DINER: Disorder-Invariant Implicit Neural Representation

Shaowen Xie\textsuperscript{1,1}, Hao Zhu\textsuperscript{1,†}, Zhen Liu\textsuperscript{1,†}, Qi Zhang\textsuperscript{2}, You Zhou\textsuperscript{1}, Xun Cao\textsuperscript{1}, Zhan Ma\textsuperscript{1}

\textsuperscript{1} School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China
\textsuperscript{2} AI Lab, Tencent Company, Shenzhen 518054, China

\textsuperscript{†} Equal contribution. Corresponding author: caoxun@nju.edu.cn

Abstract

Implicit neural representation (INR) characterizes the attributes of a signal as a function of corresponding coordinates which emerges as a sharp weapon for solving inverse problems. However, the capacity of INR is limited by the spectral bias in the network training. In this paper, we find that such a frequency-related problem could be largely solved by re-arranging the coordinates of the input signal, for which we propose the disorder-invariant implicit neural representation (DINER) by augmenting a hash-table to a traditional INR backbone. Given discrete signals sharing the same histogram of attributes and different arrangement orders, the hash-table could project the coordinates into the same distribution for which the mapped signal can be better modeled using the subsequent INR network, leading to significantly alleviated spectral bias. Experiments not only reveal the generalization of the DINER for different INR backbones (MLP vs. SIREN) and various tasks (image/video representation, phase retrieval, and refractive index recovery) but also show the superiority over the state-of-the-art algorithms both in quality and speed. Project page: https://ezio77.github.io/DINER-website/

1. Introduction

INR [34] continuously describes a signal, providing the advantages of Nyquist-sampling-free scaling, interpolation, and extrapolation without requiring the storage of additional samples [20]. By combining it with differentiable physical mechanisms such as the ray-marching rendering [12, 22], Fresnel diffraction propagation [44] and partial differential equations [11], INR becomes a universal and sharp weapon for solving inverse problems and has achieved significant progress in various scientific tasks, e.g., the novel view synthesis [39], intensity diffraction tomography [19] and multiphysics simulation [11].

However, the capacity of INR is often limited by the underlying network model itself. For example, the spectral bias [28] usually makes the INR easier to represent low-frequency signal components (see Fig. 1 and Fig. 2(c)). To improve the representation capacity of the INR model, previous explorations mainly focus on encoding more frequency bases using either Fourier basis [22, 34, 38, 42] or wavelet basis [6, 17] into the network. However, the length of function expansion is infinite in theory, and a larger model with more frequency bases runs exceptionally slowly.

Such a problem is closely related to the input signal’s frequency spectrum. The signal’s frequency tells how fast the signal attribute changes following the intrinsic order of geometry coordinates. By properly re-arranging the order of coordinates of a discrete signal, we could modulate the signal’s frequency spectrum to possess more low-frequency components. Such a re-arranged signal can be better modeled using subsequent INR. Based on this observation, we propose the DINER. In DINER, the input coordinate is first mapped to another index using a hash-table and fed into a traditional INR backbone. We prove that no matter what orders the elements in the signal are arranged, the joint optimization of the hash-table and the network parameters guarantees the same mapped signal (Fig. 2(d)) with more
low-frequency components. As a result, the representation capacity of the existing INR backbones and the task performance are largely improved. As in Fig. 1 and Fig. 2(e), a tiny shallow and narrow MLP-based INR could well characterize the input signal with arbitrary arrangement orders. The use of hash-table trades the storage for fast computation where the caching of its learnable parameters for mapping is usually at a similar size as the input signal, but the computational cost is marginal since the back-propagation for the hash-table derivation is $O(1)$.

Main contributions are summarized as follows:

1. The inferior representation capacity of the existing INR model is largely increased by the proposed DINER, in which a hash-table is augmented to map the coordinates of the original input for better characterization in the succeeding INR model.

2. The proposed DINER provides consistent mapping and representation capacity for signals sharing the same histogram of attributes and different arrangement orders.

3. The proposed DINER is generalized in various tasks, including the representation of 2D images and 3D videos, phase retrieval in lensless imaging, as well as 3D refractive index recovery in intensity diffraction tomography, reporting significant performance gains to the existing state-of-the-arts.

