Regularize implicit neural representation by itself

Zhemin Li[1] Hongxia Wang[*1] Deyu Meng[2,3]
[1] National University of Defense Technology [2] Xi'an Jiaotong University
[3] Macau University of Science and Technology
lizhemin@nudt.edu.cn,wanghongxia@nudt.edu.cn,dymeng@mail.xjtu.edu.cn

Abstract

This paper proposes a regularizer called Implicit Neural Representation Regularizer (INRR) to improve the generalization ability of the Implicit Neural Representation (INR). The INR is a fully connected network that can represent signals with details not restricted by grid resolution. However, its generalization ability could be improved, especially with non-uniformly sampled data. The proposed INRR is based on learned Dirichlet Energy (DE) that measures similarities between rows/columns of the matrix. The smoothness of the Laplacian matrix is further integrated by parameterizing DE with a tiny INR. INRR improves the generalization of INR in signal representation by perfectly integrating the signal's self-similarity with the smoothness of the Laplacian matrix. Through well-designed numerical experiments, the paper also reveals a series of properties derived from INRR, including momentum methods like convergence trajectory and multi-scale similarity. Moreover, the proposed method could improve the performance of other signal representation methods.

1. Introduction

INR uses a fully connected network (FCN) $\phi_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}^o$ to approximate the explicit solution of an implicit function $F\left(\mathbf{x}, \phi_{\boldsymbol{\theta}}, \nabla_{\mathbf{x}}\phi_{\boldsymbol{\theta}}, \nabla^2_{\mathbf{x}}\phi_{\boldsymbol{\theta}}, \ldots\right) = 0$. For an example, we can represent a gray-scale image $\mathbf{X} \in \mathbb{R}^{m\times n}$ with an INR $\phi_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^2 \mapsto \mathbb{R}$ which satisfied $\phi_{\boldsymbol{\theta}}(\frac{i}{m}, \frac{j}{n}) = \mathbf{X}_{ij}, i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}$. Compared with traditional grid representation $\mathbf{X}$, INR's representation ability to details is not restricted by grid resolution m, n as INR can predict the pixel value at any location $(x, y) \in \mathbb{R}^2$ even not equals to $(\frac{i}{m}, \frac{j}{n})$.

Besides the representation ability of INR, generalization ability is critical for a neural network. We explore the empirical generalization ability via a 256×256 gray-scale non-uniformly sampled image inpainting task as Figure 2(a) shows. Although INR fits training data perfectly in Figure 2(b), its prediction outside training data is unreasonable. Theoretical analysis of INR illustrates that a hyper-parameter controls the smoothness degree of $\phi_{\boldsymbol{\theta}}(\mathbf{x})$. Moreover, the experiments show that the best hyper-parameter varies with the missing rate (the percentage of unsampled pixels) as Figure 3 shows. Adjusting this hyper-parameter cannot make the non-uniformly missing case perform best, as different locations might have different missing rates.

A carefully designed regularizer is proposed to improve the generalization ability of INR. It is based on Adaptive and Implicit Regularization (AIR) which is a learned Dirichlet Energy (DE) [12] that measures similarities or correlations between rows/columns of $\mathbf{X}$. The smoothness of the Laplacian matrix is further integrated by parameterizing DE with a tiny INR. The structure of the proposed implicit neural representation regularizer (INRR) is shown in Figure 1(b). Because a smooth Laplacian matrix represents non-local prior and large-scale local prior in vision data, INRR can improve the generalization of INR in image representation. Numerous numerical experiments show that INRR outperforms various classical regularizers, including total variation (TV), L_2 energy, and so on. As a regularizer both in a new form and with new meaning, INRR can be combined with other signal representation methods, such as deep matrix factorization (DMF) [1].

To summarize, the contributions of our work include the following:

- Neural Tangent Kernel (NTK) [1] theoretically analyzes the generalization ability of INR and why INR performs poorly with nonuniform sampling is given.
- A tiny INR parameterized regularizer named INRR is proposed based on DE, which perfectly integrates the image's self-similarity with the smoothness of the Laplacian matrix.
- A series of properties derived from INRR, including momentum methods, multi-scale similarity, and gener-

*This work was supported by the National Key Research ,Development Program (2020YFA0713504), the National Natural Science Foundation of China (61977065) and the Macao Science and Technology Development Fund (061/2020/A2).

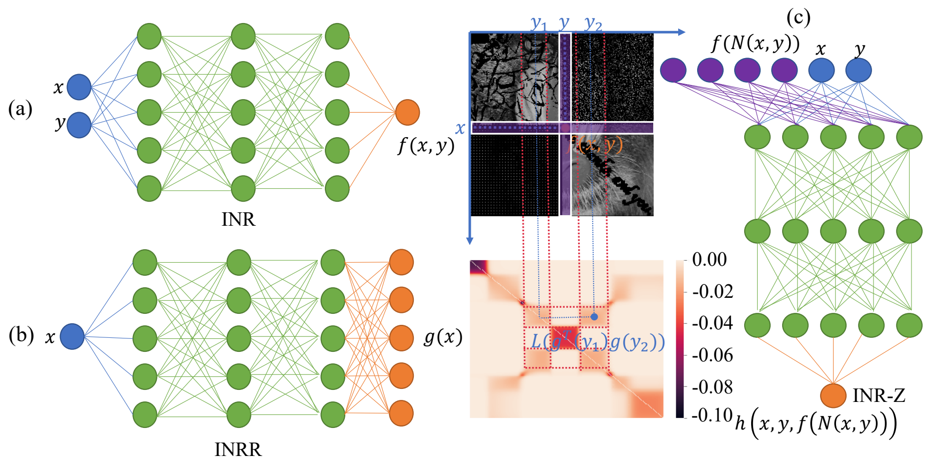
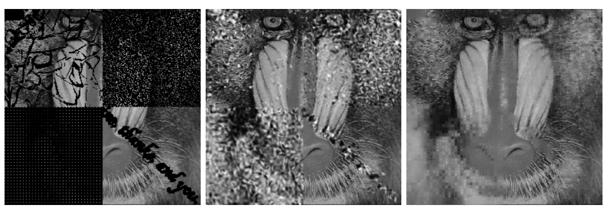


Figure 1. Overview of proposed improve scheme for INR. (a) INR is a fully connected neural network which maps from coordinate to pixel value. (b) INRR is a regularization term represented by an INR which can capture the self-similarity. (c) INR-Z improve the performance of INR by combining the neighbor pixels with coordinate together as the input of another INR.



