G^2fR: Frequency Regularization in Grid-based Feature Encoding Neural Radiance Fields

Shuxiang Xie[1,2], Shuyi Zhou[1], Ken Sakurada[2,3], Ryoichi Ishikawa[1], Masaki Onishi[2], and Takeshi Oishi[1]

[1] The University of Tokyo, Tokyo, Japan
[2] The National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
[3] Kyoto University, Kyoto, Japan

Abstract. Neural Radiance Field (NeRF) methodologies have garnered considerable interest, particularly with the introduction of grid-based feature encoding (GFE) approaches such as Instant-NGP and TensoRF. Conventional NeRF employs positional encoding (PE) and represents a scene with a Multi-Layer Perceptron (MLP). Frequency regularization has been identified as an effective strategy to overcome primary challenges in PE-based NeRFs, including dependency on known camera poses and the requirement for extensive image datasets. While several studies have endeavored to extend frequency regularization to GFE approaches, there is still a lack of basic theoretical foundations for these methods. Therefore, we first clarify the underlying mechanisms of frequency regularization. Subsequently, we conduct a comprehensive investigation into the expressive capability of GFE-based NeRFs and attempt to connect frequency regularization with GFE methods. Moreover, we propose a generalized strategy, G^2fR: Generalized Grid-based Frequency Regularization, to address issues of camera pose optimization and few-shot reconstruction with GFE methods. We validate the efficacy of our methods through an extensive series of experiments employing various representations across diverse scenarios.

Keywords: Implicit neural representation · Grid-based NeRF · 3D scene reconstruction

1 Introduction

The growing interest in Neural Radiance Field (NeRF) [24] has triggered extensive investigation into the implicit neural representation (INR) of 3D scenes across various research fields. Especially, the introduction of grid-based feature encoding (GFE) methods, such as Instant-NGP [25], Zip-NeRF [2], TensoRF [3], and Tri-MipRF [13], further enhances the performance of NeRF-related techniques in various tasks. Unlike conventional positional encoding (PE) in MLP-based NeRF, these methods use a 3D spatial grid to preserve local features and transform the input coordinates into feature space by interpolating neighboring grid points. Therefore, GFE-based methods exhibit different behavior from

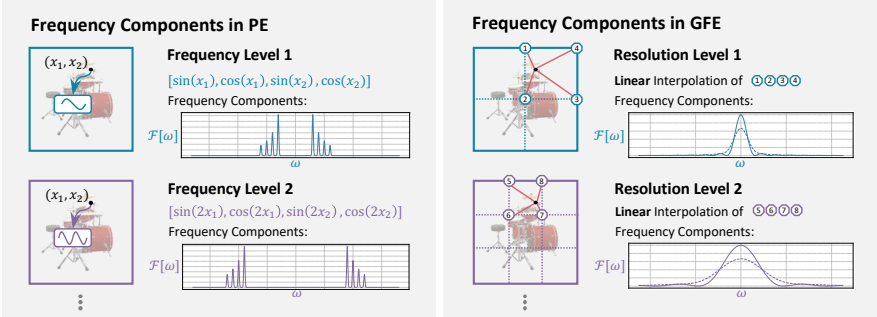


Fig. 1: General overview of the concept. **Left:** an analysis of expressive power in the frequency domain for PE NeRFs, a topic extensively explored and deliberated upon in existing literature [44]. **Right:** extending this analysis to GFE NeRFs. According to our study, there is a similar conclusion in GFE, but the determinative factor for frequency bands is the grid resolution.

PE-based methods because of the heterogeneity of the feature space. Moreover, methods such as NeuRBF [6] achieve good reconstruction performance without the need for spatial grids.

While many traditional tasks have benefited from the aforementioned methods, challenges still remain for NeRF-related methods, such as the need for accurate camera poses and large amounts of training data. Several effective strategies have been proposed for PE-based NeRF to address these challenges. Regarding the camera pose problem, many studies propose leveraging the differentiable nature of NeRF to integrate camera parameters into the computational graph [16, 42]. Concerning the issue of limited input, many works suggest employing pre-trained models to estimate and predict information for unseen areas, yielding promising performance [4, 15, 43].

Among the many existing strategies, frequency regularization emerges as a prevailing solution to address these challenges. The coarse-to-fine optimization scheduler applied in [18] avoids the local minima encountered during camera pose optimization. The progressively expanding mask on the positional encoded vectors [41] governs the frequency components directly from the input, thereby achieving novel view synthesis within a few-shot setting. These methodologies emphasize the importance of frequency regularization across diverse tasks without introducing extra computational overhead. We can explain and comprehend the promising results brought by this technique with the insights into the expressive capability of MLP-based INR [44]. Despite similar strategies being employed in GFE methods, demonstrating their beneficial impact [11, 17, 27, 38], their underlying mechanisms, necessity, and functionality in discrete grids remain inadequately addressed from a theoretical standpoint.

Therefore, in this study, we provide a detailed validation of the GFE methods with the aim of elucidating the expressive capability of GFE through Fourier analysis. Fig. 1 gives an overview of our concept. Subsequently, we revisit the

camera pose optimization and few-shot NeRF problem and delve into the theoretical support of the frequency regularization approach [18,41]. We show that by applying G^2fR: generalized grid-based frequency regularization, one may overcome the challenges associated with local minima in GFE NeRFs. Furthermore, we undertake extensive experiments to fortify our theoretical proposition.

Our contributions are summarized as follows:

- We offer a theoretical analysis of GFE NeRFs, shedding light on their expressive capability through their behavior in the frequency domain.
- We demonstrate that the supported frequency bands depend on the resolution of the grids and additionally propose G^2fR to extend frequency regularization concept to GFE NeRFs.
- We elucidate the significance of the generalization, which further supports the validity and necessity of frequency regularization.

2 Related Work

2.1 NeRF Related Implicit Neural Representations

NeRF stands out as a widely embraced INR example in the context of novel view synthesis [24]. NeRF employs a fully connected MLP for scene representation, which is used to retrieve color and density information for each input point. Then, volume rendering is applied to integrate the queried colors along camera rays, yielding estimated colors corresponding to pixels. This straightforward pipeline for dense reconstruction positions NeRF-related INRs as favorable choices for tasks such as SLAM and 3D reconstruction [23,26,40,46,48]. The implicit nature of INRs also enables easier approaches for handling information from different types of sensors [8, 12, 14, 47].

2.2 Encoding Methods in NeRFs

The encoding methods for input coordinates in NeRFs can be broadly categorized into two types: positional encoding (PE) and grid-based feature encoding (GFE).