2. Related Work

2.1. INR and inverse problem optimization

INR (sometimes called coordinate neural network) builds the mapping between the coordinate and its signal value using a neural network, promising continuous and memory-efficient modeling for various signals such as the 1D audio [8], 2D image [38], 3D shape [26], 4D light field [35] and 5D radiance field [22]. Accurate INR for these signals could be supervised directly by comparing the network output with the ground truth, or an indirect way that calculates the loss between the output after differentiable operators and the variant of the ground truth signal. Thus, INR becomes a universal tool for solving inverse problems because the forward processes in these problems are often well-known. INR has been widely applied in the optimization of inverse problems in several disciplines, such as computer vision and graphics [39], computational physics [11], clinical medicine [41], biomedical engineering [19,44], material science [5] and fluid mechanics [29,30].

2.2. Encoding high-frequency components in INR

Most INRs utilize the MLP as function approximators for characterizing the signal’s attributes. According to the approximation theory, an MLP network could approximate any function [14]. However, there is a spectral bias [28] in the network training, resulting in low performance of IN-
R for high-frequency components. Several attempts have been explored to overcome this bias and could be classified into two categories, i.e., the function expansion and the parametric encodings.

The function expansion idea treats the INR fitting as a function approximation using different bases. Mildenhall et al. [22] encoded the input coordinates using a series of \( \sin / \cos \) functions with different frequencies and achieved great success in radiance field representation. The strategy of frequency-predefined \( \sin / \cos \) functions is further improved with random Fourier features, which has been proved to be effective in learning high-frequency components both in theory and in practical [38]. Sitzmann et al. [34] replaced the classical ReLU activation with periodic activation function (SIREN). The different layers of the SIREN could be viewed as increasing different frequency supports of a signal [42]. SIREN is suited for representing complex natural signals and their derivatives. Apart from the Fourier expansion, Fathony et al. [6] represented a complex signal by a linear combination of multiple wavelet functions (MFN), where the high-frequency components could be well modeled by modulating the frequency in the Gabor filter. Lindell et al. [17] developed MFN and proposed the band-limited coordinate networks, where the frequency at each network layer could be specified at initialization. These methods have achieved significant advantages in representing high-frequency components compared with the standard MLP network. However, the performance of these INRs are limited by the frequency distribution of a signal-self, and often require a deeper or wider network architecture to improve the fitting accuracy.

From the perspective of the parameter encoding \([3, 10, 18, 37]\), each input coordinate is encoded using learned features which are fed into an MLP for fitting. Takikawa et al. [37] divided the 3D space using a sparse voxel octree structure where each point is represented using a learnable feature vector from its eight corners, achieving real-time rendering of high-quality signed distance functions. Mariel et al. [20] divided the coordinate space iteratively during the INR training (ACORN), where the encoding features for each local block are obtained by a coordinate encoder network and are fed into a decoder network to obtain the attribute of a signal. ACORN achieves nearly 40 dB P-SNR for fitting gigapixel images for the first time. Muller et al. [24] replaced the coordinate encoder network with a multi-resolution hash-table. Because the multi-resolution hash-table has higher freedom for characterizing coordinates’ features, only a tiny network is used to map the features and the attribute values of a signal. Despite the superiority of faster convergence and higher accuracy in parameter encoding, one of the key questions is still not answered, i.e., what are the geometrical meanings of these features?

Compared with these methods, the hash-table in the proposed DINDER unambiguously projects the input coordinate into another, or in other words, mapping the signal into the one with more low-frequency components. As a result, a tiny network could achieve very high accuracy compared with previous methods.