(a) Sampling (b) INR (18.1 dB) (c) INRR (23.3 dB)

Figure 2. Image fitting results. All the methods are based on the SIREN to fit an 256×256 Baboon with the sampling data in (a). (b) trained with a vanilla SIREN while (c) trained with proposed INRR.

alization ability, are revealed by well-designed numerical experiments.

2. Related Work

Implicit neural representation. Recently, INR has shown outstanding potential in representing vision data, including font, images, and videos [22, 24]. It has been applied in novel view synthesis [10, 17, 20, 30], signal compression [4, 6, 21, 26, 33], and classification [5, 19].

In these latter years, a series of works have systematically studied and advanced the representation capabilities of INR. Tancik et al. discuss why an INR with ReLU activation function can not represent the high-frequency components well and introduce a Fourier feature encode that significantly improves the representation ability of INR [27]. Furthermore, Stizmann et al. replace ReLU with a sinuous activation function and propose a specific initialization scheme. The corresponding network is named sinusoidal representation network (SIREN) [24]. Then Fathony et al. propose filter neural networks with the Fourier and Gabor as basis activation [7]. Furthermore, Band-limited Coordinate Networks (BACON) introduces the ability of multiscale INR representation [14]. Apart from fitting the training set, the generalization ability of INR is more critical in many applications.

Regularization. Improving the generalization of NN with regularization techniques such as L_1-norm, L_2-norm, and the Dropout technique has a long history [25]. These regularizations take the images or other signals as input. Recently, there has been a class of NN that use a whole NN to represent a signal, such as Deep Image Prior (DIP), Deep Matrix Factorization (DMF), and INR [1, 24, 28]. In this case, the classical signal regularization technique can be applied to the signal represented NN [12, 13, 15, 18]. Significantly, the learnable regularizer is better than those not learned [12, 18]. To our knowledge, no effort has been made to regularize INR using a learnable regularizer based on the characteristics of INR's data representation.

3. Theoretical analysis of INR

As Figure 2(b) shows, INR's generalization ability is not as well as its representation ability. We analyze INR theoretically with a proxy model NTK to answer when and why INR generalizes badly.

Implicit neural representation. INR uses a FCN $\phi_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}^o$ to approximate the explicit representation of an implicit function $F\left(\mathbf{x}, \phi_{\boldsymbol{\theta}}, \nabla_{\mathbf{x}}\phi_{\boldsymbol{\theta}}, \nabla_{\mathbf{x}}^2\phi_{\boldsymbol{\theta}}, \ldots\right) = 0$, where the FCN has L hidden layers defined as follows,

$$\begin{aligned} \mathbf{z}^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{x}^{(\ell-1)} + \mathbf{b}^{(\ell)}, \quad \mathbf{x}^{(\ell)} = \sigma\left(\mathbf{z}^{(\ell)}\right), \\ \phi_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{z}^{(L+1)}, \quad \ell = 1, 2, \ldots, L+1, \end{aligned} \tag{1}$$

with $\sigma(\cdot)$ an element-wise activation function and $\mathbf{x}^{(0)} = \mathbf{x}$,

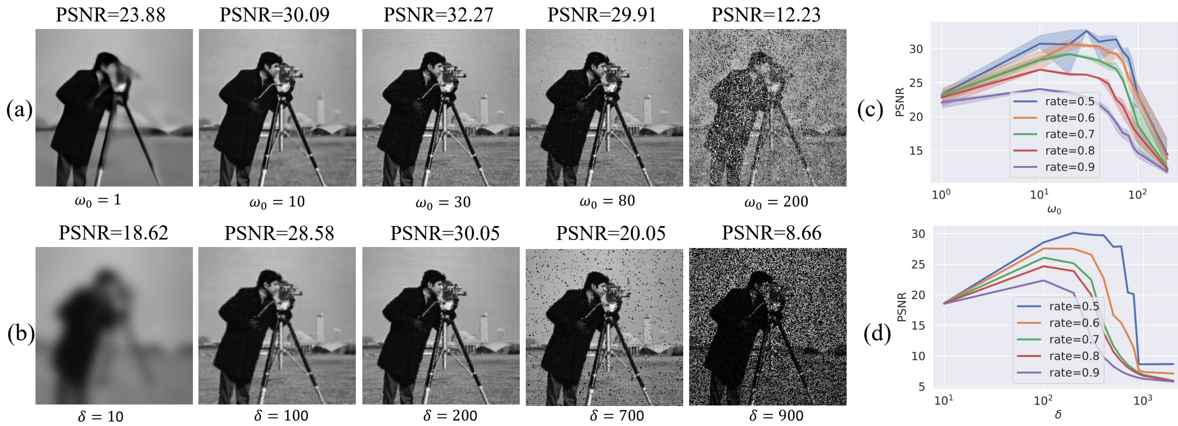


Figure 3. Fitting a 256×256 Cameraman which random missing 50% pixels with (a) SIREN and (b) NTK, respectively. ω_0 and δ are the hyper-parameters of models. (c,d) shows the PSNR value change with ω_0, δ at different random missing rate, respectively.

$\boldsymbol{\theta} = \left\{\mathbf{W}^{(\ell)}, \mathbf{b}^{(\ell)} \mid \ell = 1, 2, \ldots, L+1\right\} \sim \mathcal{D}$ at initialization, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$, and n_ℓ is the width of ℓ-th layer with $n_0 = d$, $n_{L+1} = o$.

For simplicity, we focus on a special case of INR with $F(\boldsymbol{\theta}, \mathbf{x}, z) = \|\phi_{\boldsymbol{\theta}}(\mathbf{x}) - z(\mathbf{x})\|_2^2 = 0$ and $\phi_{\boldsymbol{\theta}}(\mathbf{x}) : \mathbb{R}^2 \mapsto \mathbb{R}$, where $z(\mathbf{x}) : \mathbb{R}^2 \mapsto \mathbb{R}$ is the gray-scale image we want to represent. All the results in this paper can be easily extended to a higher dimension. The vanilla INR is formulated as

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \left\{ \mathcal{L}(\boldsymbol{\theta}, \mathcal{X}, \mathcal{Z}) = \sum_{(\mathbf{x}_i, z_i) \in \mathcal{X} \times \mathcal{Z}} F\left(\boldsymbol{\theta}, \mathbf{x}_i, z_i\right) \right\}, \tag{2}$$

where $\mathcal{X} \times \mathcal{Z} = \{(\mathbf{x}_i, z_i)\}_{i=1}^N$ is the training set, and $\mathbf{x}_i = (x_i, y_i)$ is the coordinate. The training set is sampled from the grid of matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. For example, we can use $(\frac{i}{m}, \frac{j}{n})$ as input and $\mathbf{X}_{ij}$ as the corresponding output of INR. After training, $z(\mathbf{x})$ is predicted by $\phi_{\boldsymbol{\theta}}(\mathbf{x})$ at any location $\mathbf{x} = (x, y)$ even when $\mathbf{x} \notin \mathcal{G} = \left\{(\frac{i}{m}, \frac{j}{n}) \mid i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\}\right\}$.