PE typically maps the input coordinates to higher-frequency contents before feeding them into an MLP [29, 30, 34]. As extensively studied by various researchers [24, 34, 44], PE stands out as a cornerstone of NeRF's success. Further investigations emphasize the significance of the rank of the embedding matrix concerning the encoding function [45]. Generally speaking, PE methods offer the advantage of generating compact models while preserving high-frequency details. However, a significant drawback is the typically long training time.

GFE methods explicitly generate grid maps in which the grid points encapsulate local features, then utilize linear interpolation to obtain the feature located at the input coordinate [3, 13, 25]. GFE methods usually have better memorization ability and can converge faster; nonetheless, they cost larger memory [9, 33]. Hence, there's a growing interest in relatively memory-efficient techniques. Instant-NGP employs a hash mapping technique to effectively reduce the

memory requirement down to the size of the hash table [25]. TensoRF and Tri-Miprf are inspired by tensor decomposition and represent the high-dimensional feature tensors as the outer product of 1D vectors and 2D planes [3, 13]. [37] introduces a hybrid encoding method, which leverages PE for coarse components and integrates GFE to capture high-frequency details.

2.3 NeRF with Camera Pose Optimization

Requirement for accurate camera pose inputs remains one of the the major issues of NeRF. Including camera pose parameters in the training is the most straightforward yet effective approach [16, 32, 39, 42]. To avoid local minima in the training process, many methods are proposed, including more complex representations for camera poses [5, 20], and control of PE inputs [10, 18], which can be considered as a form of *frequency regularization*. Chng *et al*. highlight the effect of applying Gaussian-based activation function, suggesting PE with Fourier contents might not be necessary [7].

This task is also feasible for GFE NeRFs. The rapid convergence of GFE facilitates the utilization of Monte Carlo approaches, effectively enhancing robustness [19]. Heo *et al*. [11] emphasize the impact of smooth gradient with respect to interpolation methods inside grids. Furthermore, Park *et al*. [27] use preconditioners to enable smoother and more robust optimization of camera intrinsic and extrinsic parameters.

2.4 Few-Shot NeRF

Another primary issue for NeRF is the requirement for a large number of input images. Many efforts have also been made targeting the Few-Shot reconstruction task. Conventional approaches typically use pre-trained models to extrapolate observed information to unobserved regions. These approaches commonly apply CNN models [4, 43] and transformers [35, 36] to extract latent features from input images. Then, they employ a learnable volume rendering technique combined with a feature decoder for color synthesis. Jain *et al*. [15] utilize a CLIP-based Vision Transformer to ensure semantic consistency between reference and query views. Moreover, *frequency regularization* is also applicable to this issue. Yang *et al*. [41] have proposed a method that masks high-frequency parts in positional encoded vectors to ensure low-frequency components in the scene are properly learned. This work makes few-shot reconstruction possible even without introducing additional pre-trained models.

It is noteworthy that frequency regularization methods have proven effective in addressing both issues. Although some studies have tried similar strategies in GFE methods [11, 17, 27, 38], the theoretical background of frequency regularization within grids remains insufficiently explored. Therefore, in our study, we closely investigate GFE methods from the fundamental essence, aiming to bridge the existing knowledge in PE NeRFs with our findings.

3 Expressive Analysis of GFE NeRFs

This section begins with an introduction to some fundamental concepts of NeRF. Then we shift our attention towards formulating the GFE NeRFs utilizing linear interpolation. Subsequently, we delve into an examination of the expressive capabilities of the GFE NeRFs. Our emphasis lies in explaining the distribution of frequency components of GFE methods and clarifying the connections and distinctions with PE method.

3.1 Preliminary: Neural Radiance Fields

We will first briefly introduce the basic concepts of NeRF. NeRF takes multiple images as input to generate an INR for a scene to synthesize novel views. Since NeRF applies volume rendering [21], the density $\sigma \in \mathbb{R}$ and color $\mathbf{c} \in \mathbb{R}^3$ at each point $\mathbf{x} \in \mathbb{R}^3$ must be modeled. Accordingly, we can denote the neural representation as $f_{\boldsymbol{\Theta}} : \mathbf{x} \rightarrow [\mathbf{c}, \sigma]^\top$, where $\boldsymbol{\Theta}$ signifies the trainable parameters. Generally speaking, the input of NeRF also contains a view direction part, but we omit this part here for brevity.

Once a pixel coordinate $\mathbf{u} \in \mathbb{R}^2$ and the camera pose $\mathbf{p}$ are given, we can estimate the color of $\mathbf{u}$ by volume rendering. Let z_i be the depth of i-th sampling point along the ray, and we obtain the 3D world coordinates of the point with a warping function as $\mathbf{x}_i = \mathcal{W}(\mathbf{u}, \mathbf{p}, z_i)$. The warping function $\mathcal{W}(\cdot)$ also includes the camera intrinsic parameters. Then, we can estimate the color $\widehat{\mathcal{I}}(\mathbf{u}, \mathbf{p})$ as:

$$\widehat{\mathcal{I}}(\mathbf{u}, \mathbf{p}) = \sum_{i=1}^{N} \exp\left(- \sum_{j=1}^{i} \sigma(\mathbf{u}, z_j) \right) \cdot \sigma(\mathbf{x}_i) \cdot \mathbf{c}(\mathbf{x}_i). \tag{1}$$

We assume the total number of input images as I, the total amount of pixels for each image as J. Then we optimize the parameters of $f_{\boldsymbol{\Theta}}$ using the input color $\mathcal{I}(\mathbf{u})$, as follows:

$$\boldsymbol{\Theta} = \arg\min_{\boldsymbol{\Theta}} \sum_{i=1}^{I} \sum_{j=1}^{J} \| \widehat{\mathcal{I}}(\mathbf{u}_j, \mathbf{p}_i; \boldsymbol{\Theta}) - \mathcal{I}_i(\mathbf{u}_j) \|_2^2. \tag{2}$$

3.2 Formulation of GFE methods

Hereby, we formulate the GFE function using linear interpolation of local features. We start with general INRs. Consistent with prior research [31, 44], we assume the INR architecture is composed of an encoding function $\Psi(\mathbf{x})$ and an MLP. Let the input and output dimensions of l-th layer of the MLP be F_{l-1} and F_l, respectively. The total number of layers is L. The l-th layer is described by the weights $\mathbf{W}^{(l)} \in \mathbb{R}^{F_l \times F_{l-1}}$ and biases $\mathbf{b}^{(l)} \in \mathbb{R}^{F_l}$, with the element-wise non-linear activation function $\rho^{(l)}(\cdot)$ applied between layers. The overall network

