Dictionary Fields: Learning a Neural Basis Decomposition

ANPEI CHEN, ETH Zürich, University of Tübingen, Switzerland
ZEXIANG XU, Adobe Research, USA
XINYUE WEI, UC San Diego, USA
SIYU TANG, ETH Zürich, Switzerland
HAO SU, UC San Diego, USA
ANDREAS GEIGER, University of Tübingen, Tübingen AI Center, Germany

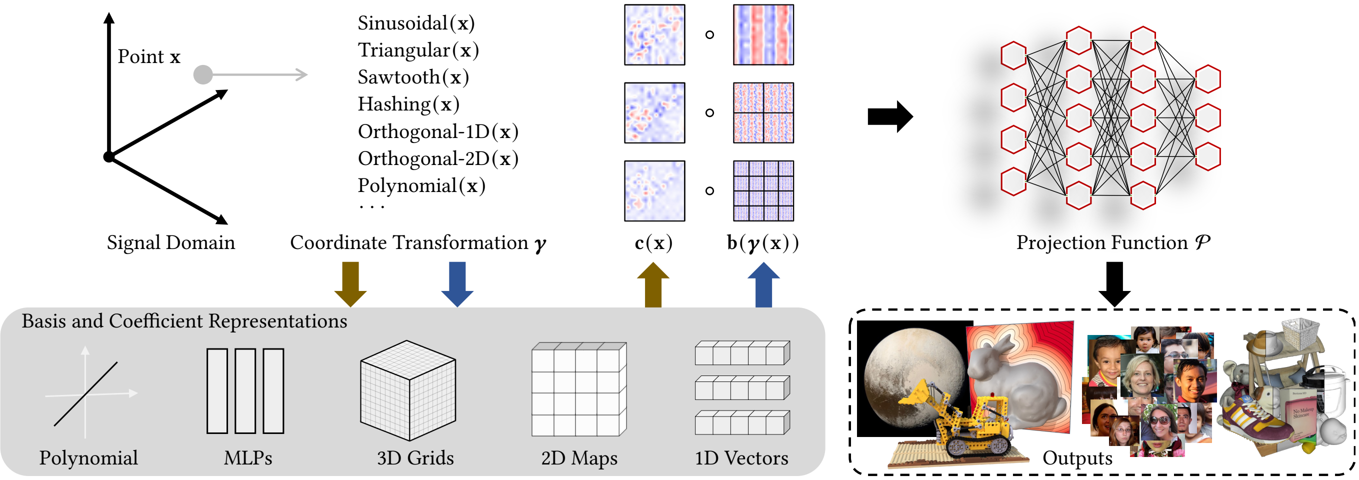


Fig. 1. **Dictionary Fields** factorize a signal into a coefficient field $\mathbf{c}(\mathbf{x})$ and basis field $\mathbf{b}(\boldsymbol{\gamma}(\mathbf{x}))$ (top-center), each of which is represented by one out of many possible field representations (bottom-left). The basis field allows for spatial repetition via a suitably chosen coordinate transformation $\boldsymbol{\gamma}(\mathbf{x})$ (top-left). The resulting Hadamard product field $\mathbf{c}(\mathbf{x}) \circ \mathbf{b}(\boldsymbol{\gamma}(\mathbf{x}))$ is passed to a projection function (e.g., MLP) which maps it to the signal's output domain (bottom-right).

We present Dictionary Fields, a novel neural representation which decomposes a signal into a product of factors, each represented by a classical or neural field representation, operating on transformed input coordinates. More specifically, we factorize a signal into a coefficient field and a basis field, and exploit periodic coordinate transformations to apply the same basis functions across multiple locations and scales. Our experiments show that Dictionary Fields lead to improvements in approximation quality, compactness, and training time when compared to previous fast reconstruction methods. Experimentally, our representation achieves better image approximation quality on 2D image regression tasks, higher geometric quality when reconstructing 3D signed distance fields, and higher compactness for radiance field reconstruction tasks. Furthermore, Dictionary Fields enable generalization to unseen images/3D scenes by sharing bases across signals during training which greatly benefits use cases such as image regression from partial observations and few-shot radiance field reconstruction. Our code is available at https://apchenstu.github.io/FactorFields/.

CCS Concepts: • **Computing methodologies → Rendering**; *Computational photography*; Shape representations.

Additional Key Words and Phrases: Neural Representation, Reconstruction, Neural Radiance Fields

ACM Reference Format:
Anpei Chen, Zexiang Xu, Xinyue Wei, Siyu Tang, Hao Su, and Andreas Geiger. 2023. Dictionary Fields: Learning a Neural Basis Decomposition. *ACM Trans. Graph.* 42, 4 (August 2023), 12 pages. https://doi.org/10.1145/3592135

1 INTRODUCTION

Effectively representing multi-dimensional digital content – like 2D images or 3D geometry and appearance – is critical for computer graphics and vision applications. These digital signals are traditionally represented discretely as pixels, voxels, textures, or polygons. Recently, significant headway has been made in developing advanced neural representations [Chen et al. 2022; Mildenhall et al. 2020; Müller et al. 2022; Sitzmann et al. 2020; Sun et al. 2022], which demonstrated superiority in modeling accuracy and

Authors' addresses: Anpei Chen, ETH Zürich, University of Tübingen, Zürich, Switzerland, anpei.chen@inf.ethz.ch; Zexiang Xu, Adobe Research, San Jose, USA, zexu@adobe.com; Xinyue Wei, UC San Diego, San Diego, USA, xiwei@ucsd.edu; Siyu Tang, ETH Zürich, Zürich, Switzerland, siyu.tang@inf.ethz.ch; Hao Su, UC San Diego, San Diego, USA, haosu@eng.ucsd.edu; Andreas Geiger, University of Tübingen, Tübingen AI Center, Tübingen, Germany, a.geiger@uni-tuebingen.de.

0730-0301/2023/8-ART $15.00
https://doi.org/10.1145/3592135

efficiency over traditional representations for different image synthesis and scene reconstruction applications. Notably, TensoRF [Chen et al. 2022] introduced a tensor factorization-based representation, which can be seen as a representation of two Vector-Matrix or three CANDECOMP-PARAFAC decomposition factors with axis-aligned orthogonal 2D and 1D projections as transformations. While TensoRF demonstrated the potential of multi-factor representations, it is naturally limited to simple orthogonal transformations.

Motivated by this observation, we propose *Dictionary Fields* (Fig. 1), a two-factor representation that is composed of (1) a basis field factor with periodic transformation to model the commonalities of patterns that are shared across the entire signal domain and (2) a coefficient field factor to express localized spatially-varying features in the signal. The target signal is then regressed from the factor product via a learned projection function (e.g., MLP). The combination of both factors allows for an efficient representation of the global and local properties of the signal. Compared to methods that model only a single factor (e.g., NeRF, Instant-NGP, DVGO, Plenoxels), jointly modeling two factors (basis and coefficients) leads to superior quality and enables compact and fast reconstruction, as we demonstrate on various downstream tasks.

We conduct a rich set of ablation experiments over the choice of basis/coefficient functions and basis transformations. In particular, we evaluate Dictionary Fields against various variants and baselines on three classical signal representation tasks: 2D image regression, 3D SDF geometry reconstruction, and radiance field reconstruction for novel view synthesis. We demonstrate that our factorized multi-scale representation is able to achieve state-of-the-art reconstruction results that are better or on par with previous methods, while achieving superior modeling efficiency. For instance, compared to Instant-NGP our method leads to better reconstruction and rendering quality, while effectively *halving* the total number of model parameters (capacity) for SDF and radiance field reconstruction, demonstrating its superior accuracy and efficiency.

Moreover, in contrast to recent neural representations that are designed for purely per-scene optimization, our factorized representation framework is able to learn basis functions across different scenes. As shown in preliminary experiments, this enables learning across-scene bases from multiple 2D images or 3D radiance fields, leading to signal representations that generalize and hence improve reconstruction results from sparse observations such as in the few-shot radiance reconstruction setting. In summary,

- We propose Dictionary Fields, a **new representation** that factorizes a signal into coefficient and basis factors which allows for exploiting similar signatures spatially and across scales.
- Our model can be trained jointly on **multiple signals**, recovering general basis functions that allow for reconstructing parts of a signal from sparse or weak observations.
- We present thorough experiments and **systematic ablation studies** that demonstrate improved performance (accuracy, runtime, memory), and shed light on the performance improvements in three pre-scene optimization and two generalization tasks.

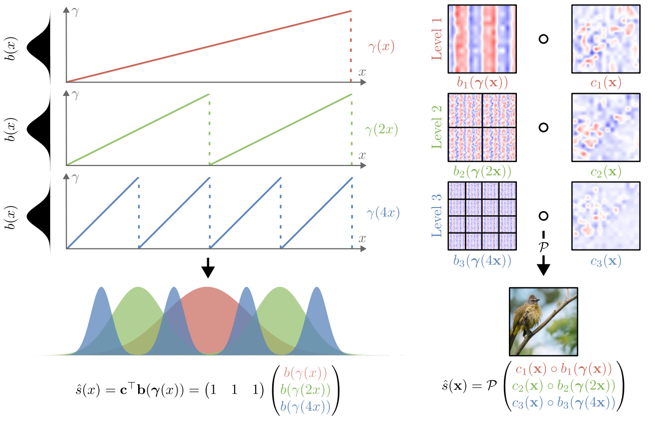
2 RELATED WORK

We now introduce standard dictionary factorization techniques and recent advances in neural fields.