Kernel regression approximate neural networks. Jacot et al. show that with infinity width of the layers in $\phi_{\boldsymbol{\theta}}$ and small learning rate, the function $\phi_{\boldsymbol{\theta}}$ converges to the kernel regression

$$\phi_{\text{NTK}}(\mathbf{x}) = \sum_{i=1}^N \left(\mathbf{K}^{-1}\mathbf{z}\right)_i k_{\text{NTK}}\left(\mathbf{x}_i, \mathbf{x}\right),$$

where $\mathbf{K}$ is an $N \times N$ kernel matrix dubbed neural tangent kernel (NTK) [8, 9, 32] with entries defined as

$$\mathbf{K}_{ij} = k_{\text{NTK}}(\mathbf{x}_i, \mathbf{x}_j) = \mathbb{E}_{\boldsymbol{\theta} \sim \mathcal{D}} \left\langle \frac{\partial \phi_{\boldsymbol{\theta}}(\mathbf{x}_i)}{\partial \boldsymbol{\theta}}, \frac{\partial \phi_{\boldsymbol{\theta}}(\mathbf{x}_j)}{\partial \boldsymbol{\theta}} \right\rangle.$$

In this paper, we consider INR $\phi' : \mathbb{R}^{2D} \mapsto \mathbb{R}$ with a feature map $\gamma(\mathbf{x}) = \frac{1}{\sqrt{D}}[\cos \mathbf{B}\mathbf{x}^\top, \sin \mathbf{B}\mathbf{x}^\top]^\top : \mathbb{R}^d \mapsto \mathbb{R}^{2D}$ as its input, where $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{B} \in \mathbb{R}^{D \times d}$, and $\mathbf{B}_{ij} \sim \mathcal{N}(0, \delta)$. Then $\phi'_{\text{NTK}}(\gamma(\mathbf{x}))$ is shift-invariant thus more suitable for image representation.

Now we analyze how INR predicts the data outside of the training set. Theorem 1 illustrates that the smoothness of represented signal is controlled by the hyper-parameter δ globally. Especially when δ tends to infinity, the prediction of $\phi'_{\text{NTK}}(\cdot)$ outside the training set all tends to the same weighted average of the training set according to Corollary 1.

Theorem 1 *Given a FCN $\phi'_{\boldsymbol{\theta}}(\cdot) : \mathbb{R}^{2D} \mapsto \mathbb{R}$ with $\mathbf{x} \in \mathbb{R}^d$, $\gamma(\mathbf{x}) \in \mathbb{R}^{2D}$, and the feature map $\gamma(\mathbf{x}) = \frac{1}{\sqrt{D}}[\cos \mathbf{B}\mathbf{x}^\top, \sin \mathbf{B}\mathbf{x}^\top]^\top$ with $\mathbf{B} \in \mathbb{R}^{D \times d}$ and $\mathbf{B}_{ij} \sim \mathcal{N}(0, \delta)$. Denote the corresponding composed NTK as $k_D(\mathbf{x}_i, \mathbf{x}_j) = h_{NTK}\left(\frac{1}{D}\mathbf{1}_D^\top \cos\left(\mathbf{B}(\mathbf{x}_i - \mathbf{x}_j)\right)\right)$, then we have*

$$\lim_{D \to \infty} k_D(\mathbf{x}_j, \mathbf{x}_j) = h_{NTK}(e^{-\delta^2 \|\mathbf{x}_i - \mathbf{x}_j\|^2}).$$

Corollary 1 *Assume the h_{NTK} in Theorem 1 satisfies $h_{NTK}(1) \neq h_{NTK}(0)$ and $h_{NTK}(1) \neq 0$, then*

$$\lim_{\delta \to \infty} \phi'_{NTK}(\gamma(\mathbf{x})) = \begin{cases} z_l, & \mathbf{x} = \mathbf{x}_l \in \{\mathbf{x}_i\}_{i=1}^N, \\ \frac{h(0)\mathbf{1}_N^\top \mathbf{z}}{(N-1)h(0)+h(1)}, & \mathbf{x} \notin \{\mathbf{x}_i\}_{i=1}^N. \end{cases}$$

INR needs to be regularized. We validate Corollary 1 by exploring the performance of $\phi'_{\text{NTK}}(\gamma(\mathbf{x}))$ in image inpainting task with different missing rates. As Figure 3(b) shows, when $\delta = 900$, $\phi'_{\text{NTK}}(\gamma(\mathbf{x}))$ at the location of outside of sampled data has the same value. Furthermore, Figure 3(a) shows that the latest SIREN [24], which represents signals without a feature map of input, is also controlled by the hyper-parameter ω_0 in the first layer as $\sin\left(\omega_0 \mathbf{W}\mathbf{x} + \mathbf{b}\right)$.

Based on the numerical result, the optimal ω_0 or δ is required so that INR generalizes the best. However, finding an optimal ω_0 or δ with non-uniformly sampled training data is impossible. Figure 3(c,d) illustrates that the optimal ω_0 or δ varies considerably according to the missing rate. It decreases with the increase of missing rate, which is consistent with the theoretical results that the sparser sampling needs a smoother fitting. As to the case with nonuniform missing, note that different locations might have different missing rate; it is tough to make INR performs well by choosing an optimal hyper-parameter.

Furthermore, the results above all based on the loss function $\mathcal{L} = \sum_{(\mathbf{x}_i, z_i) \in \mathcal{X} \times \mathcal{Z}} \|\phi_{\boldsymbol{\theta}}(\mathbf{x}_i) - z_i\|_2^2$ which is a fidelity term measured on the training data. Enforcing additional constraints on the predicted data is profitable to improve the generalization ability of INR. In the next section, we add constraints by a newly proposed regularizer named INRR.

4. Methods to regularize INR

This section presents a regularized model $\mathcal{L}(\boldsymbol{\theta}, \mathcal{X}, \mathcal{Z}) + \lambda \mathcal{R}(\boldsymbol{\theta}, \mathcal{X}, \mathcal{Z})$, where λ is a parameter that balances the loss of training data and the regularizer $\mathcal{R}$.

Now consider the priors of images on a larger scale. Since the vanilla INR's loss function is pixel-by-pixel, it ignores the structural features of images. Specifically, these features include the relationship between rows, columns, or blocks. Low rank is a well-known prior that describes the correlation between rows and columns. However, a low-rank matrix cannot express the details of a signal well because these details are located in the subspaces corresponding to the small singular value of the image.