$f_{\mathbf{\Theta}}$ is expressed as:

$$\mathbf{z}^{(0)} = \Psi(\mathbf{x}), \tag{3}$$

$$\mathbf{z}^{(l)} = \rho^{(l)}\left(\mathbf{W}^{(l)}\mathbf{z}^{(l-1)} + \mathbf{b}^{(l)}\right), l = 1, 2, \ldots, L-1, \tag{4}$$

$$f_{\mathbf{\Theta}}(\mathbf{x}) = \mathbf{W}^{(L)}\mathbf{z}^{(L-1)} + \mathbf{b}^{(L)}. \tag{5}$$

We take the multi-resolution hash encoding [25] with a total of $M \in \mathbb{N}$ levels as an example. Here, one may understand the term *resolution* as the number of grids at certain level. $\psi_m(\mathbf{x}) \in \mathbb{R}^q$, $q \in \mathbb{N}$, signifies the mapped feature at level $m \in \mathbb{N}$ and $m \leq M$ for a normalized input $\mathbf{x}$. The encoded feature $\Psi(\mathbf{x}) \in \mathbb{R}^{qM}$ is expressed as:

$$\Psi(\mathbf{x}) = [\psi_1(\mathbf{x})^\top, \ldots \psi_m(\mathbf{x})^\top, \ldots \psi_M(\mathbf{x})^\top]^\top. \tag{6}$$

Let the resolution at level m be $s_m \in \mathbb{N}$. Define the feature in the t-th grid ($t \in \mathbb{Z}$ and $t \in [0, s_m - 1]$) at the m-th level as $h_{(t;m)} \in \mathbb{R}^q$. In GFE methods, features like $h_{(t;m)}$ are stored discretely at grid points, necessitating linear interpolation to acquire features for arbitrary point inputs. Hence, $\psi_m(\cdot)$ is a linearly interpolated function of features. For illustrative purposes, we consider the situation of 1D input and 1D output, where $\mathbf{x} = x \in \mathbb{R}$ and $q = 1$. Consider a normalized input $x \in [0, 1]$, $\frac{t-1}{s_m-1} \leq x \leq \frac{t}{s_m-1}$, where t and $t-1$ represents two neighboring indices of the grids, then $\psi_m(x)$ can be written as,

$$\psi_m(x) = (s_m - 1)\left(h_{(t;m)} - h_{(t-1;m)}\right) \cdot x + (1 - t) \cdot h_{(t;m)} + t \cdot h_{(t-1;m)}. \tag{7}$$

To facilitate easier understanding, we transform Eq. (7) into a simpler form. Let us consider a 1D triangular pulse function $\Lambda(\cdot)$ defined as follows (see Fig. 2a):

$$\Lambda(x) = \max(0, 1 - |x|). \tag{8}$$

It is evident that a linearly interpolated function can be represented as a summation of triangular pulse functions as demonstrated in Fig. 2b. Therefore we can rewrite Eq. (7) as follows:

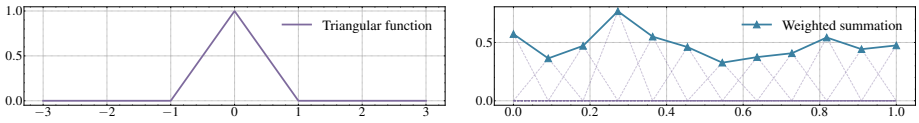
$$\psi_m(x) = \sum_{t=0}^{s_m-1} h_{(t;m)} \cdot \Lambda((s_m - 1)x - t). \tag{9}$$

It is effortless to extend this formulation to n-dimensional grids using n-dimensional triangular functions: $\Lambda_{(n)}(\mathbf{x}) = \prod_{i=1}^{n} \Lambda(x_i)$.

Up to this point, the learnable parameters $\mathbf{\Theta}$ of the whole INR as listed in Eq. (2) can be expressed as follows: $\mathbf{\Theta} = \{h_{(t;m)} | t \in [0, s_m - 1], m \in [1, M]\} \cup \{\mathbf{W}^{(l)}, \mathbf{b}^{(l)} | l \in [1, L]\}$.

3.3 Expressive Power of GFE NeRFs

Our objective is to clarify the expressive capacity of GFE NeRF through a similar approach demonstrated by [44], which comprehensively demonstrates the expressive power of PE INRs.



(a) Appearance of single 1D triangular pulse function $\Lambda(x)$.

(b) 1D Linear interpolation can be represented as summation of $\Lambda(\cdot)$ functions.

Fig. 2: Connections between linear interpolation and triangular pulse function $\Lambda(\cdot)$.

Expressive Power of PE INRs According to [44], the expressive power of PE INRs can be written as a linear combination of harmonics of Fourier features. Let $f_{\mathbf{\Gamma}}$ be a PE INR that takes $\mathbf{x}$ as input, when subjected to a polynomial approximation of the activation functions, $f_{\mathbf{\Gamma}}$ can be fully expressed as:

$$f_{\mathbf{\Gamma}}(\mathbf{x}) = \sum_{\omega \in \mathcal{H}(\mathbf{\Omega})} c_\omega \sin(\langle \omega, \mathbf{x} \rangle + \phi_\omega). \tag{10}$$

Here $\langle \cdot, \cdot \rangle$ represents the dot product; $\mathcal{H}(\mathbf{\Omega})$ represents the set of available frequencies; coefficients c_ω and ϕ_ω are learnable parameters. The size of $\mathcal{H}(\mathbf{\Omega})$ grows exponentially with the depth of the MLP network. If we apply Fourier transform to Eq. (10), we can find that in frequency domain, $f_{\mathbf{\Gamma}}$ is characterized by a series of Dirac δ functions, which corresponds to Fig. 1 (left).

Next, we focus on explaining the expressive power of GFE NeRF. According to our study, the frequency range of GFE NeRF also expands as the network goes deeper. However, in GFE cases, the network is typically shallow thus the influence from network depth is not significant. Additionally, we argue that the frequency range is almost solely related to the resolution of the grids.