3. Performance of INR

3.1. Background of expressive power of INR

Following the form of Yuce et al. [42], an INR with a 1D input \( x \) could be modeled as a function \( f_\theta(x) \) that maps the input coordinate \( x \) to its attribute, that

\[
\begin{align*}
\mathbf{z}^0 &= \gamma(x), \\
\mathbf{z}^l &= \rho^l(\mathbf{W}^l \mathbf{z}^{l-1} + \mathbf{b}^l), \quad l = 1, ..., L - 1, \\
f_\theta(x) &= \mathbf{W}^L \mathbf{z}^{L-1} + \mathbf{b}^L
\end{align*}
\]

where \( \gamma(\cdot) \) is the preprocess function which is often used to encode more frequency bases in the network, \( \mathbf{z}^l \) is the output of the \( l \)-th layer in INR, \( \rho \) is the activation function, \( \mathbf{W}^l \) and \( \mathbf{b}^l \) are the weight and bias matrix in the \( l \)-th layer, \( L \) is the number of layers in INR, \( \theta = \{\mathbf{W}^l, \mathbf{b}^l\}^L \) refers to the set of all training parameters in the network.

This INR could represent signals with functions following the form

\[
f(x) = \sum_{\omega \in \mathcal{H}_\Omega} c_{\omega} \sin(\omega (\mathbf{x}) + \phi_{\omega}),
\]

where \( \mathcal{H}_\Omega \) is the frequency set [42] determined by the frequency selected in the preprocess function \( \gamma(\cdot) \), e.g., the Fourier encoding [38], or the sin activation [34]. In other words, the expressive power of INR is restricted to functions that can be represented using a linear combination of certain harmonics of the \( \gamma(\cdot) \) [42].

3.2. Arrangement order of a signal determines the capacity of INR

According to the expressive power of an INR (Eqn. 2), a signal could be well learned when the encoded frequencies in the INR are consistent with the signal’s frequency distribution. However, there are two problems in applying this conclusion,

1. The frequency distribution of a signal could not be known in advance, especially in inverse problems. Thus proper frequencies could not be well set in designing the architecture of an INR.

2. Due to the spectral bias in network training [28], the low-frequency components in a signal will be learned first, while the high-frequency components are learned in an extremely slow convergence [2, 9, 32, 38].

We notice that most of the signals recorded or to be inversely solved today are discrete signals. The frequency distribution of a discrete signal could be changed by arranging
elements in different orders at the cost of additional storage for the arrangement rule, resulting in different satisfactions of Eqn. 2. Consequently, the capacity of an INR for representing a signal changes with different arrangement orders.

Fig. 2 gives an intuitive demonstration. The Baboon image is arranged in different orders in Fig. 2(a), and the corresponding Fourier spectrum images are shown in Fig. 2(b). The original image contains rich low-, intermediate- and high-frequency information. By sorting the Baboon according to the intensities of pixels, the high-frequency information in y-axis almost disappeared. We then arrange the pixels using a random order. Currently, the Baboon contains much high-frequency information. Then a PE+MLP network is set as the same length N as the number of elements in Y, mostly the width of H_M is set as d_{in} (the dimensions of  \vec{x}_i). Firstly, the input coordinate  \vec{x}_i is used to query the i-th hash key M(  \vec{x}_i) in the H_M. Then the mapped coordinate M(  \vec{x}_i) is fed into a standard MLP. All parameters in the H_M are set as learnable, i.e., parameters in H_M and the network parameters will be jointly optimized during the training process. Fig. 3 demonstrates the above process (The Lego Knight used comes from [15].)

Due to the hash-table, the MLP network actually learns the mapped signal. Fig. 2(d) shows the mapped pixels of the Baboon after hash-table, it is noticed that the original grid coordinates (Fig. 2(a)) are projected into irregular points (Fig. 2(d)). We sample a mesh evenly according to the minimum and the maximum values in the mapped coordinates, and the training process. Fig. 3 demonstrates the above process (The Lego Knight used comes from [15].). First, the input coordinate  \vec{x}_i is fed into a standard MLP. All parameters in the H_M are set as learnable, i.e., parameters in H_M and the network parameters will be jointly optimized during the training process. Fig. 3 demonstrates the above process (The Lego Knight used comes from [15].) 