So we turn to self-similarity, which is quite common in large and fine scales of an image. As a simple example, smooth $\mathbf{X}$ implies local similarity between adjacent rows and columns of $\mathbf{X}$. Furthermore, the non-local self-similarity of an image, which refers to the similarity between non-adjacent rows, columns, or blocks, is also very universal and valuable. In this paper, we choose Dirichlet Energy (DE) to describe images' local and non-local self-similarity. Our method is not restricted to DE.

4.1. Dirichlet Energy

Given a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, DE is formulated as follows

$$\mathcal{R}_{\text{DE}} = tr\left(\mathbf{X}^\top \mathbf{L} \mathbf{X}\right) = \sum_{1 \le i,j \le m} \mathbf{A}_{ij} \left\|\mathbf{X}_{i,:} - \mathbf{X}_{j,:}\right\|^2,$$

where $\mathbf{A} \in \mathbb{R}^{m \times m}$ is a weighted adjacency matrix along rows of $\mathbf{X}$, and $\mathbf{L} = \mathbf{D} - \mathbf{A}$ with $\mathbf{D}_{ii} = \sum_{j=1}^m \mathbf{A}_{ij}$ and $\mathbf{D}_{ij} = 0$ if $i \neq j$. As $\mathbf{A}_{ij}$ measures the similarity of rows $\mathbf{X}_{i,:}$ and $\mathbf{X}_{j,:}$, DE is a non-local self-similarity measure of $\mathbf{X}$.

However, there are two main issues in using DE: (a) $\mathbf{L}$ or $\mathbf{A}$ is unknown under the incomplete sampling of $\mathbf{X}$; (b) DE only encodes the similarity between two rows, other large-scale similarities such as block similarity cannot be captured. To solve these problems, we parameterize $\mathbf{L}$ with another tiny INR and learn it during training $\phi_{\boldsymbol{\theta}}(\mathbf{x})$.

4.2. INRR

Learning $\mathbf{L}$ during training is naive thinking when $\mathbf{L}$ is unknown. Nevertheless, we need to sufficiently extract the properties of $\mathbf{L}$ to make it meaningful and practical. There are two mathematical properties that $\mathbf{L}$ needs to satisfy: (a) positive semi-definite, (b) the sum of each row equals zero. Specially, we find the $\mathbf{L}$ of natural images has some extra priors. The natural images are usually piecewise smooth, so $\mathbf{L}$, which measures the similarity of the rows of $\mathbf{X}$, should also be nearly smooth.

Therefore, we propose an implicit neural representation regularization (INRR) which is expressed as follows:

$$\begin{cases} \mathcal{R}(\boldsymbol{\theta}) = tr\left(\left[\mathcal{T}(\mathbf{X})\right]^\top \mathbf{L}(\boldsymbol{\theta}) \mathcal{T}(\mathbf{X})\right) \\ \mathbf{L}(\boldsymbol{\theta}) = \mathbf{A}(\boldsymbol{\theta}) \cdot \mathbf{1}_{m' \times m'} \odot \mathbf{I}_{m'} - \mathbf{A}(\boldsymbol{\theta}) \\ \mathbf{A}(\boldsymbol{\theta}) = \dfrac{\exp\left(g^\top(\boldsymbol{\theta}; \mathbf{u}) g(\boldsymbol{\theta}; \mathbf{u})\right)}{\mathbf{1}_{m'}^\top \exp\left(g^\top(\boldsymbol{\theta}; \mathbf{u}) g(\boldsymbol{\theta}; \mathbf{u})\right) \mathbf{1}_{m'}} \end{cases},$$

where $\mathcal{T}(\cdot) : \mathbb{R}^{m \times n} \mapsto \mathbb{R}^{m' \times n'}$ aims to capture self-similarity in $\mathbf{X}$, $\mathbf{L}(\boldsymbol{\theta})$ measures the similarity between rows of $\mathcal{T}(\mathbf{X})$. $g(\boldsymbol{\theta}; \cdot) : \mathbb{R} \mapsto \mathbb{R}^r$ is a tiny INR, $g(\boldsymbol{\theta}; \mathbf{u}) \in \mathbb{R}^{r \times m'}$, $g(\boldsymbol{\theta}; \mathbf{u})_i = g(\boldsymbol{\theta}; \mathbf{u}_i), i = 1, 2, \ldots, m'$. And $\mathbf{u}_i$ is coordinate of sampled matrix $\mathcal{T}(\mathbf{X})$ with $\mathcal{T}(\mathbf{X})_{ij} = \phi_{\boldsymbol{\theta}}(\mathbf{u}_i, \mathbf{v}_j)$ and $\mathbf{u} = \left[\frac{1}{m'}, \frac{2}{m'}, \ldots, \frac{m'}{m'}\right]^\top$, $\mathbf{v} = \left[\frac{1}{n'}, \frac{2}{n'}, \ldots, \frac{n'}{n'}\right]^\top$. It is not difficult to verify that the parameterized Laplacian matrix keeps properties (a) and (b). Furthermore, $g(\boldsymbol{\theta}; \cdot)$ introduces the smoothness of $\mathbf{L}$ implicitly, and r restricts the rank of $\mathbf{L}$.

Take the relations between columns into account simultaneously. The whole regularized model is formulated as

$$\text{minimize}_{\boldsymbol{\theta}, \boldsymbol{\theta}_r, \boldsymbol{\theta}_c} \left\{\mathcal{L}(\boldsymbol{\theta}, \mathcal{X}, \mathcal{Z}) + \lambda_r \mathcal{R}(\boldsymbol{\theta}_r) + \lambda_c \mathcal{R}(\boldsymbol{\theta}_c)\right\},$$

where $\mathcal{R}(\boldsymbol{\theta}_r)$ and $\mathcal{R}(\boldsymbol{\theta}_c)$ are row and column regularizers respectively. $\mathcal{T}(\mathbf{X}) = \mathbf{X}$ in $\mathcal{R}(\boldsymbol{\theta}_r)$, and $\mathcal{T}(\mathbf{X}) = \mathbf{X}^\top$ in $\mathcal{R}(\boldsymbol{\theta}_c)$. λ_r, λ_c are used to balance the fidelity and regularization terms.

As the self-similarity which is represented by $g(\boldsymbol{\theta}_r; \cdot)$ or $g(\boldsymbol{\theta}_c; \cdot)$ are much simpler than the image, so the parameter number of $g(\boldsymbol{\theta}_r; \cdot)$ or $g(\boldsymbol{\theta}_c; \cdot)$ are much lesser than the one of $\phi_{\boldsymbol{\theta}}(\cdot)$, which called tiny INR.