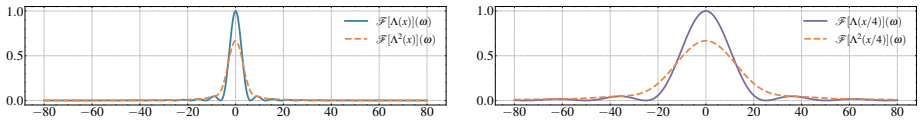
We discuss about the most straightforward scenario, 1D input and 1D output, where $L = 1$, $M = 1$, $F_0 = F_1 = 1$. Complicated cases can be handled in a similar way. Denote resolution of the only level $s_1 = s$, we can ascertain the following:

$$\mathbf{z}^{(0)} = \Psi(x) = \sum_{t=0}^{s-1} h_{(t;1)} \cdot \Lambda((s-1)x - t), \tag{11}$$

$$f_{\mathbf{\Theta}}(x) = \mathbf{z}^{(1)} = \rho^{(1)} \left(\sum_{t=0}^{s-1} \mathbf{W}^{(1)} \cdot h_{(t;1)} \cdot \Lambda((s-1)x - t) + \mathbf{b}^{(1)} \right). \tag{12}$$

We approximate $\rho(\cdot)$ using a polynomial such that $\rho^{(1)}(z) = \sum_{k=0}^{K} \alpha_k z^k$ [44]. Then we can show that $f_{\mathbf{\Theta}}(x)$ can be approximated by a series of functions $\Lambda^k(\cdot)$. Let the new coefficients for $\Lambda^k((s-1)x - t)$ be $h'_{(t;1),k}$ which can be calculated using α_k and $h_{(t;1)}$. Making C a constant term, $f_{\mathbf{\Theta}}(x)$ becomes:

$$f_{\mathbf{\Theta}}(x) = C + P(x) + \sum_{k=0}^{K} \sum_{t=0}^{s-1} h'_{(t;1),k} \cdot \Lambda^k((s-1)x - t). \tag{13}$$



(a) Plot for $\mathcal{F}[\Lambda(x)](\omega)$ and $\mathcal{F}[\Lambda^2(x)](\omega)$. **(b)** Plot for $\mathcal{F}[\Lambda(\frac{x}{4})](\omega)$ and $\mathcal{F}[\Lambda^2(\frac{x}{4})](\omega)$.

Fig. 3: Left: The bandwidth of $\Lambda^2(x)$ in frequency domain, indicated by the dashed line, does become larger compared to that of $\Lambda(x)$. **Right:** We directly scale x by $\frac{1}{4}$. If we compare the bandwidth of $\Lambda^2(x)$ and $\Lambda(\frac{x}{4})$, we may find the increase brought by larger order k in $\Lambda^k(\cdot)$ is not that apparent.

$P(x)$ here stands for the product between neighboring triangular pulses. Given that the scale of $P(x)$ is significantly small than $\Lambda^k((s-1)x-t)$ and thus contribute less, and also due to the limited space, we will focus our discussion on the latter term. Note that the coefficients of the polynomial used in the approximation usually decay very rapidly as the order k increases, especially for the ReLU activation function [22, 44]. Consequently, when $k > 2$, coefficients $h'_{(t;1),k}$ tend to take extremely small value.

We further apply Fourier transform to Eq. (13). Here, we use ξ to absorb the relatively higher-order components when $k > 2$ and Fourier transformation of $P(x)$ part. Then, due to the linearity of Fourier transform, we can obtain,

$$\begin{aligned}\mathcal{F}_{\boldsymbol{\Theta}}(\omega) &= \sum_{k=1}^{2}\sum_{t=0}^{s-1} h'_{(t;1),k}\cdot\mathcal{F}_x\left[\Lambda^k((s-1)x-t)\right]+\xi \\ &= \sum_{t=0}^{s-1} h'_{(t;1),1}\cdot(s-1)\cdot e^{-j\omega\frac{t}{s-1}}\cdot\frac{2-2\cos(\frac{\omega}{s-1})}{\omega^2} \\ &\quad + \sum_{t=0}^{s-1} h'_{(t;1),2}\cdot(s-1)^2\cdot e^{-j\omega\frac{t}{s-1}}\cdot\frac{4\frac{\omega}{s-1}-4\sin(\frac{\omega}{s-1})}{\omega^3}+\xi.\end{aligned} \tag{14}$$

It is apparent that the Fourier transform of Eq. (10) will yield a composite of multiple Dirac δ functions in the frequency domain (see Fig. 1). The result shown by Eq. (14) emphasizes a key distinction between PE and GFE methods: while the PE structure produces discrete frequency components, GFE can yield a continuous frequency distribution as described in Eq. (14).

Bandwidth of Single Pulse Let us consider $\mathcal{F}[\Lambda(x)](\omega) = \frac{2-2\cos(\omega)}{\omega^2}$ and $\mathcal{F}[\Lambda^2(x)](\omega) = \frac{4\omega-4\sin(\omega)}{\omega^3}$. Obviously, we can see $\exists\eta_1,\eta_2\in\mathbb{R}$, and $\eta_1,\eta_2>0$ that make $\mathcal{F}[\Lambda(x)](\omega)\le\eta_1\cdot\frac{1}{\omega^2}$ and $\mathcal{F}[\Lambda^2(x)](\omega)<\eta_2\cdot\frac{1}{\omega^2}$. This implies that as the frequency ω increases, the magnitude of the frequency components for both $\Lambda(x)$ and $\Lambda^2(x)$ tends to approach zero. Similar conclusions can be inferred for larger values of k. As a result, increasing k will only slightly contribute to expanding the bandwidth. Fig. 3a provides a visualization of this process. With the same concept, one may easily extend the conclusion to the discussion of $P(x)$ and its corresponding Fourier transformation.

Bandwidth and Resolution As evident from Eq. (14), a smaller resolution s leads to a narrower frequency range, and conversely, a higher resolution results in a broader bandwidth. Fig. 3b provides an illustration of this point by setting a 4-times larger resolution. Considering the quickly diminishing factor α_k and shallow network depth, the frequency bandwidth is almost entirely determined by the resolution parameter s, which implies that we may regularize the frequency components by controlling the resolution of the grids.

4 Frequency Regularization in GFE NeRFs

In this section, we propose one potential choice of frequency regularization technique G^2fR in GFE NeRFs, extending the idea from existing works in PE cases. We further discuss the importance of model generalization in both camera pose optimization and few-shot reconstruction tasks, which supports the validity of G^2fR from an alternative perspective.

4.1 From PE to GFE

Recall that in prior work [18,41], a mask was proposed to apply to positional encoding across different frequency bands. A clearer understanding of this masking technique emerges from the insights [28, 44]: the supported frequency range extends by gradually integrating higher-frequency encoded elements into the input. This masking technique can be seen as a type of *Frequency Regularization*.