Dictionary Factorization. Representing data using a smaller set of basis functions has been well studied for decades, from theory [Lee and Seung 2000; Olshausen and Field 1997] to diverse applications in computer vision and graphics, such as data compression, image inpainting and classification [de Queiroz and Chou 2016; Elad et al. 2005; Yang et al. 2009]. Two popular decomposition techniques in signal processing are transformation techniques and dictionary learning. Transformation techniques, such as Discrete Cosine and Wavelet transforms, have been widely used for decades [Ahmed et al. 1974; Grossmann and Morlet 1984]. They transform a signal into a different domain where it can be easily represented by a smaller set of coefficients. In contrast, dictionary learning aims to learn a set of basis functions that can represent the input signals sparsely [Heide et al. 2015; Mairal et al. 2009; Olshausen and Field 1996; Rubinstein et al. 2008; Wright et al. 2009; Yang et al. 2010]. By finding a linear combination of only a few basis functions, dictionary learning is able to achieve efficient representations that capture the essential structure of the signals.

In recent years, several attempts have been made to improve the performance of signal processing tasks by combining standard dictionary factorization techniques with neural networks. For instance, [Fu et al. 2019; Zheng et al. 2021] combined convolutional neural networks with dictionary factorization for image compression and denoising. In contrast, our work does not address a decomposition/compression problem, but a reconstruction problem based on gradient descent, since the feature grid/tensor is unknown initially. Concurrently, [Huang et al. 2022; Wu et al. 2022] perform frequency-wise decomposition using fast Fourier transformation or sine activations. In our work, we utilize dictionary factorization that decomposes a signal into a learned coefficient field and a basis field for efficient and general signal representation.

Neural Fields. Recently, neural field representations have emerged as a promising replacement for traditional representations in representing natural signals, such as 2D images, 3D geometry, 5D radiance field. Seminal works Occupancy Networks [Mescheder et al. 2019], IMNet [Chen and Zhang 2019] and DeepSDF [Park et al. 2019] propose to represent the 3D surface implicitly as the continuous decision boundary of an MLP classifier or by regressing a signed distance value, providing a continuous implicit 3D mapping thus allows for the extraction of 3D meshes at any resolution. While this representation is able to generate high-quality meshes, it fails to model high-frequency signals, such as images due to the implicit smoothness bias of MLPs. To address this issue, a number of approaches based on positional encoding [Lindell et al. 2022; Mildenhall et al. 2020; Shekarforoush et al. 2022; Tancik et al. 2020], transform spatial coordinates to Fourier space with a set of sinusoidal functions and then utilize an MLP to obtain the prediction. Similarly, SIREN [Sitzmann et al. 2020] and MFN [Fathony et al. 2021] propose to leverage periodic activation functions for implicit neural representations, achieving promising complicated signal modeling while preserving high-order gradients. To balance memorization


b(x)
γ
γ(x)
x
γ(2x)
γ(4x)
ŝ(x) = c⊤b(γ(x)) = (1 1 1) (b(γ(x)) b(γ(2x)) b(γ(4x)))
Level 1
Level 2
Level 3
b1(γ(x))
c1(x)
b2(γ(2x))
c2(x)
b3(γ(4x))
c3(x)
∘
𝒫
ŝ(x) = 𝒫 (c1(x) ∘ b1(γ(x)) c2(x) ∘ b2(γ(2x)) c3(x) ∘ b3(γ(4x)))

(a) 1D Illustration of Eq. (2) (b) 2D Illustration of Eq. (3)

Fig. 2. **Coefficient and Basis Factorization.** (a) Choosing a (periodic) coordinate transformation $\boldsymbol{\gamma}(x)$ allows for applying the same basis function $b(x)$ at multiple spatial locations and scales. For clarity, we have chosen constant coefficients $\mathbf{c} = \mathbf{1}$ and a single shared Gaussian basis for this example. (b) Composing multiple bases at different spatial resolutions with their respective coefficients yields a powerful representation for signal $s(\mathbf{x})$. In this example we use a single (distinct) basis at each of the 3 levels. In practice, we use multiple learned bases and coefficient fields at each resolution.

and generalization, [Ramasinghe and Lucey 2021] propose to learn a positional embedding based on the classic graph-Laplacian regularization. Though using the pure MLP to map coordinates to the target signal domain provides global and compact modeling, they require a long training time for local details recovery. Recent works [Chen et al. 2022; Fridovich-Keil et al. 2022; Müller et al. 2022; Sun et al. 2022; Takikawa et al. 2022] introduce grid embedding for fast optimization and fine details recovery.

Neural fields have also been widely deployed as representation for various other graphics and vision applications, such as novel view synthesis [Aliev et al. 2019; Chen et al. 2022; Liu et al. 2020; Lombardi et al. 2019; Mildenhall et al. 2020; Thies et al. 2019; Verbin et al. 2022; Xu et al. 2022; Zhou et al. 2018], generative models [Chan et al. 2022, 2021; Gao et al. 2022; Schwarz et al. 2020], 3D surface reconstruction [Chabra et al. 2020; Hui et al. 2022; Jiang et al. 2020; Kobayashi et al. 2022; Niemeyer et al. 2020; Wang et al. 2021; Yariv et al. 2021; Yu et al. 2022], image processing [Chen et al. 2021a], graphics asset modeling [Kuznetsov et al. 2021; Rainer et al. 2019; Zhu et al. 2021], inverse rendering [Bi et al. 2020a,b; Boss et al. 2021a,b; Zhang and Ohn-Bar 2021; Zhang et al. 2022], dynamic scene modeling [Fridovich-Keil et al. 2023; Li et al. 2021, 2020; Park et al. 2021; Pumarola et al. 2021; Ramasinghe et al. 2023; Song et al. 2023] and scene understanding [Matthew et al. 2023; Peng et al. 2022]. Such applications may all directly benefit from the development of new signal representations.

3 DICTIONARY FIELDS

We seek to compactly represent a continuous Q-dimensional signal $\mathbf{s} : \mathbb{R}^D \rightarrow \mathbb{R}^Q$ on a D-dimensional domain. We assume that signals are not random, but structured and hence share similar signatures within the same signal (spatially and across different scales) as well as between different signals. In the following, we develop our Dictionary Fields model step-by-step, starting from a standard basis expansion.

Let us first consider a 1D signal $s(\mathbf{x}) : \mathbb{R}^D \rightarrow \mathbb{R}$. Using basis expansion, we decompose $s(\mathbf{x})$ into a set of coefficients $\mathbf{c} = (c_1, \ldots, c_K)^\top$ with $c_k \in \mathbb{R}$ and basis functions $\mathbf{b}(\mathbf{x}) = (b_1(\mathbf{x}), \ldots, b_K(\mathbf{x}))^\top$ with $b_k : \mathbb{R}^D \rightarrow \mathbb{R}$:

$$\hat{s}(\mathbf{x}) = \mathbf{c}^\top \mathbf{b}(\mathbf{x}) \tag{1}$$

Note that we denote $s(\mathbf{x})$ as the true signal and $\hat{s}(\mathbf{x})$ as its approximation.

Representing the signal $s(\mathbf{x})$ using a global set of basis functions is inefficient as information cannot be shared spatially. We hence generalize the above formulation by (i) exploiting a spatially varying coefficient field $\mathbf{c}(\mathbf{x}) = (c_1(\mathbf{x}), \ldots, c_K(\mathbf{x}))^\top$ with $c_k : \mathbb{R}^D \rightarrow \mathbb{R}$ and (ii) transforming the coordinates of the basis functions via a coordinate transformation function $\boldsymbol{\gamma} : \mathbb{R}^D \rightarrow \mathbb{R}^B$. Note that in general transformed coordinate dimension B does not need to match signal domain dimension D, and hence the domain of the basis functions also changes accordingly: $b_k : \mathbb{R}^B \rightarrow \mathbb{R}$:

$$\hat{s}(\mathbf{x}) = \mathbf{c}(\mathbf{x})^\top \mathbf{b}\big(\boldsymbol{\gamma}(\mathbf{x})\big) \tag{2}$$

When choosing $\boldsymbol{\gamma}$ to be a periodic function, this formulation allows us to apply the same basis at multiple spatial locations and optionally also at multiple scales while varying the coefficients $\mathbf{c}$, as illustrated in Section 3.2 and Fig. 2 (a). Standard approaches with patch-wise basis (e.g., [Tang et al. 2018]) can be viewed as a special case of Eq. (2) when the coefficient field is piecewise-constant. However, the linear representation significantly limited the model's capability and many have more than a single dimension (e.g., 3 in the case of RGB images or 4 in the case of radiance fields).

We further generalize our model to Q-dimensional signals $\mathbf{s}(\mathbf{x})$ by introducing a projection function $\mathcal{P} : \mathbb{R}^K \rightarrow \mathbb{R}^Q$ and replacing the inner product with the element-wise/Hadamard product (denoted by $\circ$ in the following):

$$\hat{\mathbf{s}}(\mathbf{x}) = \mathcal{P}\big(\mathbf{c}(\mathbf{x}) \circ \mathbf{b}\big(\boldsymbol{\gamma}(\mathbf{x})\big)\big) \tag{3}$$

We refer to Eq. (3) as **Dictionary Fields (DiF)**. Note that in contrast to the scalar product $\mathbf{c}^\top\mathbf{b}$ in Eq. (2), the output of $\mathbf{c} \circ \mathbf{b}$ is a K-dimensional vector which comprises the individual coefficient-basis products as input to the projection function $\mathcal{P}$ which itself can be either linear or non-linear. In the linear case, we have $\mathcal{P}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ with $\mathbf{A} \in \mathbb{R}^{Q \times K}$. Moreover, note that for $Q = 1$ and $\mathbf{A} = (1, \ldots, 1)$ we recover Eq. (2) as a special case. In our experiments, we use a shallow Multi-Layer Perceptron (MLP) to model $\mathcal{P}(\mathbf{x})$.

In our formulation, $\boldsymbol{\gamma}$ is a deterministic functions while $\mathcal{P}$, $\mathbf{c}$ and $\mathbf{b}$ are parametric mappings (e.g., polynomials, multi-layer perceptrons or 3D feature grids) whose parameters (collectively named θ below) are optimized. Note that the parameters θ can be optimized either for a single signal or jointly for multiple signals. When optimizing for multiple signals jointly, we share the parameters of the projection function and basis field $\mathbf{b}$ (but not the parameters of the coefficient field $\mathbf{c}$) across signals. The projection operator $\mathcal{P}$ can also be utilized to model the volumetric rendering operation when reconstructing a 3D radiance field from 2D image observations, see Section 3.3.