5. Experiments

5.1. Experimental setting

Data types and missing patterns. We consider five gray-scale benchmark images of size $m \times n = 256 \times 256$, including Baboon, Man, Barbara, Boats, and Cameraman. Moreover, we study matrix completion with three different missing patterns: random missing, patch missing, and textural

Table 1. PSNR (dB) of recovered images by INR based models with different missing patters include random missing, patch missing, and textural missing. Four images are tested including Baboon, Barbara, Man and Boats.

		INR	INR-Z	TV	L_2	AIR	**INRR**		INR	INR-Z	TV	L_2	AIR	**INRR**
Random		21.0	21.5	23.8	23.1	24.2	**24.7**		28.6	28.8	30.2	30.5	30.3	**30.7**
Patch	Baboon	25.3	27.3	24.5	26.3	31.6	**33.8**	Barbara	26.1	27.1	28.8	28.5	28.9	**29.5**
Textural		20.8	23.5	21.4	23.9	27.2	**28.3**		26.5	27.9	26.5	27.9	29.0	**29.4**
Random		23.5	22.8	25.6	25.6	25.7	**25.9**		28.1	27.6	29.5	29.7	29.4	**29.9**
Patch	Man	25.6	26.5	22.7	26.9	31.3	**32.1**	Boats	27.6	28.5	27.8	29.5	33.4	**34.4**
Textural		22.6	22.6	24.2	24.9	25.3	**26.4**		24.1	27.0	25.2	28.1	27.9	**28.9**

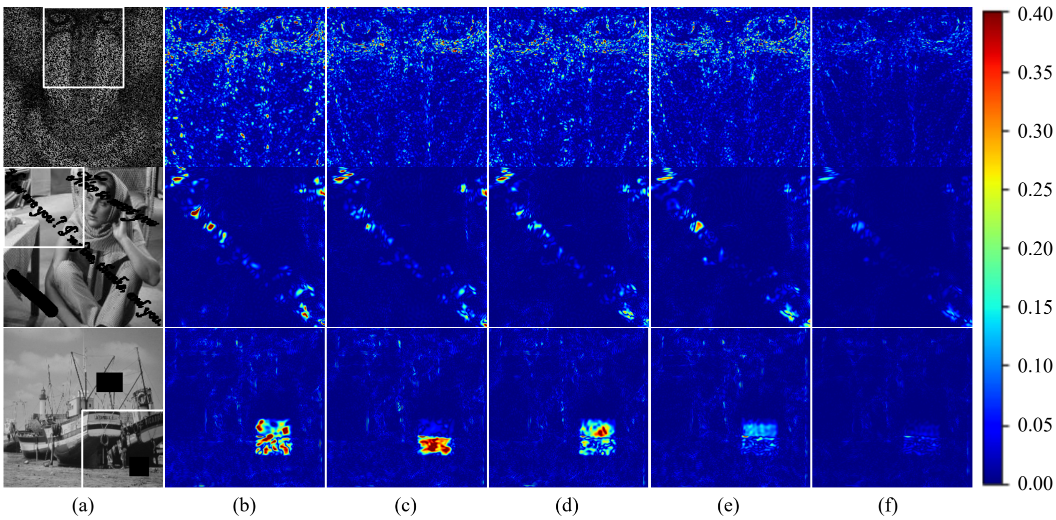


Figure 4. Residual of image inpainting, i.e., $|\mathbf{X} - \mathbf{X}^*|$ with three types of missing data by different regularized INR including (b) INR without regularization, (c) with TV, (d) L_2, (e) AIR, and (f) INRR. The hyper-parameters of benchmark models and algorithms are adopted from the original paper.

missing, which is shown as different parts in Figure 1(a). The default missing rate is 50%.

Network settings. In this section, the INR defaults to SIREN when not otherwise specified [24]. The INR network is organized in five hidden layers SIREN whose widths are all the same as 256. As to INRR, five hidden layers SIREN is chosen with the same width 32, and the output dimension $r = \max(m, n)$. We use Adam with default settings in [11] to train all the networks.

Peered methods The peered methods include

1. TV: $\mathcal{R}_{\text{TV}} = \sum_{\mathbf{x}_i=(x_i,y_i)\in\mathcal{G}} \|\nabla_{\mathbf{x}}\phi_{\boldsymbol{\theta}}(\mathbf{x}_i)\|_1$, $\mathcal{G} = \left\{\frac{1}{m}, \ldots, \frac{m}{m}\right\} \times \left\{\frac{1}{n}, \ldots, \frac{n}{n}\right\}$, which is the discrete version on $\mathcal{G}$.
2. L_2: $\mathcal{R}_{L_2} = \sum_{\ell=1}^{L+1} \left\|\mathbf{W}^{(\ell)}\right\|_2$ which is a common regularizer which is used to regularize NN.
3. INR-Z: Combining the neighbor of the input with coordinate $[x, y, f(N(x, y))] \in \mathbb{R}^{(N_0+2)\times 1}$ as the input of a new INR $h(\cdot) : \mathbb{R}^{N_0+2} \mapsto \mathbb{R}$ as Figure 1(c) shows.
4. AIR: Adaptive and implicit regularization [12]
5. INRR: Implicit neural representation regularization proposed in this paper.

5.2. Image representation with various missing patterns

We apply INRR for matrix completion (or image inpainting) on three types of missing patterns. A few related models are also used for comparison.

Adaptive to training data. We compare vanilla INRR with several improved models in the following experiments, including TV, L_2, AIR, INRR, and INR-Z. Table 1 lists the PSNRs of recovered images using the aforementioned improved models for different data with different missing patterns. The results show that the non-local regularization methods, including AIR and INRR, significantly outperform the vanilla INR. Furthermore, INRR is much better than AIR since INRR integrates the smoothness of Laplacian matrix into the DE regularizer. The residual of recovered images $|\mathbf{X} - \mathbf{X}^*|$ corresponding to Table 1 are shown in Figure 4. Unlike other INR-regularized methods that perform well for random missing cases but poorly for other missing

Table 2. PSNR of recovered images by five data representation models without regularization and with INRR under mixture missing. The hyper-parameters of benchmark models and algorithms are adopted from the original paper. The value in parentheses represents the increment of the PSNR value after using the INRR.