We aim to extend the concept of frequency regularization to GFE NeRFs. Firstly, the encoding function $\Psi(\mathbf{x})$ should include multiple resolution levels. Multi-resolution hash encoding [25] serves as a good example, as illustrated in Eq. (6). The rationale behind this requirement becomes evident: employing multiple resolution levels facilitates accommodating various bandwidths. Subsequently, we apply a similar mask as used in the PE cases [18, 41] to the feature mapping functions $\psi_m(\mathbf{x})$ at different levels. We modify Eq. (6) as follows:

$$\Psi(\mathbf{x}) = [w_1(\tau)\cdot\psi_1(\mathbf{x})^\top, \dots w_i(\tau)\cdot\psi_i(\mathbf{x})^\top, \dots w_M(\tau)\cdot\psi_M(\mathbf{x})^\top]^\top, \tag{15}$$

where $w_i(\tau) \in [0,1]$. $\tau \in [0, M]$ is a parameter indicating the optimization progress. Let the total iteration number be T, and the first βT iterations $\beta \in (0,1)$, denote the duration for frequency regularization. For iteration $t \in [0, T]$, we define $\tau = M \cdot \min(\frac{t}{\beta T}, 1)$. Without specific design, we follow the style of BARF [18]: We fix $w_1(\tau) = 1$ and for $i \geq 2$ define $w_i(\tau) = \frac{1}{2}[1 - \cos((\tau - i)\pi)]$ when $i \leq \tau < i+1$, $w_i(\tau) = 0$ when $\tau < i$, and $w_i(\tau) = 1$ when $\tau \geq i+1$. We notice that many studies [11, 17, 27] apply binary masks instead of smooth ones as introduced above. Empirically, we find that there was no significant difference between binary and smooth masking techniques, since both of them manage to control the frequency components.

We call this method G^2fR: generalized grid-based frequency regularization, since it is not limited to hash encoding and can be extended to other GFE methods. Certainly, G^2fR is not the only approach. For example, in [13], a series of real-time computed Mipmaps can serve the same purpose.

4.2 Model Generalization with G^2*f*R

We next focus on discussing how G^2fR performs in terms of the model's ability to generalize. We clarify this point by analyzing specific tasks and establishing a connection between generalization capability and frequency regularization through the concept of *the Lipschitz Constant*.

Model Generalization in Different Tasks The following two notable tasks in NeRF are greatly related to model generalization:

1. Camera poses optimization: Denoting the camera parameters at iteration i as $\mathbf{p}_i$, at iteration $i+1$, the camera parameters evolve to $\mathbf{p}_{i+1} = \mathbf{p}_i + \Delta\mathbf{p}$, implying that rendering always occurs from novel viewpoints. Therefore, model generalization is significant, as by changing viewpoints, we might observe regions lacking supervision data.
2. Few-Shot NeRF: When the input viewpoints of NeRF are sparse, the information for reconstion may be insufficient. Thus, it becomes important to *generalize* the observed data to encompass the unobserved regions.

In the event of poor generalization, there exists a risk of overfitting and stumbling upon random local minima. Results from [18, 41] demonstrate this points.

The Lipschitz Constant Generalization cannot be evaluated without detailed context. One idea is that the model ought to learn a function within a space constrained by certain priors, ensuring that the function's value does not exhibit drastic variations. Thus we can discuss the problem in terms of *Lipschitz continuous*. Recall that for any function $f : \mathbb{R}^n \to \mathbb{R}^m$, if $\forall x, y$ within its domain and $x, y \in \mathbb{R}^n$, $\exists L \geq 0, L \in \mathbb{R}$, such that

$$\|f(x) - f(y)\|_i \leq L_f \|x - y\|_i, \tag{16}$$

f can be considered as *Lipschitz continuous* and the smallest such bound L_f is called as *the Lipschitz constant* for f. Usually i takes the value of 2 or ∞. An intuitive interpretation of this theory suggests that a smaller value of the Lipschitz constant corresponds to a reduced variation in the function's output, thereby indicating a more robust and generalizable model, and vice versa.

We now turn our attention to Eq. (15). Considering random input $\mathbf{x}$ and $\mathbf{y}$ within the domain for $\Psi(\cdot)$, we take L_2 norm in Eq. (16) and then have

$$L_\Psi = \sup \left(\frac{1}{\|\mathbf{x} - \mathbf{y}\|_2} \cdot \sqrt{\sum_{m=1}^{M} w_m^2 \cdot \langle \psi_m(\mathbf{x}) - \psi_m(\mathbf{y}), \psi_m(\mathbf{x}) - \psi_m(\mathbf{y}) \rangle} \right), \tag{17}$$

where w_m follows the definition in Sec. 4.1, $\langle \cdot, \cdot \rangle$ stands for dot product. L_Ψ value tends to grow larger as the optimization moves on, since $\{w_m\}$ indicates a mask that includes more and more levels into calculation. Therefore the whole system will have better generalization capability at the beginning phase, which corresponds to the learning process of the low-frequency components as discussed in Sec. 3.3.

Scene	Camera pose registration										View synthesis quality				
	Rotation error (°) ↓					Translation error ↓ ($\times 10^2$)					PSNR ↑				
	NGP w/	NGP w/o	Mtrf w/	Mtrf w/o	BARF	NGP w/	NGP w/o	Mtrf w/	Mtrf w/o	BARF	NGP w/	NGP w/o	Mtrf w/	Mtrf w/o	BARF
Chair	0.101	0.580	**0.048**	0.053	0.124	0.514	2.549	**0.184**	0.267	0.470	**33.98**	32.86	31.78	31.80	30.98
Drums	**0.028**	0.131	0.031	0.032	0.047	**0.149**	0.193	0.164	0.240	0.314	**24.98**	24.60	22.95	22.53	23.88
Ficus	**0.063**	0.410	0.069	0.067	0.103	**0.274**	1.593	0.289	0.431	0.671	**26.68**	25.30	26.31	26.41	25.79
Hotdog	0.164	0.203	**0.099**	0.183	0.134	0.891	1.307	**0.365**	0.489	2.331	**36.28**	35.35	34.79	34.86	33.91
Lego	**0.049**	1.181	0.051	0.354	0.074	**0.167**	4.361	0.178	0.933	0.430	**32.53**	31.33	28.72	27.54	28.03
Materials	**0.045**	2.234	0.459	1.168	1.023	**0.189**	9.715	3.031	4.762	3.113	**27.76**	23.22	27.43	27.03	25.03
Mic	**0.042**	0.965	0.043	0.556	0.392	0.139	2.627	**0.164**	2.104	0.411	**33.30**	30.02	33.10	32.83	30.87
Ship	0.729	1.475	**0.513**	0.540	0.988	1.426	3.048	0.986	**0.562**	1.877	**29.19**	28.07	26.57	27.69	27.04
Bicycle	**0.216**	0.346	0.693	0.255	7.095	**0.283**	0.703	0.795	0.309	0.322	**22.34**	21.08	20.72	20.87	16.70
Bonsai	**0.108**	0.137	0.457	0.125	5.519	**0.214**	0.329	0.654	0.239	0.415	**29.28**	25.72	23.95	25.49	19.99
Counter	**0.068**	0.529	0.191	0.152	26.91	0.101	0.483	0.203	**0.083**	0.509	**25.86**	20.96	23.15	24.42	12.16
Garden	**0.063**	0.295	0.157	0.086	11.95	**0.104**	0.646	0.219	0.199	0.260	**24.05**	21.62	21.01	21.73	15.78
Kitchen	**0.048**	0.052	0.259	0.063	19.89	**0.039**	0.128	0.277	0.064	0.402	**27.20**	26.76	23.10	24.34	13.02
Room	0.142	0.128	0.537	**0.091**	7.494	0.154	0.268	0.532	**0.132**	0.220	**29.45**	29.21	25.19	27.03	18.37