4. Disorder-invariant INR

4.1. Hash-mapping for INR

Given paired discrete signal Y = \{(  \vec{x}_i,  \vec{y}_i)\}_{i=1}^N, where  \vec{x}_i be the i-th d_{in}-dimensional coordinate, and  \vec{y}_i be the corresponding d_{out}-dimensional signal attribute. Following the analysis mentioned above, an ideal arrangement rule  M^* : \mathbb{R}^{d_{in}} \rightarrow \mathbb{R}^{d_{in}} should meets the following rule,

\[ M^* = \arg \min_{M} \sum_{k=1}^{K_M} |\omega_k| \]

\[ \Omega_M \subseteq \mathcal{H}_M, \quad \Omega_M = \{\omega_k\}_{k=1}^{K_M} \]

where  \Omega_M is the set of frequency by mapping the signal following the M,  \mathcal{H}_M is the supported frequency set of the INR network (Eqn. 2), K_M is the number of frequency in the arranged signal, |.| returns the absolute value of . The signal with this arrangement could be well learned since both the problems of improper frequency setting (Eqn. 3b) and the spectral bias (Eqn. 3a) are taken into account.

However, this strategy requires prior knowledge of the signal distribution, which is only suitable for the compression task. In contrast, it loses the ability to optimize inverse problems where the signal distribution to be optimized could not be achieved in advance. In this subsection, we detail the proposed DINER to handle this problem.

We specifically design a full-resolution hash-table H_M to model the mapping mentioned above. The hash-table

\[ |\cdot| \]

\[ \theta^* = \arg \min_{\theta} \mathcal{L} \left( \mathcal{P} \left( \{  \vec{f}_0(  \vec{x}_i)\}_i^N \right), \mathcal{P} \left( \{  \vec{y}_i\}_i^N \right) \right) \]

\[ \theta^*, H_M^* = \arg \min_{\theta, H_M} \mathcal{L} \left( \mathcal{P} \left( \{  \vec{f}_0(  H_M(  \vec{x}_i))\}_i^N \right), \mathcal{P} \left( \{  \vec{y}_i\}_i^N \right) \right) \]

\[ \theta^*, H_M^* = \arg \min_{\theta, H_M} \mathcal{L} \left( \mathcal{P} \left( \{  \vec{f}_0(  H_M(  \vec{x}_i))\}_i^N \right), \mathcal{P} \left( \{  \vec{y}_i\}_i^N \right) \right) \]

where  \mathcal{P} is a physical process and is an identical transformation for the representation task,  \mathcal{L} is the loss function according to the measurements and the reconstructed results,  \mathcal{H}_M is the i-th key in the hash-table. Because the hash-index operation has no gradient, the first equation in Eqn. 4b could be simplified to the second one in Eqn. 4b.

It is noticed that paired relationship between the coordinate  \vec{x}_i and value  \vec{y}_i in Eqn. 4a is broken in the Eqn. 4b. There is only one independent variable  \vec{y}_i in the loss function Eqn. 4b. Assuming parameters in  \theta are initialized with the same values, all keys in H_M are set with the same one.
(e.g., 0) in every experiment, and the batch size is set as \( N \), when applying a different order to the signal \( Y \), e.g., \( Y' = (\bar{x}_j, \bar{y}_i)_{j=1,i=1}^{N} \), the training of \( \theta \) and \( \mathcal{H}_M \) for signals \( Y \) and \( Y' \) share the same optimization progress in every gradient update since all parameters in Eqn. 4b for \( Y' \) are equivalent to ones for \( Y'' \). As a result, the same \( \theta'' \) will be optimized while the \( \mathcal{H}_M' \) of \( Y' \) is also an inverse arrangement of the \( \mathcal{H}_M \) of \( Y \). This equivalence is not limited to the \( Y' \) with an inverse order; actually, it could be easily proved that the equivalence holds for \( Y \) with an arbitrary order.

Fig. 2(d), (e), and (f) illustrate this equivalence. Although the Baboon is arranged with different orders, the hash-table maps them into the same signal (Fig. 2(d), (f)), and DINER optimizes them with similar PSNR values 31.03, 31.08, 31.05. In summary:

**Proposition 2.** The DINER is disorder-invariant, and signals with the same histogram distribution of attributes share an optimized network with the same parameter values.

### 4.3. Discussions

**Backbone network.** The backbone of the proposed DINER is not limited to the standard MLP used above; actually, other network structures such as the SIREN could also be integrated with the hash-table and get better performance than the original structure. Please refer to the experimental section for more details.