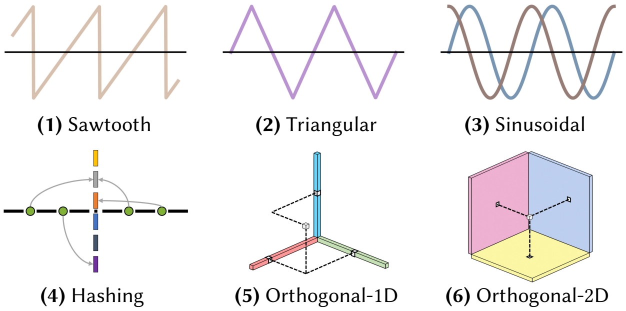


Fig. 3. **Coordinate Transformations.** We show various periodic (top) and non-periodic (bottom) coordinate transformations $\boldsymbol{\gamma}$ used in our framework.

3.1 Field Representations $\mathbf{c}$ and $\mathbf{b}$

For modeling the coefficient field $\mathbf{c}$ and the basis field $\mathbf{b}$, we consider various different representations as illustrated in Fig. 1 (bottom-left). MLPs have been proposed as signal representations in Occupancy Networks [Mescheder et al. 2019], DeepSDF [Park et al. 2019] and NeRF [Mildenhall et al. 2020]. While MLPs excel in compactness and induce a useful smoothness bias, they are slow to evaluate and hence increase training and inference time. To address this, DVGO [Sun et al. 2022] proposes a 3D voxel grid representation for radiance fields. While voxel grids are fast to optimize, they increase memory significantly and do not easily scale to higher dimensions. To better capture the sparsity in the signal, Instant-NGP [Müller et al. 2022] proposes a hash function in combination with 1D feature vectors instead of a dense voxel grid. Our work allows any of the above representations to model the coefficients and bases. We analyze and compare various combinations in our experiments, considering the dense grid as the default setting of our DiF model.

3.2 Coordinate Transformation $\boldsymbol{\gamma}$

The coordinates input to the basis field $\mathbf{b}$ are transformed by a coordinate transformation function $\boldsymbol{\gamma} : \mathbb{R}^D \rightarrow \mathbb{R}^B$.

Local Basis. The coordinate transformation $\boldsymbol{\gamma}$ enables the application of the same basis function $\mathbf{b}$ at *multiple locations* as illustrated in Fig. 2. In this paper, we consider sawtooth, triangular, sinusoidal (as in NeRF [Mildenhall et al. 2020]) and hashing (as in Instant-NGP [Müller et al. 2022]) transformations, see Fig. 3.

Multi-scale Basis. The coordinate transformation $\boldsymbol{\gamma}$ also allows for applying the same basis $\mathbf{b}$ at *multiple spatial resolutions* of the signal by transforming the coordinates $\mathbf{x}$ with (periodic) transformations of different frequencies as illustrated in Fig. 2. This is crucial as signals typically carry both high and low frequencies, and we seek to exploit our basis representation across the full spectrum to model fine details of the signal as well as smooth signal components.

Specifically, we model the target signal with a set of multi-scale basis functions. We arrange the basis into L levels where each level covers a different scale. Let $[\mathbf{u}, \mathbf{v}]$ denote the bounding box of the signal along on dimension. The corresponding scale is given by $(\mathbf{v} - \mathbf{u})/f_l$ where f_l is the frequency at level l. A large scale basis (e.g., level 1) has a low frequency and covers a large region of the target signal while a small scale basis (e.g., level L) has a large frequency f_L covering a small region of the target signal.

We implement our multi-scale representation (PR) by multiplying the scene coordinate $\mathbf{x}$ with the level frequency f_l before feeding it to the coordinate transformation function $\boldsymbol{\gamma}$ and then concatenating the results across the different levels $l = 1, \ldots, L$:

$$\boldsymbol{\gamma}^{\text{PR}}(\mathbf{x}) = \left(\boldsymbol{\gamma}(\mathbf{x} f_1), \ldots, \boldsymbol{\gamma}(\mathbf{x} f_L)\right) \tag{4}$$

Here, $\boldsymbol{\gamma}$ is any of the coordinate transformations in Fig. 3, and $\boldsymbol{\gamma}_{\text{PR}}$ is the final coordinate transform of our multi-scale representation. As shown in Fig. 2 (b), this results in the target signal being decomposed as the product of spatial varying coefficient maps and multi-level basis maps which comprise repeated local basis functions.

3.3 Projection $\mathcal{P}$

To represent multi-dimensional signals, we introduced a projection function $\mathcal{P} : \mathbb{R}^K \rightarrow \mathbb{R}^Q$ that maps from the K-dimensional Hadamard product $\mathbf{c} \circ \mathbf{b}$ to the Q-dimensional target signal. We distinguish two cases in our framework: the case where direct observations from the target signal are available (e.g., pixels of an RGB image) and the indirect case where observations are projections of the target signal (e.g., pixels rendered from a radiance field).

Direct Observations. In the simplest case, the projection function realizes a learnable linear mapping $\mathcal{P}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ with parameters $\mathbf{A} \in \mathbb{R}^{Q \times K}$ to map the K-dimensional Hadamard product $\mathbf{c} \circ \mathbf{b}$ to the Q-dimensional signal. However, a more flexible model is attained if $\mathcal{P}$ is represented by a shallow non-linear multi-layer perceptron (MLP) which is the default setting in all of our experiments.

Indirect Observations. In some cases, we only have access to *indirect* observations of the signal. For example, when optimizing neural radiance fields, we typically only observe 2D images instead of the 4D signal (density and radiance). In this case, we extend $\mathcal{P}$ to also include the differentiable volumetric rendering process. More concretely, we first apply a multi-layer perceptron to map the view direction and the product features $\mathbf{c} \circ \mathbf{b}$ at a particular location to a color value and a volume density. Next, we follow NeRF [Mildenhall et al. 2020] and render RGB pixels using volumetric integration , see [Mildenhall et al. 2020] for details. Note that in this case, the *composition* of the learned MLP and the volume rendering function constitute the projection function $\mathcal{P}$.

3.4 Space Contraction

We normalize the input coordinates $\mathbf{x} \in \mathbb{R}^D$ to $[0, 1]$ before passing them to the coordinate transformations $\boldsymbol{\gamma}(\mathbf{x})$ by applying a simple space contraction function to $\mathbf{x}$. We distinguish two settings:

For *bounded signals* with D-dimensional bounding box $[\mathbf{u}, \mathbf{v}]$ (where $\mathbf{u}, \mathbf{v} \in \mathbb{R}^D$), we utilize a simple linear mapping to normalize all coordinates to the range $[0, 1]$:

$$\text{contract}(\mathbf{x}) = \frac{\mathbf{x} - \mathbf{u}}{\mathbf{v} - \mathbf{u}} \tag{5}$$

For *unbounded signals* (e.g., an outdoor radiance field), we adopt Mip-NeRF 360's [Barron et al. 2022] space contraction function[1]:

$$\operatorname{contract}(\mathbf{x}) = \begin{cases} \mathbf{x} & \|\mathbf{x}\|_2 \leq 1 \\ \left(2 - \frac{1}{\|\mathbf{x}\|_2}\right)\left(\frac{\mathbf{x}}{\|\mathbf{x}\|_2}\right) & \|\mathbf{x}\|_2 > 1 \end{cases} \tag{6}$$

3.5 Optimization

Given samples $\{(\mathbf{x}, \mathbf{s}(\mathbf{x}))\}$ from the signal, we minimize

$$\underset{\theta}{\operatorname{argmin}} \ \mathbb{E}_{\mathbf{x}}\left[\|\mathbf{s}(\mathbf{x}) - \hat{\mathbf{s}}_\theta(\mathbf{x})\|_2 + \Psi(\theta_{\mathbf{c}})\right] \tag{7}$$

where $\Psi(\theta_{\mathbf{c}})$ is a regularizer on the coefficients. We optimize this objective using stochastic gradient descent.

Sparsity Regularization. While using the ℓ_0 norm for sparse coefficients is desirable, this leads to a difficult optimization problem. Instead, we use a simpler strategy which we found to work surprisingly well. We regularize our objective by randomly dropping a subset of the K features of our model by setting them to zero with probability μ. This forces the signal to be represented with random combinations of features at every iteration, encouraging sparsity and preventing co-adaptation of features. We implement this dropout regularizer using a random binary vector $\mathbf{m}$ which we multiply element-wise with the product field: $\mathbf{m} \circ \mathbf{c} \circ \mathbf{b}$.

Initialization. During all our experiments, we initialize the basis factors using the discrete cosine transform (DCT) basis functions, while initializing the parameters of the coefficient factors and projection MLP randomly. We experimentally found this to improve the quality of the solution.

Multiple Signals. When optimizing for multiple signals jointly, we share the parameters of the projection function and basis fields (but not the parameters of the coefficient fields) across signals. As evidenced by our experiments in Section 4.3, sharing bases across different signals while encouraging sparse coefficients improves generalization and enables reconstruction from sparse observations.

4 EXPERIMENTS

We now present extensive evaluations of our Dictionary Field representation. We first briefly discuss our implementation and hyperparameter configuration. We then compare the performance of DiF with previously proposed representations on both per-signal reconstruction (optimization) and across-signal generalization tasks. At the end of this section, we examine the properties of our method by varying the level number L, different types of transformation function $\boldsymbol{\gamma}$, field representation of the coefficient $\mathbf{c}$ and basis $\mathbf{b}$, and field connector $\circ$. In the following section, we denote the different model variants of our DiF factorizations labeled "DiF-xx", where "xx" indicates the differences from the default setting "DiF-Grid". For example, "DiF-MLP-B" refers to using an MLP basis representation, and "DiF-Hash-B" stands for using a hash coordinate transformation function for basis.