Table 1: Quantitative results of camera pose optimization are presented for both synthetic and real-world scenes. In the majority of tested scenes, the optimization accuracy demonstrates enhancement following the application of G^2fR across both methodologies. In the cases of using Mtrf, we observe a decline in performance for the results in real-world scenes. Further explanations are provided in Sec. 5.3.

5 Experimental Results

We validate the efficacy of G^2fR for both camera pose optimization and few-shot reconstruction tasks employing GFE NeRFs. Our evaluation utilizes Instant-NGP [25] alongside a multi-resolution version of CP-TensoRF [3], denoted as Mtrf, on a platform equipped with a GeForce RTX 4090 GPU. We also make comparisons with several existing methods.

5.1 Camera Pose Optimization

Experiment Settings Following the setting of BARF [18] and CamP [27], we perturb the $\mathfrak{se}(3)$ camera poses with additive noise $\mathcal{N}(0, 0.15\mathbf{I})$ in the synthetic dataset [24], and $\mathcal{N}(0, 0.01\mathbf{I})$ in the real-world dataset [1]. We conduct experiments with and without applying G^2fR while ensuring the same perturbation.

Results Interpretation We offer quantitative evaluation in Tab. 1 and qualitative evaluation in Fig. 4. The findings described in Tab. 1 strongly support the efficacy of the proposed frequency regularization strategy. G^2fR not only is useful within the widely applied Instant NGP paradigm [25], but also holds promise for other multi-resolution GFE NeRFs, such as Mtrf.

Results from the real-world dataset in Tab. 1 demonstrate that G^2fR continues to show good performance when integrated with NGP. There is improvement in both pose accuracy and reconstructed image quality. In the case of using Mtrf, the proposed G^2fR technique seems to have negative effects. Similarly, BARF [18], as a classical method using frequency regularization, also faces

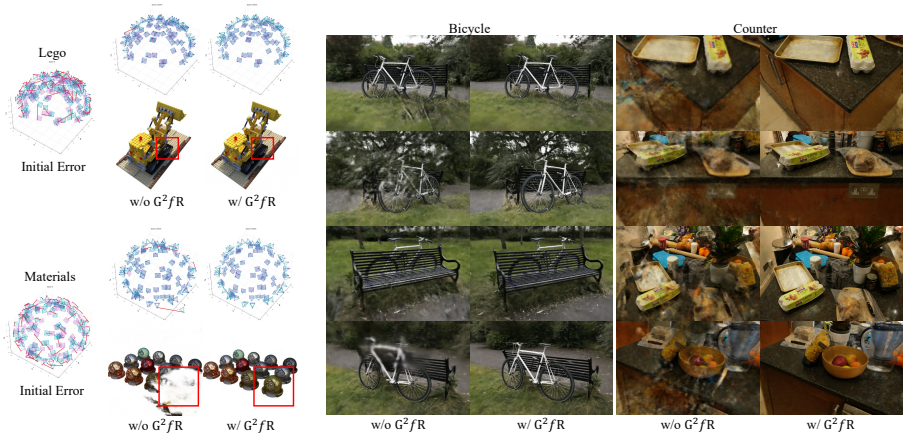


Fig. 4: Qualitative results of camera pose optimization on NeRF-synthetic and MipNeRF-360 dataset using NGP. Both pose accuracy and novel view synthesis quality are improved with the application of G^2fR.

Methods	Rot. error (°) ↓	Trans. error↓ ($\times 10^2$)	PSNR↑	Memory	Iterations	Time
BARF [18]	0.076	0.300	28.32	∼ 6MB	$200k$	$\sim 5h$
L2G-NeRF [5]	0.100	0.433	27.97	∼ 11MB	$200k$	$\sim 3.5h$
RCPR [11]	0.064	0.280	32.21	∼ 150MB	$200k$	$\sim 2h$
CamP [27]	0.164	0.884	<u>32.33</u>	∼ 525MB	$200k$	$\sim 4.5h$
NGP+G^2fR	<u>0.043</u>	**0.143**	**33.55**	∼ 150MB	$20k$	$\sim 5min$
Mtrf+G^2fR	**0.039**	<u>0.148</u>	29.71	∼ 11MB	$50k$	$\sim 20min$

Table 2: Comparative results include rotation and translation errors, PSNR, model memory usages, optimization iterations, and training time. Results of *Lego* from synthetic dataset are shown. After implementing G^2fR, it becomes evident that both NGP and Mtrf reveal commendable performance and achieve efficiency in both time and memory usages.

some difficulties in handling the complex scenes. These phenomena can be considered as one limitation of G^2fR that degrades the memorization capability of the model, which will be discussed in Sec. 5.3.

Comparative Experiments We also compare the performance of our methods with several existing methods that demonstrate amazing performance in the task of concurrent camera pose optimization using synthetic dataset. The tested methods include, BARF [18]: frequency regularization in PE NeRF; L2G-NeRF [5]: local-to-global camera pose representation; RCPR[4] [11]: smooth interpolation instead of linear; CamP [27]: preconditioners for camera pose and intrinsic parameters. Since some of these methods are relatively sensitive to rotation errors, we set the $\mathfrak{se}(3)$ pose perturbation as $\mathcal{N}(0, 0.15\mathbf{I})$ for translation

[4] The codes of [11] are not publicly available when we write this paper. In this study, we use our own implementation according to the description in the paper.