	DMF	**DMF+**	ReLU	**ReLU+**	Siren	**Siren+**	Garbor	**Garbor+**	Fourier	**Fourier+**
Baboon	8.2	**22.1 (13.9)**	20.9	**21.4 (0.6)**	17.6	**22.7 (5.1)**	17.5	**22.8 (5.4)**	14.3	**22.8 (8.5)**
Man	8.4	**21.7 (13.3)**	22.2	**22.5 (0.3)**	18.9	**22.8 (3.8)**	17.9	**22.0 (4.1)**	15.3	**22.1 (6.9)**
Barbara	8.8	**25.0 (16.3)**	25.2	**26.4 (1.2)**	23.3	**25.6 (2.3)**	19.4	**25.0 (5.5)**	17.4	**25.0 (7.7)**
Boats	10.9	**24.4 (13.4)**	24.0	**25.2 (1.2)**	22.2	**25.7 (3.4)**	17.6	**24.6 (7.0)**	14.9	**24.6 (9.7)**
Cameraman	6.9	**24.5 (17.6)**	25.7	**25.8 (0.1)**	23.9	**25.7 (1.8)**	18.2	**24.7 (6.5)**	15.2	**24.7 (9.4)**

patterns, INRR consistently gives visually appealing results. To conclude, INRR achieves decent results qualitatively and quantitatively independent of sampling mode of training data.

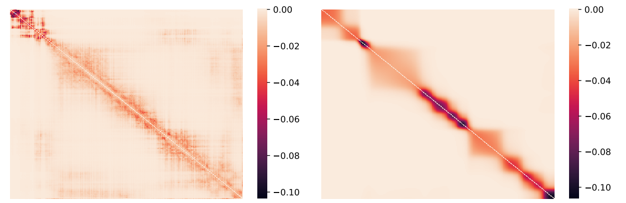
Adaptive to data representation. To distinguish the effect of INRR regularizer from the vanilla INR model, Table 2 lists the PSNRs of recovered images by several data representations which INRR regularizes. The data representation includes deep matrix factorization (DMF) [1], FCN with ReLU activation function, SIREN [24], the filter neural network with Gabor and Fourier filter, respectively [7]. The INRR regularized models are denoted by '.+' in Table 2. The mixture missing pattern is shown in Figure 2(a). The results shown in Table 2 and Figure 2(b)(c) both illustrate that INRR significantly improves the performance of recently proposed data representation methods without regularization. Overall, INRR is a general regularizer not limited to being combined with a particular data representation model.

6. Why INRR performs better

Now we have shown that INRR achieves excellent performance in image representation (image inpainting as an example) under different missing patterns. In this section, the reasons why INRR performs better than other peered methods are analyzed carefully. Firstly, the smoothness of $\mathbf{L}$ learned by INRR is demonstrated by experiments. Then a heuristic connection between INRR, implicit bias, and the momentum method is built.

6.1. Tiny INR smooths Laplacian matrix implicitly

Parameterizing DE with a tiny INR is the key of INRR. In this section, we focus on illustrating the benefit of this parameterization. A 256×256 Baboon is down-sampled to 128×128, and then AIR and INRR are used to regularize INR to recover the original image based on the sampled data. Figure 5 shows the Laplacian matrix $\mathbf{L}$ learned by AIR and INRR, respectively. The $\mathbf{L}$ learned by AIR (Figure 5(a)) is discontinuous with high probability at those locations that are not sampled, while the $\mathbf{L}$ learned by INRR (Figure 5(b)) is much more continuous. The continuous $\mathbf{L}$ introduced by the tiny INR is more consistent with practice.



(a) AIR (b) INRR

Figure 5. $\mathbf{L}$ at $t = 2000$ learned by (a) AIR and (b) INRR, respectively. All the methods are based on the SIREN to super-resolution Baboon, which is down-sampled from 256×256 to 128×128.

6.2. INRR behaves like a momentum

We connect INRR with the momentum method in this subsection. As Figure 8 shows, INRR tends to vanish during training. Then INR with INRR converges to the vanilla INR model. First, compare INRs with and without INRR by the optimization trajectory. In Figure 6, we plot the MSE's trajectory during training. At the beginning of training, the observed and unobserved MSEs of the five models drop similarly. However, these five models perform dramatically differently near the convergence. When the observed MSE becomes smaller, the model learns details in observed elements. The unobserved MSE increased during the observed MSE decrease in the vanilla INR, INR+TV, and INR+L_2 cases; we name this phenomenon over-fitting. INRR and AIR keep the decaying trend for both observed and unobserved MSEs. Significantly, the proposed INRR keeps the decaying trend better than AIR due to the extra smoothness introduced by a tiny INR.

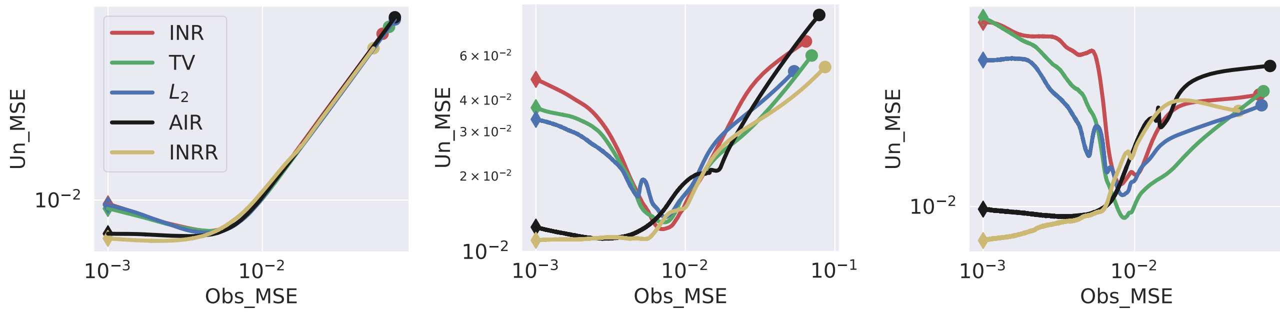
Looking back into the training process of INRR, the update of $\mathbf{X}(t+1)$ involves both $\mathbf{X}(t)$ and $\mathbf{L}(t)$, and the update of $\mathbf{L}(t)$ depends on $\mathbf{X}(t-1)$. To understand the training dynamics, we consider the following simplified model:

$$\text{minimize}_{\mathbf{X},\mathbf{L}} \left\{\mathcal{L} + \lambda tr(\mathbf{X}^\top \mathbf{L}\mathbf{X})\right\},$$

and we have

$$\nabla_{\mathbf{X}(t)}\mathbf{L} = \nabla_{\mathbf{X}(t)}\mathcal{L} + 2\lambda \mathbf{L}(t)\mathbf{X}(t),$$

where $\mathcal{L}$ is the fidelity term, $\mathbf{L}(t)$ is the function of $\{\mathbf{X}(t_0) \mid t_0 < t\}$ as $\mathbf{L}(t)$ is updated based on $\mathbf{X}(t-1)$.