**Complexity.** Although the number of parameters of the hash-table is much larger than the network, the training cost is very small because only one hash-key needs to be updated for training an MLP with batchsize 1. As a result, the computational complexity of training hash-table is \( O(1) \) in each iteration of training. Note that, the \( O(1) \) complexity holds when the size of hash-table is small. With the increase of the size of the hash-table, more training time are used because the communication cost between the memory and the cache in GPU is increased (see Tabs. 2, 3 for more details).

### 5. Experiments

To verify the performance of the proposed DINER, four separate tasks are conducted: 2D image fitting, neural representation for 3D imaging, phase retrieval in lensless imaging, and 3D Refractive Index recovery in intensity diffraction tomography.

#### 5.1. 2D Image Fitting

**5.1.1 Dataset and Algorithm Setup**

The 2D image fitting task is adopted to test the performance of the proposed DINER and to illustrate the change of frequency distribution of the signal. We use 30 high-resolution images with 1200 × 1200 resolution from the SAMPLING category of the TESTIMAGES dataset [1] to evaluate the performance of various algorithms. Each image is generated using custom Octave/MATLAB software scripts specifically written to guarantee the precise positioning and value of every pixel and contains rich low-, intermediate- and high-frequency information. In the following experiments, all our results are obtained with the same network configuration 2 × 64 unless otherwise stated.

#### 5.1.2 Comparisons of frequency distribution with and without hash-table

As noticed in the Sec. 4.1 and the Figs. 2, the hash-table maps the original signal with more low-frequency contents. Tab. 1 provides statistics of the mean frequency distribution of the original image and the learned INR over 30 images. The ratios of the intermediate- and high-frequency information are all reduced after mapping, while the low-frequency information are increased.

<table>
<thead>
<tr>
<th>Freq. bands</th>
<th>Original Image</th>
<th>Learned INR (MLP)</th>
<th>Learned INR (SIREN)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.4426</td>
<td>0.2484</td>
<td>0.1753</td>
</tr>
<tr>
<td></td>
<td>0.1337</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Figure 4. Comparisons of learned INRs in the DINER with MLP and SIREN backbones, respectively.](figure)

Table 1. Comparisons of the ratio of frequency distributions between the original image and the learned INR. Two backbones, i.e., the MLP and SIREN, are both compared.
5.1.3 Comparisons with the State-of-the-arts

We compare the proposed DINER with the Fourier feature positional encoding (PE+MLP) [38], SIREN [34] and InstantNGP [24]. Noting that, two backbones, i.e., the standard MLP with ReLU activation and the SIREN with periodic function activation, are all combined with the proposed hash-table to better evaluate the performance. We control the size of the hash-table used in the InstantNGP to guarantee the similar parameters with ours, e.g., $2^{21}$ in the 2D image fitting task while ours has a length of $1200 < 2^{21}$. Apart from this, all 5 methods are trained with the same $L_2$ loss between the predicted value and the ground truth, and other parameters are set with the default values by authors.

Fig. 1 shows the PSNR of various methods at different epochs. It is noticed that the SIREN and PE+MLP converges quickly at the early stage and reaches about 24dB and 21dB finally. On the contrary, the proposed two methods both provide higher accuracy than backbones. The PSNRs of two backbones for image fitting are increased 14dB and 16dB using the hash-table, respectively. Additionally, although the InstantNGP converges very fast to about 30dB at about 200 epochs, the curve tends to be stable at the last 2800 epochs. The proposed DINER with MLP backbone achieves an advantage of 5dB than the InstantNGP.

Fig. 5 shows the qualitative results at 3000 epochs. The proposed methods outperform the SIREN and PE+MLP. Our methods provide more clear details especially in high-frequency boundaries, such as the bell tower (yellow box) in Fig. 5. The fitted image of the InstantNGP is very similar to the GT at first sight, however many noises appear in the zoom-in results, for example there are a lot of noisy points in the wall (red box) and the sky (yellow box) of Fig. 5, results in a lower PSNR metric.