Scene	View synthesis quality PSNR ↑		SSIM ↑		LPIPS ↓		Scene	View synthesis quality PSNR ↑		SSIM ↑		LPIPS ↓	
	NGP w/	NGP w/o	NGP w/	NGP w/o	NGP w/	NGP w/o		NGP w/	NGP w/o	NGP w/	NGP w/o	NGP w/	NGP w/o
Chair	**25.72**	17.02	**0.925**	0.847	**0.076**	0.281	Bicycle	20.63	**21.08**	0.452	**0.487**	0.443	**0.419**
Drums	**14.67**	11.79	**0.724**	0.710	**0.335**	0.530	Bonsai	**24.15**	21.92	**0.825**	0.817	**0.174**	0.189
Ficus	**17.90**	16.07	**0.869**	0.849	**0.177**	0.245	Counter	**20.84**	19.16	**0.689**	0.653	**0.269**	0.324
Hotdog	**27.51**	18.07	**0.919**	0.876	**0.074**	0.198	Garden	**24.32**	23.51	0.679	**0.682**	0.272	**0.253**
Lego	**22.46**	19.85	**0.860**	0.826	**0.094**	0.184	Kitchen	**25.02**	23.92	**0.857**	0.833	**0.129**	0.157
Materials	**17.89**	16.12	**0.813**	0.730	**0.153**	0.282	Room	**25.14**	24.57	0.805	**0.821**	**0.186**	0.208
Mic	**29.92**	26.88	**0.967**	0.943	**0.030**	0.092	–						
Ship	**18.38**	16.23	**0.683**	0.663	**0.234**	0.306	–						

Table 3: Quantitative results of few-shot reconstruction on both the NeRF-synthetic and MipNeRF-360 datasets utilizing NGP. Following the implementation of G^2fR, performance enhancements are evident in most cases. The performance for outdoor environments such as *Bicycle* and *Garden* appears less impressive compared to simpler indoor cases.

Fig. 5: Qualitative results of few-shot reconstruction on NeRF-synthetic and MipNeRF-360 dataset. The improvement brought by G^2fR can be considered as obvious and significant. The noise inside the scene is clearly reduced.

parts, $\mathcal{N}(0, 0.03\mathbf{I})$ for rotation parts, and introduce no intrinsic error. The other settings remain default values. Tab. 2 shows the results of the scene *Lego*.

5.2 Few-Shot NeRF

Experiment Settings We use NGP to demonstrate the efficacy of G^2fR in Few-shot reconstruction task on both the NeRF-synthetic dataset and the MipNeRF-360 dataset. In the case of the synthetic dataset, we use 10 images as inputs, while for real-world scenes, 50 images are employed, constituting approximately 25% of the total image count. Training spans $200k$ iterations.

Results Interpretation As evidenced by the results presented in Tab. 3 and Fig. 5, G^2fR demonstrates significant efficacy across various scenarios. By ap-

plying G^2fR, we can effectively reduce the noise caused by overfitting of high frequency signals. In some outdoor cases from real-world datasets, the impact of implementing G^2fR may not be as significant as in synthetic cases. We leave the analysis of this issue to Sec. 5.3. Nonetheless, it is noteworthy that the only technique employed herein is frequency regularization, a method typically compatible with other strategies. Consequently, the findings of these experiments substantiate the validity of G^2fR in the few-shot reconstruction task.

5.3 Limitation and Future Work

As demonstrated in Tab. 1, the performance of Mtrf shows a decline after applying G^2fR. This phenomenon reveals a limitation of G^2fR: when the model is relatively small in size, such as Mtrf, it tends to fail to model the scene at the beginning of the optimization. Given constrained ability to capture only low-frequency signals, it is unsurprising that it struggles to accurately model the scene, thereby yielding suboptimal directions for pose optimization and exacerbating camera pose inaccuracies. These erroneous camera poses then continue to adversely affect scene modeling, culminating in the failure of pose optimization. On the other hand, if the model still possess enough memorization capability at coarse stages, this problem might not be critical. NGP is a good example that can model the scene properly using only low resolutions.

A similar phenomenon has been reported by Gao *et al.*, emphasizing the importance of accurately aligning frequency bands in PE with the target signals [10]. The findings regarding outdoor scenes presented in Tab. 3 can also be explain through their theory. Gao *et al.* propose the implementation of adaptive frequency bands within PE as a solution [10]. While adjusting frequency bands during training in PE NeRFs is straightforward, it poses a greater challenge in GFE methods. Therefore, one of our forthcoming research targets will be the integration of adaptive frequency bands into GFE methodologies.

6 Conclusion

In this paper, we present a theoretical analysis concerning the expressive capacity of grid-based feature encoding (GFE) methods, which are widely employed in NeRF applications. To the best of our knowledge, this study represents the initial endeavor to explain the distribution of frequency components within the GFE system. Additionally, we propose the adoption of frequency regularization techniques based on this analysis in two distinct tasks: concurrent camera pose optimization and few-shot reconstruction. We conduct experiments on both simulated and real-world data. The experiment results convincingly validate the efficacy of the proposed G^2fR, thereby advocating for the generalizability of the frequency regularization approach across diverse tasks.

Acknowledgements

This work was partially supported by JST PRESTO Grant Number JPMJPR22C4. We would also like to express our thanks and gratitude to Mr. Ryuhei Hamaguchi and Mr. Kimihiro Akiyama for their invaluable comments and advice.

References

1. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5470–5479 (2022)
2. Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Zip-nerf: Anti-aliased grid-based neural radiance fields. ICCV (2023)
3. Chen, A., Xu, Z., Geiger, A., Yu, J., Su, H.: Tensorf: Tensorial radiance fields. In: European Conference on Computer Vision. pp. 333–350. Springer (2022)
4. Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14124–14133 (2021)
5. Chen, Y., Chen, X., Wang, X., Zhang, Q., Guo, Y., Shan, Y., Wang, F.: Local-to-global registration for bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8264–8273 (2023)
6. Chen, Z., Li, Z., Song, L., Chen, L., Yu, J., Yuan, J., Xu, Y.: Neurbf: A neural fields representation with adaptive radial basis functions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4182–4194 (2023)
7. Chng, S.F., Ramasinghe, S., Sherrah, J., Lucey, S.: Gaussian activated neural radiance fields for high fidelity reconstruction and pose estimation. In: European Conference on Computer Vision. pp. 264–280. Springer (2022)
8. Deng, J., Wu, Q., Chen, X., Xia, S., Sun, Z., Liu, G., Yu, W., Pei, L.: Nerf-loam: Neural implicit representation for large-scale incremental lidar odometry and mapping. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8218–8227 (2023)
9. Fridovich-Keil, S., Yu, A., Tancik, M., Chen, Q., Recht, B., Kanazawa, A.: Plenoxels: Radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5501–5510 (2022)
10. Gao, Z., Dai, W., Zhang, Y.: Adaptive positional encoding for bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3284–3294 (2023)
11. Heo, H., Kim, T., Lee, J., Lee, J., Kim, S., Kim, H.J., Kim, J.H.: Robust camera pose refinement for multi-resolution hash encoding. arXiv preprint arXiv:2302.01571 (2023)
12. Herau, Q., Piasco, N., Bennehar, M., Roldão, L., Tsishkou, D., Migniot, C., Vasseur, P., Demonceaux, C.: Moisst: Multi-modal optimization of implicit scene for spatiotemporal calibration. arXiv preprint arXiv:2303.03056 (2023)
13. Hu, W., Wang, Y., Ma, L., Yang, B., Gao, L., Liu, X., Ma, Y.: Tri-miprf: Tri-mip representation for efficient anti-aliasing neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 19774–19783 (2023)