(a) Random (b) Textural (c) Patch

Figure 6. Trajectories of training INR model without regularization, with TV, L_2, AIR and INRR regularization for inpainting the Baboon image with three types of missing types of missing pixels: (a) randomly missing 50%, (b) textural missing, and (c) patch missing. The dot point indicates $\mathbf{X}(0)$ and the diamond shape of different color indicate the MSE on training data achieve 10^{-3}.

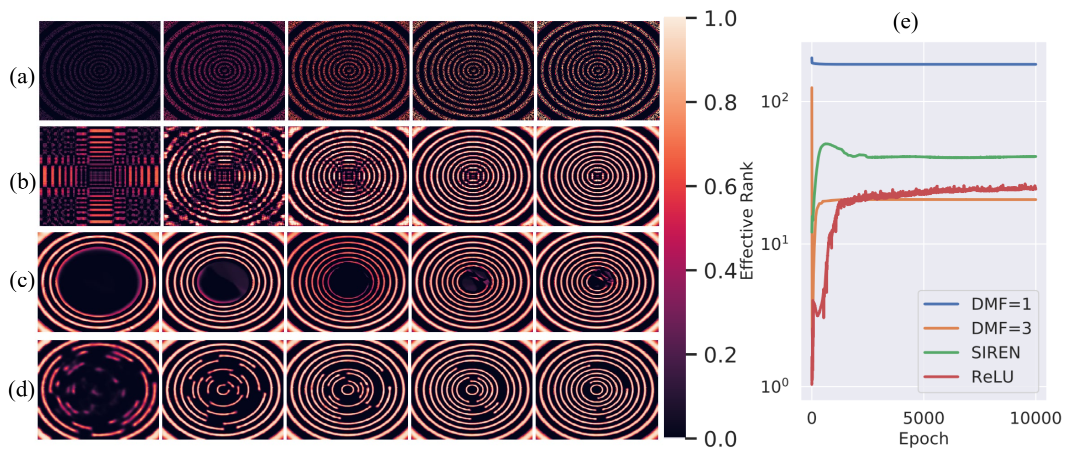


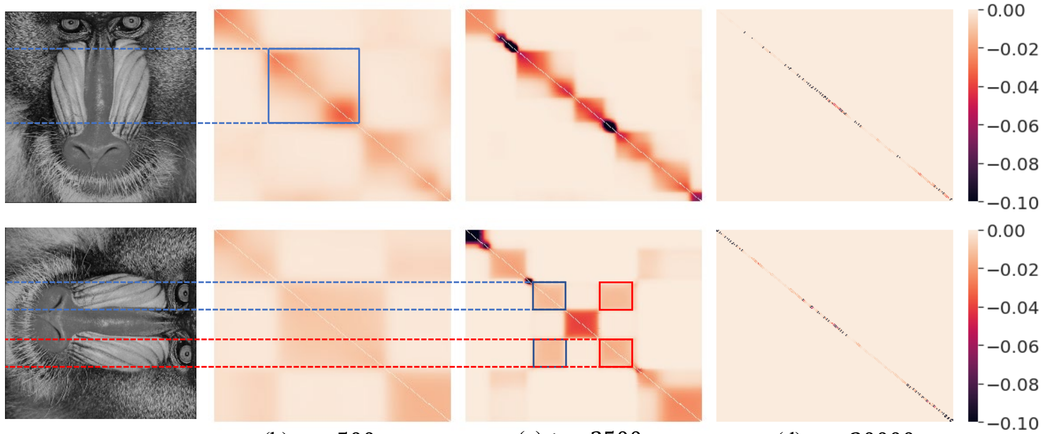
Figure 7. Fitting a 256×256 synthetic data which randomly missing 50% pixels at different training step with (a) single matrix (DMF with only one factor), (b) DMF with three factors, (c) ReLU neural network, and (d) SIREN. The effective rank of the fitted matrix is shown in (e).

Therefore, every iteration step of INRR leverages all the previously learned information $\{\mathbf{X}(t_0) \mid t_0 < t\}$. Note that the update of both vanilla INR, INR+TV, and INR+L_2 only depend on $\mathbf{X}(t)$. From this viewpoint, INRR shares a similar spirit as the momentum method, which leverages history to improve performance.

6.3. INRR connects implicit bias with multi-scale self-similarity

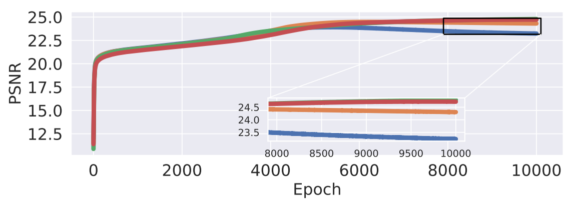
Implicit bias of NN. We then demonstrate other properties of INRR by connecting implicit bias with multi-scale self-similarity. The implicit bias of NN is used to explain the generalization ability of NN in recent years [1–3, 16, 29, 31, 34]. As Figure 7 shows, we fit synthetic data with DMF with one factor, DMF with three factors, ReLU FCN, and SIREN, respectively. The synthetic data is sampled from function $s(x, y) = \sin\left(25\pi \sin\left(\frac{\pi}{3} \cdot \sqrt{x^2 + y^2}\right)\right)$, where $\{(x_i, y_j)|i, j\}$ is a uniform 256×256 grid on $[-1, 1] \times [-1, 1]$, where the local frequency of the synthetic data increases from boundary to center. All the networks except DMF with one factor evolve from a low complexity pattern to a high complexity one. ReLU FCN and SIREN first fit the low-frequency components and then gradually fit the high-frequency components [2, 3, 16, 31]. More specifically, the effective rank of DMF with three factors, SIREN, and ReLU FCN, increases gradually as the line plot, where the effective rank can measure the effective dimension of the matrix with more accuracy than discrete rank [1, 23].