4.1 Implementation

We implement our DiF using vanilla PyTorch without customized CUDA kernels. Performance is evaluated on a single RTX 6000 GPU using the Adam optimizer [Kingma and Ba 2015] with a learning rate of 0.02.

We instantiate DiF using $L = 6$ levels with frequencies (linearly increasing) $f_l \in [2., 3.2, 4.4, 5.6, 6.8, 8.]$, and feature channels $K = [4, 4, 4, 2, 2, 2]^\top \cdot 2^\eta$, where η controls the number of feature channels. We use $\eta = 3$ for our 2D experiments and η= 0 for our 3D experiments. The model parameters θ are distributed across 3 model components: coefficients $\theta_{\mathbf{c}}$, basis $\theta_{\mathbf{b}}$, and projection function $\theta_{\mathcal{P}}$. The size of each component can vary greatly depending on the chosen representation.

In the following experiments, we refer to the default model setting as "DiF-Grid", which implements the coefficients $\mathbf{c}$ and bases $\mathbf{b}$ with learnable tensor grids, $\mathcal{P}(\mathbf{x}) = \operatorname{MLP}(\mathbf{x})$, and $\boldsymbol{\gamma}(\mathbf{x}) = \operatorname{Sawtooth}(\mathbf{x})$, where $\operatorname{Sawtooth}(\mathbf{x}) = \mathbf{x} \bmod 1.0$. In the DiF-Grid setting, the total number of optimizable parameters is mainly determined by the resolution of the coefficient $M_{\mathbf{c}}^l$ and basis grids $M_{\mathbf{b}}^l$:

$$|\theta| = |\theta_{\mathcal{P}}| + |\theta_{\mathbf{c}}| + |\theta_{\mathbf{b}}| = |\theta_{\mathcal{P}}| + \sum_{l=1}^{L} {M_{\mathbf{c}}^l}^D + K_l \cdot {M_{\mathbf{b}}^l}^D \tag{8}$$

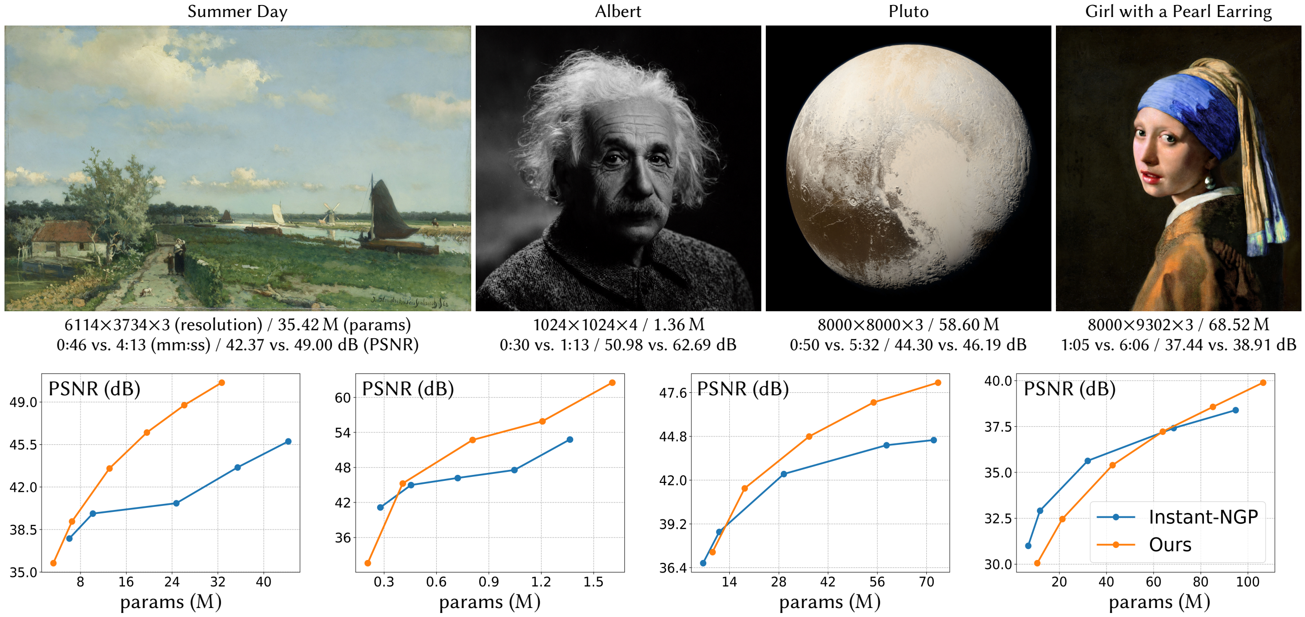
We implement the basis grid using linearly increasing resolutions $M_{\mathbf{b}}^l \in [32, 128]^T \cdot \frac{min(\mathbf{v}-\mathbf{u})}{1024}$ with interval $[32, 128]$ and scene bounding box $[u, v]$. This leads to increased resolution for modeling higher-resolution signals in our experiments. We use the same coefficient grid resolution $M_{\mathbf{c}}^l$ across all L levels for query efficiency and to lower per-signal memory footprint.

We first evaluate the accuracy and efficiency of our DiF-Grid representation on various multi-dimensional signals, comparing it to several recent neural signal representations. Towards this goal, we consider three popular benchmark tasks for evaluating neural representations: 2D image regression, 3D Signed Distance Field (SDF) reconstruction and Radiance Field Reconstruction / Novel View Synthesis. We evaluate each method's ability to approximate high-frequency patterns, interpolation quality, compactness, and robustness to ambiguities and sparse observations.

4.2 Single Signals

2D Image Regression. In this task, we directly regress RGB pixel colors from pixel coordinates. We evaluate our DiF-Grid on fitting four complex high-resolution images, where the total number of pixels ranges from 4 M to 213 M. In Fig. 4, we show the reconstructed images with the corresponding model size, optimization time, and image PSNRs, and compare them to Instant-NGP [Müller et al. 2022], a state-of-the-art neural representation that supports image regression and has shown superior quality over prior art including Fourier Feature Networks [Tancik et al. 2020] and SIREN [Sitzmann et al. 2020]. Compared to Instant-NGP, our model consistently achieves higher PSNR on all images when using the same model size, demonstrating the superior accuracy and efficiency of our model. On the other hand, while Instant-NGP achieves faster optimization owing to its highly optimized CUDA-based framework, our model, implemented in pure PyTorch, leads to comparably fast training while

[1] In our implementation, we slightly modify Eq. (6) to map coordinates to a unit ball centered at 0.5 which avoids negative coordinates when indexing feature grids.



6114×3734×3 (resolution) / 35.42 M (params)
0:46 vs. 4:13 (mm:ss) / 42.37 vs. 49.00 dB (PSNR)

1024×1024×4 / 1.36 M
0:30 vs. 1:13 / 50.98 vs. 62.69 dB

8000×8000×3 / 58.60 M
0:50 vs. 5:32 / 44.30 vs. 46.19 dB

8000×9302×3 / 68.52 M
1:05 vs. 6:06 / 37.44 vs. 38.91 dB

Fig. 4. **2D Image Regression.** This figure shows images represented using our DiF-Grid model. The respective image resolutions and numbers of model parameters are shown below each image. Moreover, we also report a comparison to Instant-NGP (first number) in terms of optimization time and PSNR metrics (Instant-NGP vs Ours) at the bottom using the same number of model parameters. Note that our method achieves better reconstruction quality on all images when using the same model size. While optimization is slower than Instant-NGP, we use a vanilla PyTorch implementation without customized CUDA kernels. “Summer Day” credit goes to Johan Hendrik Weissenbruch and rijksmuseum. “Albert” credit goes to Orren Jack Turner. “Pluto” credit goes to NASA. “Girl With a Pearl Earring” renovation ©Koorosh Orooj (CC BY-SA 4.0).

relying on a vanilla PyTorch implementation without custom CUDA kernels which simplifies future extensions.

Signed-Distance Field Reconstruction. Signed Distance Function (SDF), as a classic geometry representation, describes a set of continuous iso-surfaces, where a 3D surface is represented as the zero level-set of the function. We evaluate our DiF-Grid on modeling several challenging object SDFs that contain rich geometric details and compare with previous state-of-the-art neural representations, including Fourier Feature Networks [Tancik et al. 2020], SIREN [Sitzmann et al. 2020], and Instant-NGP [Müller et al. 2022]. To allow for fair comparisons in terms of the training set and convergence, we use the same training points for all methods by pre-sampling 8 M SDF points from the target meshes for training, with 80% points near the surface and the remaining 20% points uniformly distributed inside the unit volume. Following the evaluation setting of Instant-NGP, we randomly sample 16 M points for evaluation and calculate the geometric IOU metric based on the SDF sign.

$$gIoU = \frac{\sum(s(\mathbf{X}) > 0) \cap (\hat{s}(\mathbf{X}) > 0)}{\sum(s(\mathbf{X}) > 0) \cup (\hat{s}(\mathbf{X}) > 0)} \tag{9}$$

where $\mathbf{X}$ is the evaluation point set, $s(\mathbf{X})$ are the ground truth SDF values, and $\hat{s}(\mathbf{X})$ are the predicted SDF values.

Fig. 5 shows a quantitative and qualitative comparison of all methods. Our method leads to visually better results, it recovers high-frequency geometric details and contains less noise on smooth surfaces (e.g., the elephant face). The high visual quality is also reflected by the highest gIoU and Chamfer Distance (CD) value of all methods. Meanwhile, our method also achieves the fastest reconstruction speed, while using less than half of the number of parameters used by CUDA-kernel enabled Instant-NGP, demonstrating the high accuracy, efficiency, and compactness of our factorized representation.