Tab. 2 lists the training time of 5 methods. The InstantNGP is implemented with the tiny-cuda-nn [23], while other 4 methods are implemented with the Pytorch. All 5 methods are trained on a NVIDIA A100 40GB GPU. The optimization of hash-table requires additional 3 seconds on the DINER (SIREN) and DINER (MLP) backbone, verifying the low complexity of optimizing hash-table.

5.2. Neural Representation for 3D video

Video describes a dynamic 3D scene $I(t, x, y)$ composed of multiple frames. Accurate representation for 3D video is becoming a popular task [4, 13, 16, 31] in the community of INR. We compare the proposed DINER with the SIREN [34] and the state-of-the-art INR NeRV [4]. Noting only the Hash+SIREN architecture is evaluated in this task. The NeRV is implemented using their default parameters, while the SIREN and the proposed Hash+SIREN both use the same network structure with the size $4 \times 64$. All three methods are evaluated on the videos ‘BasketballPass’ [36], ‘ReadySetGo’ and ‘ShakeNDry’ of the UVG dataset [21] and are trained with 500 epochs. The first 30 frames with 1920×1080 resolution are used in our experiment.

Fig. 6(a) and (b) show the mapping of the coordinates with and without mapping. Fig. 6(c) illustrates the learned INR. It is noticed that the low-frequency property also appears in the learned INR of the 3D video in our DINER. Tab. 3 shows the quantitative comparisons. The proposed method outperforms the NeRV both in quality and speed with 14 dB and $5 \times$ improvements, respectively. Because...
5.3. Phase Recovery in Lensless Imaging

Lensless imaging [25] observes specimen in a very close distance without any optical lens. By directly recording a large hash-table is used in DINER, more time are taken in the transmission between the memory and the cache in the GPU, resulting more training time of DINER than the SIREN. Fig. 7 shows the qualitative comparisons. Noting that the SIREN with a tiny network could not provide reasonable representation for the ‘ReadySetGo’ data with the SIREN. Fig. 7 shows the qualitative comparisons. Notational video is mapped with little high-frequency component ever the high-frequency details are lost such as the character ‘U’ (left-top corner) and the red logo of horsehead (right-top corner) in the purple box, as well as the folds of the trousers in the orange box. On the contrary, the original video is mapped with little high-frequency component in the proposed DINER. As a result, the details mentioned above could be well represented.

Table 3. Comparisons of training 3D videos ‘ReadySetGo’ and ‘ShakeNDry’ with 500 epochs.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Network Para.</th>
<th>Training time</th>
<th>PSNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIREN</td>
<td>8.77K</td>
<td>706s</td>
<td>21.22 dB</td>
</tr>
<tr>
<td>NeRV</td>
<td>97.24M</td>
<td>7445s</td>
<td>36.08 dB</td>
</tr>
<tr>
<td>DINER</td>
<td>8.77K</td>
<td>1309s</td>
<td>50.74 dB</td>
</tr>
</tbody>
</table>

The loss function is built by comparing the measurements with the results from applying the Eqn. 5 to the network output. We compare our method with the current SOTAs, i.e., the diffractive neural field (DNF) [44] and the PhysenNet [40]. Due to the lack of the ground truth in real images, we only provide comparisons on the open-source synthetic data [44] here, please refer the supplementary material for more qualitative comparisons on real data.

Fig. 8 shows the PSNR of reconstructed measurements over training time on lensless imaging, the diffractive measurements, it provides the advantage of wide field of view observation and has become an attractive microscopic technique [43] for analyzing the properties of the specimen. We take the classic multi-height lensless imaging as an example, where N measurements $I_z$ are captured under different specimen-to-sensor distances $z \in \{z_1, ..., z_N\}$ for the specimen’s amplitude and phase imaging recovery. In multi-height lensless imaging, $I_z$ could be modeled as applying Fresnel propagation to the complex field $O(x, y)$ of the specimen, i.e.,

$$I_z = |PSF_z \cdot [P(x, y) \cdot O(x, y)]|^2,$$

where $P(x, y)$ is the illumination pattern, $PSF_z$ is the point spread function of Fresnel propagation over distance $z$ between the specimen and the sensor.

