g., instant-NGP for NeRF. For example, on the Lego example, our PyTorch PREF produces the final result in 16 minutes on a single RTX3090, considerably faster than the original NeRF and comparable to the PyTorch implementation of NGP. We are in the process of implementing PREF on CUDA analogous, and hopefully, it may achieve comparable performance to the CUDA version of NGP.

C IMAGE REGRESSION

Image regression and completion. To evaluate PREF's robustness, we demonstrate PREF on image completion tasks. We use the commonly adopted setting (Huang et al., 2021; Fathony et al., 2020): given 25% pixels of an image, we set out to predict another 25% pixels. We evaluate PREF vs. SOTA on two benchmark datasets - Nature and Text. Specifically, we compare PREF with a dense grid counterpart, and two state-of-the-art coordinate-based MLPs (Mildenhall et al., 2020; Sitzmann et al., 2020). The dense grid uses a $8 \times 100 \times 100$ resolution whereas PREF uses two 20×9 grids that correspond to the highest frequency of $2^7 = 128$. The two embedding techniques above use the same MLP with three linear layers, 256 hidden dimensions, and ReLU activation. We use Positional Encoding (PE) which consists of a 5-layer MLP with 7 frequencies encoding. We adopt SIREN from (Sitzmann et al., 2020) that uses a 4-layer MLP and sine activation. Detailed comparisons are listed in Tab. 3.

Optimization details. All experiments use the same training configuration. Specifically, we adopt the Adam optimizer (Kingma & Ba, 2015) with default parameters ($\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e^{-8}$), a learning rate of $1e^{-4}$. We use L_1 loss with $15k$ iterations to produce the final results.

D LEVEL OF DETAIL FILTERING

Recall that the continuous embedding field of PREF is synthesized from a phasor volume under various frequencies. Therefore, thanks to Fourier transforms, various tools such as convolution in the continuous embedding fields can be conveniently and efficiently implemented as multiplications. This is therefore a unique advantage of PREF compared with its spatial embedding alternatives (Chen et al., 2020; Sun et al., 2021; Yu et al., 2021a; Müller et al., 2022).

Let $\mathcal{M}(; \theta)$ and $\mathbf{P}[u, v, w]$ be the optimized MLP and phasor volume, respectively. $\mathcal{T}$ represents the inverse Fourier Transform. Recall that we obtain a reconstruction field by $\Phi(x) = \mathcal{M}(\mathcal{T}(\mathbf{P}; x); \theta)$. Modification to the original signal via convolution-based filtering can now be derived as:

$$\Phi^*(x) = \mathcal{M}(\mathcal{T}(\mathbf{P} \circ G; x); \theta) \tag{13}$$

where $\circ$ denotes element-wise multiplication and $G \in \mathbb{C}^{l \times N^3}$ is a filter.

Now, we explore how to manipulate $\Phi(x)$ via the optimized phasor volume $\mathbf{P}$ and kernel G. For simplicity, we only the Gaussian filter G while more sophisticated filters can also be applied in the same. Assume

$$G(k) = \exp(-k^T k \sigma^2), \tag{14}$$

where $k = [u/N, v/N, w/N]$ and G covers the complete frequency span of $\mathbf{P}$; that is, we can scale the magnitude of phasor features frequency-wise. For example, by varying the Gaussian kernel size using σ, PREF can denoise the neural representation of the signal at different scales.