Radiance Field Reconstruction. Radiance field reconstruction aims to recover the 3D density and radiance of each volume point from multi-view RGB images. The geometry and appearance properties are updated via inverse volume rendering, as proposed in NeRF [Mildenhall et al. 2020]. Recently, many encoding functions and advanced representations have been proposed that significantly improve reconstruction speed and quality, such as sparse voxel grids [Fridovich-Keil et al. 2022], hash tables [Müller et al. 2022] and tensor decomposition [Chen et al. 2022].

In Table 1, we quantitatively compare DiF-Grid with several state-of-the-art fast radiance field reconstruction methods (Plenoxel [Fridovich-Keil et al. 2022], DVGO [Sun et al. 2022], Instant-NGP [Müller et al. 2022] and TensoRF-VM [Chen et al. 2022]) on both synthetic [Mildenhall et al. 2020] as well as real Tanks and Temple objects [Knapitsch et al. 2017]. Note that, we re-run Instant-NGP with the official code, using the same input (RGB) and iterations ($30k$) setting for a fair comparison. Our method achieves high reconstruction quality, significantly outperforming NeRF, Plenoxels, and DVGO on both datasets, while being significantly more compact than Plenoxels and DVGO. We also outperform Instant-NGP and are on par with TensoRF regarding reconstruction quality, while being highly compact with only 5.1 M parameters, less than one-third of TensoRF-VM and one-half of Instant-NGP. Our DiF-Grid also optimizes faster than TensoRF, at slightly over 10 minutes, in

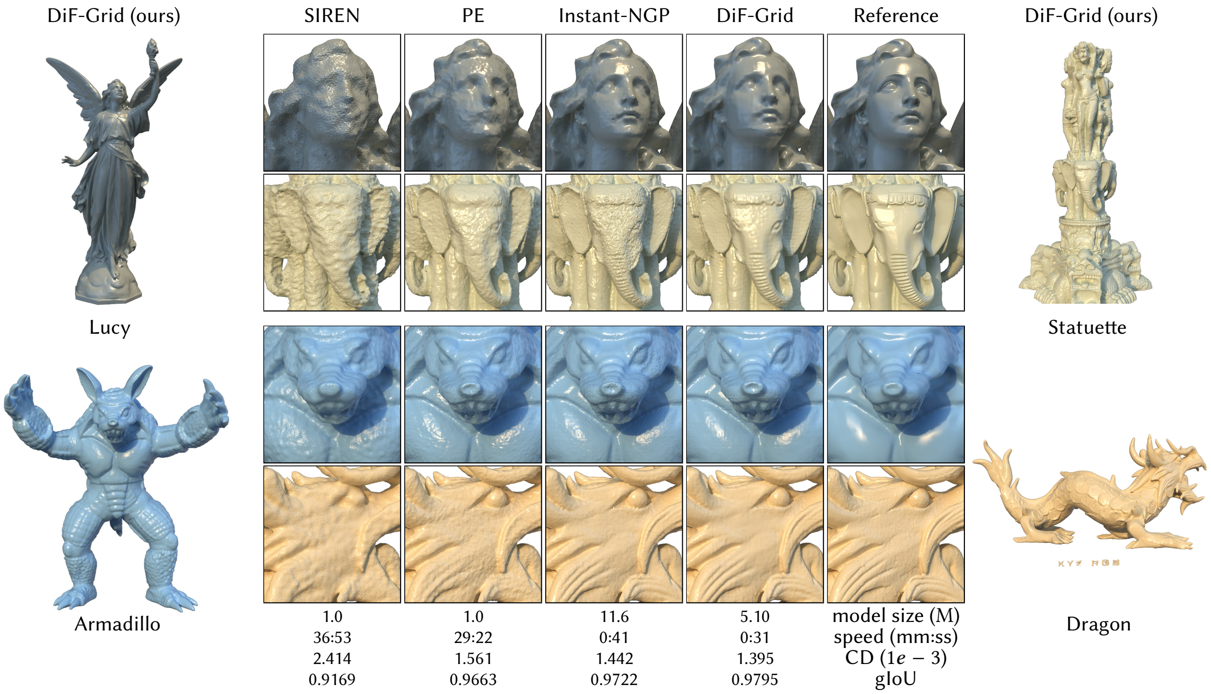


Fig. 5. **Signed-Distance Field Reconstruction.** We reconstruct SDFs from 8.0 M training points. We show qualitative visual comparisons on the top and quantitative comparisons on the bottom including the number of parameters, reconstruction time, gIoU, and chamfer distance (CD). DiF-Grid and Instant-NGP [Müller et al. 2022] are trained for 10k iterations, while SIREN [Sitzmann et al. 2020] and NeRF with Frequency Position Encoding (PE) [Tancik et al. 2020] are trained for 200k iterations.

Table 1. **Novel View Synthesis with Radiance Fields.** We compare our method to previous radiance field reconstruction methods on the Synthetic-NeRF [Mildenhall et al. 2020] and Tanks and Temples [Knapitsch et al. 2017] datasets. We report the scores reported in the original papers whenever available. We also show the average reconstruction time and model size for the Synthetic-NeRF dataset to compare the efficiency of the methods.

			Synthetic-NeRF				Tanks and Temples	
Method	BatchSize	Steps	Time ↓	Size (M)↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑
NeRF [Mildenhall et al. 2020]	4096	300k	~35h	01.25 🟡	31.01	0.947	25.78	0.864
Plenoxels [Fridovich-Keil et al. 2022]	5000	128k	11.4m ⚪	194.5	31.71	0.958	27.43	0.906
DVGO [Sun et al. 2022]	5000	30k	15.0m	153.0	31.95 🟤	0.957	28.41 🟤	0.911 🟤
Instant-NGP [Müller et al. 2022]	10k-85k	30k	03.9m 🟡	11.64 🟤	32.59 ⚪	0.960 🟤	27.09	0.905
TensoRF-VM [Chen et al. 2022]	4096	30k	17.4m	17.95	33.14 🟡	0.963 🟡	28.56 ⚪	0.920 ⚪
DiF-Grid (Ours)	4096	30k	12.2m 🟤	05.10 ⚪	33.14 🟡	0.961 ⚪	29.00 🟡	0.938 🟡

addition to our superior compactness. Additionally, unlike Plenoxels and Instant-NGP which rely on their own CUDA framework for fast reconstruction, our implementation uses the standard PyTorch framework, making it easily extendable to other tasks.

In general, our model leads to state-of-the-art results on all three challenging benchmark tasks with both high accuracy and efficiency. Note that the baselines are mostly single-factor, utilizing either a local field with an identical coordinate transformation (such as DVGO and Plenoxels), or a global field with a many-to-one coordinate transformation (such as Instant-NGP). In contrast, our DiF model is a two-factor method, incorporating both local coefficient and global basis fields, hence resulting in better reconstruction quality and memory efficiency.

4.3 Generalization

Recent advanced neural representations such as NeRF, SIREN, ACORN, Plenoxels, Instant-NGP and TensoRF optimize each signal separately, lacking the ability to model multiple signals jointly or learning useful priors from multiple signals. In contrast, our DiF representation not only enables accurate and efficient per-signal reconstruction (as demonstrated in Section 4.2) but it can also be applied to generalize across signals by simply sharing the basis field across signal instances. We evaluate the benefits of basis sharing by conducting experiments on image regression from partial pixel observations and few-shot radiance field reconstruction. For these experiments, instead of DiF-Grid, we adopt DiF-MLP-B (i.e., (5) in the Table 3) as our DiF representation, where we utilize a tensor grid to model

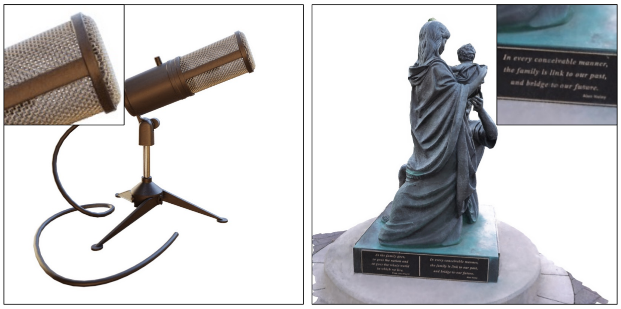

In every conceivable manner,
the family is link to our past,
and bridge to our future.

Fig. 6. **Radiance Field Reconstruction**. We evaluate our DiF using NeRF-Synthetic and Tanks and Temples datasets, our method is able to reconstruct high-quality surface details.

the coefficient and 6 tiny MLPs (two layers with 32 neurons each) to model the basis. We find that DiF-MLP-B performs better than DiF-Grid in the generalization setting, owing to the strong inductive smoothness bias of MLPs.

Image Regression from Sparse Observations. Unlike the image regression experiments conducted in Sec. 4.2 which use all image pixels as observations during optimization, this experiment focuses on the scenario where only part of the pixels are used during optimization. Without additional priors, a single-signal optimization easily overfits in this setting due to the sparse observations and the limited inductive bias, hence failing to recover the unseen pixels.

We use our DiF-MLP-B model to learn data priors by pre-training it on 800 facial images from the FFHQ dataset [Karras et al. 2018] while sharing the MLP basis and projection function parameters. The final image reconstruction task is conducted by optimizing the coefficient grids for each new test image.

In Fig. 7, we show the image regression results on three different facial images with various masks and compare them to baseline methods that do not use any data priors, including Instant-NGP and our DiF-MLP-B without pre-training. As expected, Instant-NGP can accurately approximate the training pixels but results in random noise in the untrained mask regions. Interestingly, even without pre-training and priors from other images, our DiF-MLP-B is able to capture structural information to some extent within the same image being optimized; as shown in the eye region, the model can learn the pupil shape from the right eye and regress the left eye (masked during training) by reusing the learned structures in the shared basis functions. As shown on the right of Fig. 7, our DiF-MLP-B with pre-trained prior clearly achieves the best reconstruction quality with better structures and boundary smoothness compared to the baselines, demonstrating that our factorized DiF model allows for learning and transferring useful prior information from the training set.