We model $O(x, y)$ using the proposed method with the SIREN backbone and the network size is $2 \times 64$. The loss function is built by comparing the measurements with the results from applying the Eqn. 5 to the network output. We compare our method with the current SOTAs, i.e., the diffractive neural field (DNF) [44] and the PhysenNet [40]. Due to the lack of the ground truth in real images, we only provide comparisons on the open-source synthetic data [44] here, please refer the supplementary material for more qualitative comparisons on real data.

Fig. 9 provides qualitative comparisons on lensless imaging.
The forward imaging process of sensor placed at location $\Delta$ illuminations. A popular technique for recovering the 3D refractive index measures the squared amplitude of the light scattered from the specimen at different angles multiple times and has become a common source of optical contrast for imaging specimens without staining or labelling, and plays an important role in many areas, e.g., the morphogenesis, cellular pathophysiology, biochemistry [27]. Intensity diffraction tomography measures the squared amplitude of the light scattered from the specimen at different angles multiple times and has become a popular technique for recovering the 3D refractive index.

Given the 3D refractive index $\mathbf{n} = (n_{re} + jn_{im})$ of a specimen, where $n_{re}$ and $n_{im}$ are the real and imaginary parts of the specimen’s refractive index, respectively. The forward imaging process of sensor placed at location $\rho$ could be modeled as

$$I_\rho = A_\rho \Delta \varepsilon,$$

where $A_\rho$ records the sample-intensity mapping with the illuminations. $\Delta \varepsilon = \Delta \varepsilon_{re} + j \Delta \varepsilon_{im}$ is the complex-valued permittivity contrast and could be obtained by solving

$$n_{re} = \frac{1}{2} \left( (n_0^2 + \Delta \varepsilon_{re}) + \sqrt{((n_0^2 + \Delta \varepsilon_{re})^2 - 4n_0^2 \Delta \varepsilon_{im}^2)} \right),$$

$$n_{im} = \frac{\Delta \varepsilon_{im}}{2 \cdot n_0},$$

where $n_0$ is the refractive index of the background medium.

We model the $(\Delta \varepsilon_{re}, \Delta \varepsilon_{im})$ using the DINER with network size $2 \times 64$. We compare our method with the SOTA, i.e., DeCAF [19] which uses the PE+MLP structure with size $8 \times 256$. All other parameters and loss functions in our method share the same setting with the DeCAF.

Fig. 10 compares our method with the DeCAF on the 3D Granulocyte Phantom data using the Fiji software [33]. Because the ground truth is not open-source until the submission, we take a screenshot of the ground truth in their paper as a reference. The MSE values labelled in Fig. 10 are computed by comparing the reconstructed measurements using the network output and the real measurements. Since the hash-table could map a high-frequency signal in a low-frequency way, our results provide more details on the surface of the Granulocyte. While the surface of the Granulocyte is over-smoothed in the results of the DeCAF since the PE+MLP could not accurately model the high-frequency components (Please refer to the supplemented video on different heights for better comparison).

5.5. Discussion

The aforementioned experiments are all focused on discrete signals. To query an unseen coordinate in a continuous signal, it is suggested to apply a post-interpolation operation to the network output instead of feeding interpolated hash-key to the network (see Fig. 11), such as the exploration of Plenoxels which interpolates the grid of density and harmonic coefficients [7] instead of feeding unseen position and direction coordinates to the network directly [22].

6. Conclusions

In this work, we have proposed the DINER which could greatly improve the accuracy of current INR backbones by introducing an additional hash-table. We have pointed out that the accuracy of INR for representing a signal is determined by the arrangement order of elements in it. The proposed DINER could map the input discrete signal into a low-frequency one, which is invariant if only the arrangement order changes while the histogram of attributes is not changed. For this reason, the accuracy of different INR backbones could be greatly improved. Extensive experiments have verified the high accuracy and efficiency of the proposed DINER for tasks of signal fitting and inverse problem optimization.

However, the current DINER could only process discrete signals. In the future, we will focus on continuous mapping methods instead of discrete hash-table-based mapping to extend the advantages for continuous signals such as the signed distance function [26].
References


[14] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and


6152