14. Hwang, I., Kim, J., Kim, Y.M.: Ev-nerf: Event based neural radiance field. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 837–847 (2023)
15. Jain, A., Tancik, M., Abbeel, P.: Putting nerf on a diet: Semantically consistent few-shot view synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5885–5894 (2021)
16. Jeong, Y., Ahn, S., Choy, C., Anandkumar, A., Cho, M., Park, J.: Self-calibrating neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5846–5854 (2021)
17. Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8456–8465 (2023)
18. Lin, C.H., Ma, W.C., Torralba, A., Lucey, S.: Barf: Bundle-adjusting neural radiance fields. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5741–5751 (2021)
19. Lin, Y., Müller, T., Tremblay, J., Wen, B., Tyree, S., Evans, A., Vela, P.A., Birchfield, S.: Parallel inversion of neural radiance fields for robust pose estimation. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 9377–9384. IEEE (2023)
20. Ma, Q., Paudel, D.P., Chhatkuli, A., Van Gool, L.: Continuous pose for monocular cameras in neural implicit representation. arXiv preprint arXiv:2311.17119 (2023)
21. Max, N.: Optical models for direct volume rendering. IEEE Transactions on Visualization and Computer Graphics **1**(2), 99–108 (1995)
22. Mehmeti-Göpel, C.H.A., Hartmann, D., Wand, M.: Ringing relus: Harmonic distortion analysis of nonlinear feedforward networks. In: International Conference on Learning Representations (2020)
23. Meuleman, A., Liu, Y.L., Gao, C., Huang, J.B., Kim, C., Kim, M.H., Kopf, J.: Progressively optimized local radiance fields for robust view synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16539–16548 (2023)
24. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
25. Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Transactions on Graphics (ToG) **41**(4), 1–15 (2022)
26. Ortiz, J., Clegg, A., Dong, J., Sucar, E., Novotny, D., Zollhoefer, M., Mukadam, M.: isdf: Real-time neural signed distance fields for robot perception. arXiv preprint arXiv:2204.02296 (2022)
27. Park, K., Henzler, P., Mildenhall, B., Barron, J.T., Martin-Brualla, R.: Camp: Camera preconditioning for neural radiance fields. ACM Transactions on Graphics (TOG) **42**(6), 1–11 (2023)
28. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., Courville, A.: On the spectral bias of neural networks. In: International Conference on Machine Learning. pp. 5301–5310. PMLR (2019)
29. Rahimi, A., Recht, B.: Random features for large-scale kernel machines. Advances in neural information processing systems **20** (2007)
30. Saragadam, V., LeJeune, D., Tan, J., Balakrishnan, G., Veeraraghavan, A., Baraniuk, R.G.: Wire: Wavelet implicit neural representations. In: Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18507–18516 (2023)
31. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in neural information processing systems **33**, 7462–7473 (2020)
32. Sucar, E., Liu, S., Ortiz, J., Davison, A.: iMAP: Implicit mapping and positioning in real-time. In: Proceedings of the International Conference on Computer Vision (ICCV) (2021)
33. Sun, C., Sun, M., Chen, H.T.: Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5459–5469 (2022)
34. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems **33**, 7537–7547 (2020)
35. Wang, P., Chen, X., Chen, T., Venugopalan, S., Wang, Z., et al.: Is attention all nerf needs? arXiv preprint arXiv:2207.13298 (2022)
36. Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2021)
37. Wang, Y., Gong, Y., Zeng, Y.: Hyb-nerf: A multiresolution hybrid encoding for neural radiance fields. arXiv preprint arXiv:2311.12490 (2023)
38. Wang, Y., Han, Q., Habermann, M., Daniilidis, K., Theobalt, C., Liu, L.: Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3295–3306 (2023)
39. Wang, Z., Wu, S., Xie, W., Chen, M., Prisacariu, V.A.: Nerf–: Neural radiance fields without known camera parameters. arXiv preprint arXiv:2102.07064 (2021)
40. Wei, F., Chabra, R., Ma, L., Lassner, C., Zollhöfer, M., Rusinkiewicz, S., Sweeney, C., Newcombe, R., Slavcheva, M.: Self-supervised neural articulated shape and appearance models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15816–15826 (2022)
41. Yang, J., Pavone, M., Wang, Y.: Freenerf: Improving few-shot neural rendering with free frequency regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8254–8263 (2023)
42. Yen-Chen, L., Florence, P., Barron, J.T., Rodriguez, A., Isola, P., Lin, T.Y.: inerf: Inverting neural radiance fields for pose estimation. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1323–1330. IEEE (2021)
43. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4578–4587 (2021)
44. Yüce, G., Ortiz-Jiménez, G., Besbinar, B., Frossard, P.: A structured dictionary perspective on implicit neural representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19228–19238 (2022)
45. Zheng, J., Ramasinghe, S., Li, X., Lucey, S.: Trading positional complexity vs deepness in coordinate networks. In: European Conference on Computer Vision. pp. 144–160. Springer (2022)

46. Zhong, X., Pan, Y., Behley, J., Stachniss, C.: Shine-mapping: Large-scale 3d mapping using sparse hierarchical implicit neural representations. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 8371–8377. IEEE (2023)
47. Zhou, S., Xie, S., Ishikawa, R., Sakurada, K., Onishi, M., Oishi, T.: Inf: Implicit neural fusion for lidar and camera. In: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 10918–10925. IEEE (2023)
48. Zhu, Z., Peng, S., Larsson, V., Xu, W., Bao, H., Cui, Z., Oswald, M.R., Pollefeys, M.: Nice-slam: Neural implicit scalable encoding for slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12786–12796 (2022)