Few-Shot Radiance Field Reconstruction. Reconstructing radiance fields from few-shot input images with sparse viewpoints is highly challenging. Previous works address this by imposing sparsity assumptions [Kim et al. 2022; Niemeyer et al. 2022] in per-scene optimization or training feed-forward networks [Chen et al. 2021b; Kulhanek et al. 2022; Yu et al. 2021] from datasets. Here we consider 3 and 5 input views per scene and seek a novel solution that leverages data priors in pre-trained basis fields of our DiF model during the optimization task. It is worth-noting that the views are chosen in a quarter sphere, thus the overlapping region between views is quite limited.

Specifically, we first train DiF models on 100 Google Scanned Object scenes [Downs et al. 2022], which contains 250 views per scene. During cross-scene training, we maintain 100 per-scene coefficients and share the basis $\mathbf{b}$ and projection function $\mathcal{P}$. After cross-scene training, we use the mean coefficient values of pre-trained coefficient fields as the initialization, while fixing the pre-trained functions ($\mathbf{b}$ and $\mathcal{P}$) and fine-tuning the coefficient field for new scenes with few-shot observations. In this experiment, we compare results from both DiF-MLP-B and DiF-Grid with and without the pre-training. We also compare with Instant-NGP and previous few-shot reconstruction methods, including PixelNeRF [Yu et al. 2021] and MVSNeRF [Chen et al. 2021b], re-train with the same training set and test using the same 3 or 5 views. As shown in Table 2 and Fig. 8, our pre-trained DiF representation with MLP basis provides strong regularization for few-shot reconstruction, resulting in fewer artifacts and better reconstruction quality than the single-scene optimization methods without data priors and previous few-shot reconstruction methods that also use pre-trained networks. In particular, without any data priors, single-scene optimization methods (Instant-NGP and ours w/o prior) lead to a lot of outliers due to overfitting to the few-shot input images. Previous methods like MVSNeRF and PixelNeRF achieve plausible reconstructions due to their learned feed-forward prediction which avoids per-scene optimization. However, they suffer from blurry artifacts. Additionally, the strategy taken by PixelNeRF and MVSNeRF assumes a narrow baseline and learns correspondences across views for generalization via feature averaging or cost volume modeling which does not work as effectively in a wide baseline setup. On the other hand, by pre-training shared basis fields on multiple signals, our DiF model can learn useful data priors, enabling the reconstruction of novel signals from sparse observations via optimization.

4.4 Influence of Design Choices in DiF

In this section, we aim to analyze the properties of these variations and offer a comprehensive understanding of the components of the proposed representation. We conduct extensive evaluations on the four main components of our Dictionary Fields: level number L, coordinate transformation function $\boldsymbol{\gamma}$, field representation $\mathbf{c}$ and $\mathbf{b}$, and field connector $\circ$.

We present a comprehensive assessment of the representations' capabilities in terms of efficiency, compactness, reconstruction quality, as well as generalizability, with a range of tasks including 2D image regression (with all pixels), and per-scene and across-scene 3D radiance field reconstruction. Note that, the settings in per-scene and across-scene radiance field reconstruction are the same as introduced in Section 4.2 and Section 4.3, while for the 2D image regression task, we use the same model setting as in Section 4.2 and test on 256 high fidelity images at a resolution of 1024 × 1024 from the DIV2K dataset [Agustsson and Timofte 2017]. To enable meaningful comparisons, we evaluate the variations within the same

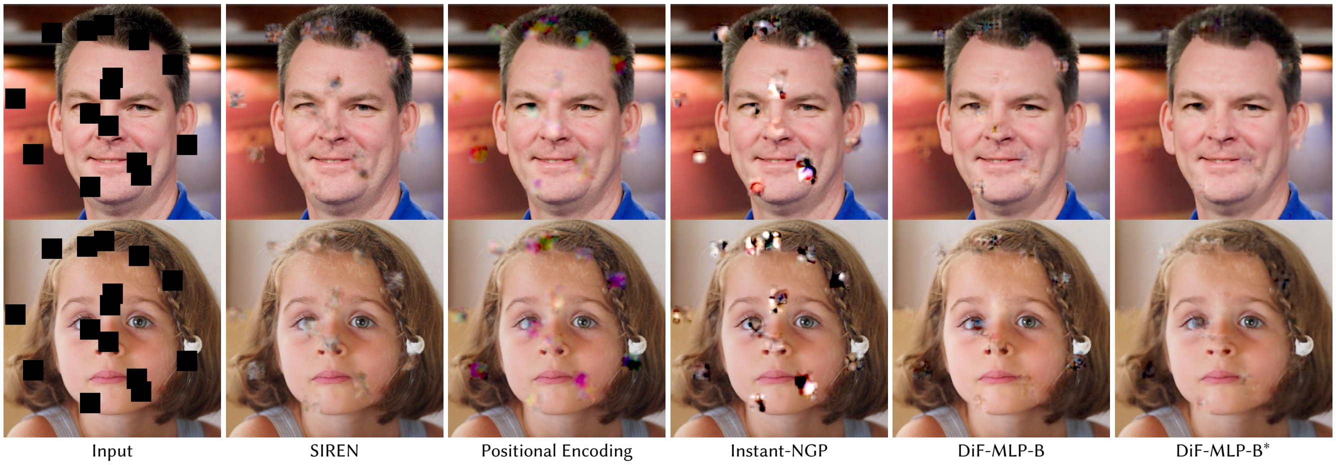


Fig. 7. **Image Regression from Sparse Observations.** Results obtained by fitting each model to all unmasked pixels. We use randomly placed black squares as masks for the bottom two rows and an image of text and small icons as mask for the top row. The symbol * denotes pre-training of the basis factors using the FFHQ facial image set. Our pre-trained model (DiF-MLP-B*) learns robust basis fields which lead to better reconstruction compared to the per-scene baselines Instant-NGP and DiF-MLP-B. Images ©FFHQ Dataset (CC BY-SA 4.0).

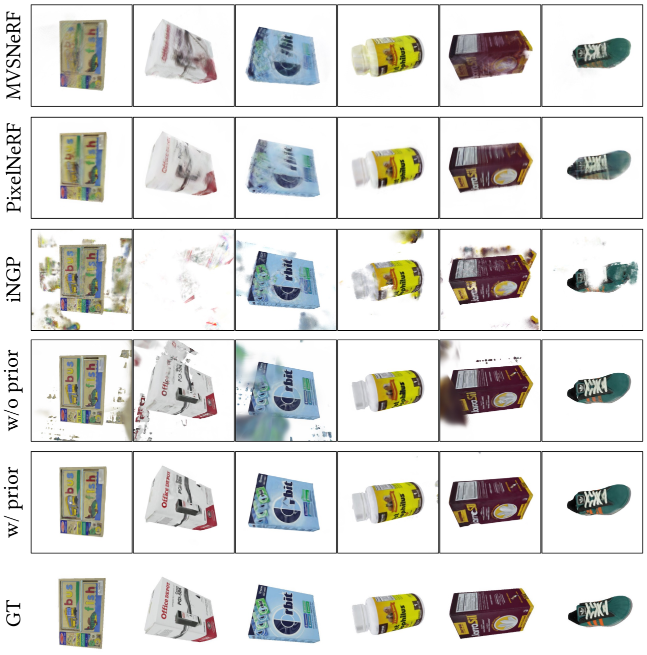


Fig. 8. **Radiance Fields from 5 Views.** We visualize novel view synthesis results of six test scenes, corresponding to the quantitative results in Table 2. We show our DiF-MLP-B model w/ and w/o pre-trained data priors (bottom two rows) and compare it to Instant-NGP, PixelNeRF and MVSNeRF (top three rows). Our model with pre-trained basis factors can effectively utilize the learned data priors, resulting in superior qualitative results with fewer outliers compared to single-scene models (iNGP and ours w/o priors), as well as sharper details compared to feed-forward models (PixelNeRF and MVSNeRF).

Table 2. **Few-shot Radiance Field Reconstruction.** We show quantitative comparisons of few-shot radiance field reconstruction from 3 or 5 viewpoints regarding optimization time and novel view synthesis quality (PSNRs and SSIMs). Results are averaged across 10 test scenes. The results of Instant-NGP and our DiF models are generated based on per-scene optimization, while DiF models with * use pre-trained basis factors across scenes. We train the feed-forward networks of PixelNeRF and MVSNeRF using the same dataset we learn our shared basis factors, and the results of PixelNeRF and MVSNeRF are generated from the networks via direct feed-forward inference. We also fine-tuned the trained models with another $10k$ iterations using the input source views, labeled with "-ft". Our DiF-MLP-B* with pre-trained MLP basis factors leads to the best reconstruction quality.

		3 views		5 views	
Method	Time↓	PSNR↑	SSIM↑	PSNR↑	SSIM↑
iNGP	03:38 ⚪	14.74	0.776	20.79	0.860
DiF-Grid	13:39	18.13	0.805	20.83	0.847
DiF-MLP-B	18:24	16.31	0.804	22.26	0.900 🟤
PixelNeRF	00:00 🟡	21.37 🟤	0.878 🟤	22.73	0.896
MVSNeRF	00:00 🟡	20.50	0.868	22.76	0.891
PixelNeRF-ft	25:18	22.21 🟡	0.882 ⚪	23.67 🟤	0.895
MVSNeRF-ft	13:06 🟤	18.51	0.864	20.49	0.887
DiF-Grid*	13:18 ⚪	20.77	0.871	25.41 ⚪	0.915 ⚪
DiF-MLP-B*	18:44	21.96 ⚪	0.891 🟡	26.91 🟡	0.927 🟡

code base and report their performance using the same number of iterations number, batch size, training point sampling strategy and pyramid frequencies. Correspondingly, the results for Instant-NGP, EG3D, OccNet, NeRF, DVGO, TensoRF-VM/-CP are based on our reimplementation of the original methods in our Dictionary Fields with the corresponding design parameters shown in the tables.

Field Representation **c** *and* **b**. In Table 3, we compare various functions for representing the factors in our framework (especially our DiF model) including MLPs, Vectors, 2D Maps and 3D Grids, encompassing most previous representations. Note that discrete feature grid functions (3D Grids, 2D Maps, and Vectors) generally lead to faster reconstruction than MLP functions (e.g. DiF-Grid is faster

Table 3. Design Study on Field Representations **c** and **b**.

	Design				Performance (PSNR/SSIM)				
	Name	N	Representations	$\boldsymbol{\gamma}$	Time (mm:ss)	Size (M)	2D Images	RF	Few-shot RF
(1)	**TensoRF-VM***	2	2D Maps; Vectors	$\text{Orthog}_{1,2D}(\mathbf{x})$	- /16:20/13:06	- /4.55/4.93	- / -	30.47/0.940	26.79/0.908 ⚪
(2)	**DiF-Grid**	2	3D Grids; 3D Grids	$\text{Sawtooth}(\mathbf{x})$	01:13/**12:10**/11:35	0.99/5.10/7.32	39.51/0.963 🟡	33.14/0.961 🟡	25.41/0.915
(3)	**DiF-DCT**	2	3D Grids; 3D Grids	$\text{Sawtooth}(\mathbf{x})$	00:53/ - / -	**0.18**/ - / -	23.16/0.606	- / -	- / -
(4)	**DiF-Hash-B**	2	Vectors; 3D Grids	$\text{Hashing}(\mathbf{x})$	00:55/13:10/10.15	1.09/4.37/3.28	37.53/0.949 ⚪	32.80/0.960 ⚪	26.53/0.924 🟤
(5)	**DiF-MLP-B**	2	MLP; 3D Grids	$\text{Sawtooth}(\mathbf{x})$	01:24/18:18/18:23	**0.18**/0.62/2.53	28.76/0.819	29.62/0.932	26.91/0.927 🟡
(6)	**DiF-MLP-C**	2	3D Grid; MLP	$\text{Sawtooth}(\mathbf{x})$	01:13/13:38/**08:23**	0.87/4.54/4.86	34.72/0.910 🟤	32.57/0.956 🟤	23.54/0.875
(7)	**TensoRF-CP***	3	Vectors×3	$\text{Orthog}_{1D}(\mathbf{x})$	**00:43**/28:05/12:42	0.39/**0.29**/**0.29**	33.79/0.899	31.14/0.944	22.62/0.867

Table 4. Design Study on Coordinate Transformations $\boldsymbol{\gamma}$, scores are reported in (PSNR/SSIM).

	$\boldsymbol{\gamma}_i(\mathbf{x})$	2D Images	RF	Few-shot RF
(1)	$\text{Hashing}(\mathbf{x})$	37.53/0.949	32.80/0.960	26.62/0.919 🟡
(2)	$\text{Sinusoidal}(\mathbf{x})$	38.21/0.953 🟤	32.85/0.961 🟤	25.43/0.908 ⚪
(3)	$\text{Triangular}(\mathbf{x})$	39.38/0.962 ⚪	32.95/0.960 ⚪	24.78/0.904
(4)	$\text{Sawtooth}(\mathbf{x})$	39.51/0.963 🟡	33.14/0.961 🟡	25.41/0.915 🟤

Table 5. Design Study on Levels Number L, scores are reported in (PSNR/SSIM).

	Name	L	2D Images	RF	Few-shot RF
(1)	OccNet*	1	13.90/0.437	20.60/0.849	- / -
(2)	NeRF*	10	28.99/0.816	27.81/0.919	- / -
(3)	DiF-Hash-B	1	30.97/0.891	31.11/0.941 🟤	24.13/0.881 🟤
(4)	DiF-Hash-B	6	37.53/0.949 🟤	32.80/0.960 ⚪	26.62/0.919 🟡
(5)	DiF-Grid	1	38.73/0.973 ⚪	31.08/0.942	23.88/0.882
(6)	DiF-Grid	6	39.51/0.963 🟡	33.14/0.961 🟡	25.41/0.915 ⚪

Table 6. Quantify comparison on element-wise product ∘ vs. concatenation ⊕, scores are reported in (PSNR/SSIM).

Name		2D Images	RF	Few-shot RF
(1) TensoRF-CP	∘	33.79/0.899 🟡	31.14/0.944 🟡	23.19/0.879 🟡
	⊕	25.67/0.683 ⚪	26.75/0.905 ⚪	21.43/0.856 ⚪
(2) TensoRF-VM	∘	- / -	30.47/0.940 🟡	26.99/0.911 🟡
	⊕	- / -	29.86/0.939 ⚪	24.67/0.885 ⚪
(3) DiF-Grid	∘	39.51/0.963 🟡	33.14/0.961 🟡	25.41/0.915 🟡
	⊕	37.76/0.946 ⚪	32.95/0.960 ⚪	24.71/0.894 ⚪

than DiF-MLP-B and DiF-MLP-C). While all variants can lead to reasonable reconstruction quality for single-signal optimization, our DiF-Grid representation that uses grids for both factors achieves the best performance on the image regression and single-scene radiance field reconstruction tasks. On the other hand, the task of few-shot radiance field reconstruction benefits from basis functions that impose stronger regularization. Therefore, representations with stronger inductive biases (e.g., the Vectors in TensoRF-VM and MLPs in DiF-MLP-B) lead to better reconstruction quality compared to other variants.

Coordinate Transformation $\boldsymbol{\gamma}$. In Table 4, we evaluate four coordinate transformation functions using our DiF representation. These transformation functions include sinusoidal, triangular, hashing and sawtooth. Their transformation curves are shown in Fig. 3. In general, in contrast to the random hashing function, the periodic transformation functions (2, 3, 4) allow for spatially coherent information sharing through repeated patterns, where neighboring points can share spatially adjacent features in the basis fields, hence preserving local connectivity. We observe that the periodic basis achieves clearly better performance in modeling dense signals (e.g., 2D images). For sparse signals such as 3D radiance fields, all four transformation functions achieve high reconstruction quality on par with previous state-of-the-art fast radiance field reconstruction approaches [Chen et al. 2022; Müller et al. 2022; Sun et al. 2022].

Level Number L. Our DiF model adopts multiple levels of transformations to achieve pyramid basis fields, similar to the usage of a set of sinusoidal positional encoding functions in NeRF [Mildenhall et al. 2020]. We compare multi-level models (including DiF and NeRF) with their reduced single-level versions that only use a single transformation level in Table 5. Note that Occupancy Networks (OccNet, row (1)) do not leverage positional encodings and can be seen as a single-level version of NeRF (row (2)) while the model with multi-level sinusoidal encoding functions (NeRF) leads to about 10dB PSNR performance boost for both 2D image and 3D reconstruction tasks. On the other hand, the single-level DiF models are also consistently worse than the corresponding multi-level models in terms of speed and reconstruction quality, despite the performance drops being not as severe as those in purely MLP-based representations.

Field Connector ∘. Another key design choice of our DiF model is to adopt the element-wise product to connect multiple factors. Directly concatenating features from different components is an alternative choice and exercised in several previous works [Chan et al. 2022; Mildenhall et al. 2020; Müller et al. 2022]. In Table 6, we compare the performance of the element-wised product against the direct concatenation in three model variants. Note that the element-wise product consistently outperforms the concatenation operation in terms of reconstruction quality for all models on all applications, demonstrating the effectiveness of using the proposed product-based factorization.

5 CONCLUSION

In this work, we present a novel signal representation – Dictionary Fields (DiF) – that factorizes a signal as a product of localized coefficient field and a global basis field with periodic transformations. We extensively evaluate our DiF model on three signal reconstruction tasks including 2D image regression, 3D SDF reconstruction, and radiance field reconstruction. We demonstrate that our DiF model leads to state-of-the-art reconstruction quality, better or on par with previous methods on all three tasks, while achieving faster reconstruction and more compact model sizes than most methods. Our DiF model is able to generalize across scenes by learning shared basis field factors from multiple signals, allowing us to reconstruct new signals from sparse observations. We show that, using such pre-trained basis factors, our method enables high-quality few-shot radiance field reconstruction from only 3 or 5 views, outperforming previous methods like PixelNeRF and MVSNeRF in the sparse view / wide baseline setting.

Limitations. Currently, our learned coefficient and basis fields are unconstrained and hence not easily interpretable. We imagine that enforcing sparsity more explicitly will lead to more interpretable basis patterns and consider this as interesting future work.

ACKNOWLEDGMENTS

Many thanks to Bozidar Antic, Zehao Yu, Hansheng Chen, Shaofei Wang for helpful discussion and suggestions. This work was supported by the SNF grant 200021, 204840.

REFERENCES

Eirikur Agustsson and Radu Timofte. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study.

Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. 1974. Discrete cosine transform. *IEEE transactions on Computers* (1974).

Kara-Ali Aliev, Dmitry Ulyanov, and Victor S. Lempitsky. 2019. Neural Point-Based Graphics. *arXiv.org* 1906.08240 (2019).

Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. 2022. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*.

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

Sai Bi, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Yannick Hold-Geoffroy, David Kriegman, and Ravi Ramamoorthi. 2020b. Deep reflectance volumes: Relightable reconstructions from multi-view photometric images. (2020).

Mark Boss, Raphael Braun, Varun Jampani, Jonathan T Barron, Ce Liu, and Hendrik Lensch. 2021a. Nerd: Neural reflectance decomposition from image collections. In *ICCV*.

Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan Barron, and Hendrik Lensch. 2021b. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *NeurIPS* (2021).

Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. 2020. Deep Local Shapes: Learning Local SDF Priors for Detailed 3D Reconstruction. In *ECCV*.

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. 2022. Efficient Geometry-aware 3D Generative Adversarial Networks. In *CVPR*.

Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. 2021. Pi-GAN: Periodic Implicit Generative Adversarial Networks for 3D-Aware Image Synthesis. In *CVPR*.

Anpei Chen, Zexiang Xu, Andreas Geiger, Jingyi Yu, and Hao Su. 2022. TensoRF: Tensorial Radiance Fields. In *ECCV*.

Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. 2021b. MVSNeRF: Fast Generalizable Radiance Field Reconstruction from Multi-View Stereo. In *ICCV*.

Yinbo Chen, Sifei Liu, and Xiaolong Wang. 2021a. Learning Continuous Image Representation With Local Implicit Image Function. In *CVPR*.

Zhiqin Chen and Hao Zhang. 2019. Learning Implicit Fields for Generative Shape Modeling. In *CVPR*.

Ricardo L. de Queiroz and Philip A. Chou. 2016. Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform. *TIP* (2016).

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. 2022. Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items. *arXiv.org* 2204.11918 (2022).

Michael Elad, J-L Starck, Philippe Querre, and David L Donoho. 2005. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and computational harmonic analysis* (2005).

Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. 2021. Multiplicative Filter Networks. In *ICLR*.

Sara Fridovich-Keil, Giacomo Meanti, Frederik Warburg, Benjamin Recht, and Angjoo Kanazawa. 2023. K-Planes: Explicit Radiance Fields in Space, Time, and Appearance. *arXiv.org* 2301.10241 (2023).

Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. 2022. Plenoxels: Radiance Fields without Neural Networks. In *CVPR*.

Xueyang Fu, Zheng-Jun Zha, Feng Wu, Xinghao Ding, and John Paisley. 2019. JPEG Artifacts Reduction via Deep Convolutional Sparse Coding. In *ICCV*.

Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. 2022. GET3D: A Generative Model of High Quality 3D Textured Shapes Learned from Images. In *Advances In Neural Information Processing Systems*.

Alexander Grossmann and Jean Morlet. 1984. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM journal on mathematical analysis* (1984).

Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. 2015. Fast and flexible convolutional sparse coding. In *CVPR*.

Binbin Huang, Xinhao Yan, Anpei Chen, Shenghua Gao, and Jingyi Yu. 2022. PREF: Phasorial Embedding Fields for Compact Neural Representations. *arXiv.org* 2205.13524 (2022).

Ka-Hei Hui, Ruihui Li, Jingyu Hu, and Chi-Wing Fu. 2022. Neural Wavelet-domain Diffusion for 3D Shape Generation. In *ACM Trans. on Graphics*, Soon Ki Jung, Jehee Lee, and Adam W. Bargteil (Eds.).

Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. 2020. Local Implicit Grid Representations for 3D Scenes. In *CVPR*.

Tero Karras, Samuli Laine, and Timo Aila. 2018. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv.org* (2018).

Mijeong Kim, Seonguk Seo, and Bohyung Han. 2022. InfoNeRF: Ray Entropy Minimization for Few-Shot Neural Volume Rendering. In *CVPR*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. 2017. Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction. *ACM Trans. on Graphics* 36, 4 (2017).

Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. 2022. Decomposing NeRF for Editing via Feature Field Distillation. In *NIPS*.

Jonas Kulhanek, Erik Derner, Torsten Sattler, and Robert Babuska. 2022. ViewFormer: NeRF-Free Neural Rendering from Few Images Using Transformers. In *ECCV*, Shai Avidan, Gabriel J. Brostow, Moustapha Cisse, Giovanni Maria Farinella, and Tal Hassner (Eds.).

Alexandr Kuznetsov, Krishna Mullia, Zexiang Xu, Miloš Hašan, and Ravi Ramamoorthi. 2021. NeuMIP: multi-resolution neural materials. *ACM Trans. on Graphics* (2021).

Daniel Lee and H Sebastian Seung. 2000. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems* (2000).

Tianye Li, Mira Slavcheva, Michael Zollhoefer, Simon Green, Christoph Lassner, Changil Kim, Tanner Schmidt, Steven Lovegrove, Michael Goesele, and Zhaoyang Lv. 2021. Neural 3D Video Synthesis. *arXiv.org* 2103.02597 (2021).

Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. 2020. Neural Scene Flow Fields for Space-Time View Synthesis of Dynamic Scenes. *arXiv.org* 2011.13084 (2020).

David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. 2022. Bacon: Band-limited Coordinate Networks for Multiscale Scene Representation. In *CVPR*.

Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. 2020. Neural Sparse Voxel Fields. In *NeurIPS*.

Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. 2019. Neural Volumes: Learning Dynamic Renderable Volumes from Images. In *ACM Trans. on Graphics*.

Julien Mairal, Francis R. Bach, Jean Ponce, and Guillermo Sapiro. 2009. Online dictionary learning for sparse coding. In *ICML*, Andrea Pohoreckyj Danyluk, Léon Bottou, and Michael L. Littman (Eds.).

Matthew, Aditya Kusupati, Alex Fang, Vivek Ramanujan, Aniruddha Kembhavi, Roozbeh Mottaghi, and Ali Farhadi. 2023. Neural Radiance Field Codebooks.

arXiv.org 2301.04101 (2023).

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy Networks: Learning 3D Reconstruction in Function Space. In *CVPR*.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2020. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*.

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

Michael Niemeyer, Jonathan Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. 2022. RegNeRF: Regularizing Neural Radiance Fields for View Synthesis from Sparse Inputs. In *CVPR*.

Michael Niemeyer, Lars M. Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable Volumetric Rendering: Learning Implicit 3D Representations without 3D Supervision. In *CVPR*.

Bruno A Olshausen and David J Field. 1996. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. (1996).

Bruno A Olshausen and David J Field. 1997. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision research* (1997).

Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. 2019. DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation. In *CVPR*.

Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable neural radiance fields. In *ICCV*.

Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. 2022. BEVSegFormer: Bird's Eye View Semantic Segmentation From Arbitrary Camera Rigs. *arXiv.org* 2203.04050 (2022).

Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *CVPR*.

Gilles Rainer, Wenzel Jakob, Abhijeet Ghosh, and Tim Weyrich. 2019. Neural btf compression and interpolation. In *Computer Graphics Forum*. Wiley Online Library.

Sameera Ramasinghe and Simon Lucey. 2021. Learning Positional Embeddings for Coordinate-MLPs. *arXiv.org* 2112.11577 (2021).

Sameera Ramasinghe, Violetta Shevchenko, Gil Avraham, and Anton Van Den Hengel. 2023. BaLi-RF: Bandlimited Radiance Fields for Dynamic Scene Modeling. *arXiv.org* 2302.13543 (2023).

Ron Rubinstein, Michael Zibulevsky, and Michael Elad. 2008. *Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit*. Technical Report. Computer Science Department, Technion.

Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. 2020. GRAF: Generative Radiance Fields for 3D-Aware Image Synthesis. In *NeurIPS*.

Shayan Shekarforoush, David Lindell, David J Fleet, and Marcus A Brubaker. 2022. Residual multiplicative filter networks for multiscale reconstruction. (2022).

Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. 2020. Implicit Neural Representations with Periodic Activation Functions. In *NIPS*.

Liangchen Song, Anpei Chen, Zhong Li, Zhang Chen, Lele Chen, Junsong Yuan, Yi Xu, and Andreas Geiger. 2023. NeRFPlayer: Streamable Dynamic Scene Representation with Decomposed Neural Radiance Fields. *TVCG* (2023).

Cheng Sun, Min Sun, and Hwann-Tzong Chen. 2022. Direct Voxel Grid Optimization: Super-fast Convergence for Radiance Fields Reconstruction. *CVPR* (2022).

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

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

Danhang Tang, Mingsong Dou, Peter Lincoln, Philip L. Davidson, Kaiwen Guo, Jonathan Taylor, Sean Ryan Fanello, Cem Keskin, Adarsh Kowdle, Sofien Bouaziz, Shahram Izadi, and Andrea Tagliasacchi. 2018. Real-time compression and streaming of 4D performances. *ACM Trans. on Graphics* (2018).

Justus Thies, Michael Zollhöfer, and Matthias Nießner. 2019. Deferred neural rendering: image synthesis using neural textures. *ACM Trans. on Graphics* (2019).

Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T. Barron, and Pratul P. Srinivasan. 2022. Ref-NeRF: Structured View-Dependent Appearance for Neural Radiance Fields. *CVPR* (2022).

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. In *NeurIPS*.

John Wright, Allen Y. Yang, Arvind Ganesh, S. Shankar Sastry, and Yi Ma. 2009. Robust Face Recognition via Sparse Representation. (2009).

Zhijie Wu, Yuhe Jin, and Kwang Moo Yi. 2022. Neural Fourier Filter Bank. *arXiv.org* 2212.01735 (2022).

Yinghao Xu, Sida Peng, Ceyuan Yang, Yujun Shen, and Bolei Zhou. 2022. 3D-aware Image Synthesis via Learning Structural and Textural Representations. *CVPR* (2022).

Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. 2010. Image Super-Resolution Via Sparse Representation. *TIP* (2010).

Jianchao Yang, Kai Yu, Yihong Gong, and Thomas S. Huang. 2009. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*.

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. 2021. Volume rendering of neural implicit surfaces. In *NeurIPS*.

Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. 2021. pixelNeRF: Neural Radiance Fields From One or Few Images. In *CVPR*.

Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. 2022. MonoSDF: Exploring Monocular Geometric Cues for Neural Implicit Surface Reconstruction. In *NeurIPS*.

Jimuyang Zhang and Eshed Ohn-Bar. 2021. Learning by Watching. In *CVPR*.

Yanan Zhang, Jiaxin Chen, and Di Huang. 2022. CAT-Det: Contrastively Augmented Transformer for Multi-modal 3D Object Detection. In *CVPR*.

Hongyi Zheng, Hongwei Yong, and Lei Zhang. 2021. Deep Convolutional Dictionary Learning for Image Denoising. In *CVPR*.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo magnification: learning view synthesis using multiplane images. *ACM Trans. on Graphics* (2018).

Junqiu Zhu, Yaoyi Bai, Zilin Xu, Steve Bako, Edgar Velázquez-Armendáriz, Lu Wang, Pradeep Sen, Miloš Hašan, and Ling-Qi Yan. 2021. Neural complex luminaires: representation and rendering. *ACM Trans. on Graphics* (2021).