Multi-scale similarity captured by INRR. Then we turn to explain the multi-scale similarity seized by INRR. Due to the implicit bias of fidelity term, INRR can capture different scales of data similarity. The heatmaps of Laplacian matrices $\mathbf{L}_r(t)$ and $\mathbf{L}_c(t)$ for Baboon are shown in Figure 8. A few large blocks appear in $\mathbf{L}_r(500)$ and $\mathbf{L}_c(500)$ Figure 8(b), which reflect the similarity in a large scale. Then the size of blocks becomes smaller while the number of blocks increases at $t = 2500$ in Figure 8(c), which reflects

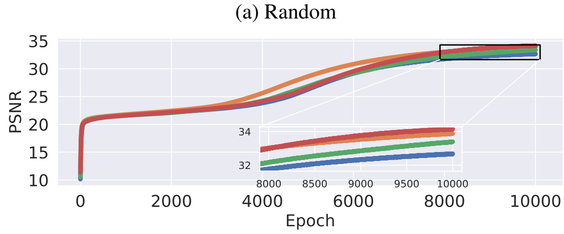


(a) Baboon (b) t = 500 (c) t = 2500 (d) t = 20000

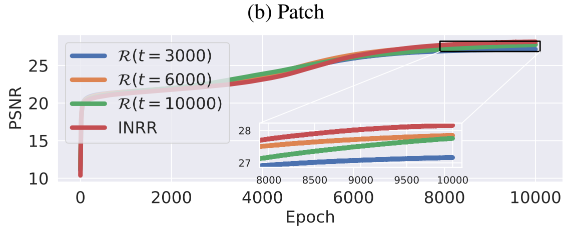
Figure 8. Learned $\mathbf{L}_r(t)$ and $\mathbf{L}_c(t)$ during training. (a): first and second rows depict the Baboon image and its rotation. (b)-(d): first/second row shows the heatmap of $\mathbf{L}_r/\mathbf{L}_c$ at different t. A darker color indicates a stronger similarity captured by the adaptive regularizer. The (i, j)-th element in the heatmap of $\mathbf{L}_r(t)$ has a darker color than the i, j'-th element indicates that the i-th row is more related to the j-th row compared with the j'-th row.



(a) Random



(b) Patch



(c) Textural

Figure 9. Contrasting the adaptive regularizer with the fixed regularizer for Baboon image recovery. We consider inpainting the Baboon image with three types of missing pixels: (a) randomly missing 50% pixels, (b) patch missing, and (c) textural missing. The red lines plot the PSNR during the training of the vanilla INRR. The remaining three lines in each figure indicate replacing $\mathbf{L}_r$ and $\mathbf{L}_c$ with $\mathbf{L}_r(t)$ and $\mathbf{L}_c(t)$ at $t = 3000, 6000$ and 10000, respectively.

the similarity on a smaller scale. The values in these blocks reflect the substantial similarity of the corresponding highlighted patches of the original Baboon, which echoes our intuition. Moreover, as the training goes further, both $\mathbf{L}_r(t)$ and $\mathbf{L}_c(t)$ focus on reflecting the similarity of the neighbor at $t = 20000$ in Figure 8(d), which is similar to the TV.

The importance of learned INRR. The results confirm that INRR captures the similarity from large to small. Next, we experimentally illustrate that the learned $\mathbf{L}_c$ and $\mathbf{L}_c$ by INRR are crucial for image representation. Fix $\mathbf{L}_r$ and $\mathbf{L}_c$ at a specific training step for INRR, and then compare INRR with the overall adaptive $\mathbf{L}_r$ and $\mathbf{L}_c$.

We contrast the vanilla INRR and INRR with fixed Laplacian matrices (let $t = 3000, 6000$ and 10000 respectively) for Baboon image inpainting. Figure 9 shows how the PSNR changes during training. INRR, which continuously updates the regularization during training, performs best for all missing patterns. Fixing Laplacian matrices helps reduce the computation costs during training. However, as the optimal t^* is varied with missing patterns, the learned Laplacian matrices are more applicable.

7. Conclusion

This paper proposes a novel regularizer named INRR, which significantly improves INR's representation performance, especially when the training data is sampled arbitrarily. INRR parameterizes the Laplacian matrix in DE by a tiny INR and then adaptively learns the non-local similarities hidden in image data. INRR is a generic framework for integrating multiple prior into a single regularizer, decreasing the redundancy of the regularizer. The connection among INRR, momentum term, implicit bias, and multi-scale self-similarity deserve further theoretical analysis.

References

[1] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *NeurIPS*, 2019. 1, 2, 6, 7

[2] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 2021. 7

[3] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In *IJCAI*, 2021. 7

[4] Emilien Dupont, Adam Goli'nski, Milad Alizadeh, Yee Whye Teh, and A. Doucet. Coin: Compression with implicit neural representations. *ArXiv*, abs/2103.03123, 2021. 2

[5] Emilien Dupont, Hyunjik Kim, S. M. Ali Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you should treat it like one. *ICML*, 2022. 2

[6] Emilien Dupont, Hrushikesh Loya, Milad Alizadeh, Adam Goli'nski, Yee Whye Teh, and A. Doucet. Coin++: Data agnostic neural compression. *ArXiv*, abs/2201.12904, 2022. 2

[7] Rizal Fathony, Anit Kumar Sahu, Devin Willmott, and J. Zico Kolter. Multiplicative filter networks. *ICLR*, 2021. 2, 6

[8] Eugene Golikov, Eduard Pokonechnyy, and Vladimir Korviakov. Neural tangent kernel: A survey. *ArXiv*, abs/2208.13614, 2022. 3

[9] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *NeurIPS*, 2018. 3

[10] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. *ICCV*, 2021. 2

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 5

[12] Zhemin Li, Tao Sun, Hongxia Wang, and Bao Wang. Adaptive and implicit regularization for matrix completion. *SIAM Journal on Imaging Sciences*, 15(4):2000–2022, 2022. 1, 2, 5

[13] Zhemin Li, Zhi-Qin John Xu, Tao Luo, and Hongxia Wang. A regularized deep matrix factorized model of matrix completion for image restoration. *IET Image Process.*, 16:3212–3224, 2022. 2

[14] David B. Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. *CVPR*, 2022. 2

[15] Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S. Kamilov. Image restoration using total variation regularized deep image prior. *ICASSP*, pages 7715–7719, 2019. 2

[16] Tao Luo, Zheng Ma, Zhi-Qin John Xu, and Yaoyu Zhang. Theory of the frequency principle for general deep neural networks. *CSIAM Transactions on Applied Mathematics*, 2021. 7

[17] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. *CVPR*, 2020. 2

[18] Gary Mataev, Peyman Milanfar, and Michael Elad. Deepred: Deep image prior powered by red. *ICCV*, 2019. 2

[19] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. *arXiv: Computer Vision and Pattern Recognition*, 2021. 2

[20] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *ECCV*, 2020. 2

[21] Francesca Pistilli, Diego Valsesia, Giulia Fracastoro, Enrico Magli Politecnico, and Di Torino. Signal compression via neural implicit representations. *ICASSP*, 2022. 2

[22] Pradyumna Reddy, Zhifei Zhang, Matthew Fisher, Hailin Jin, Zhaowen Wang, and Niloy J. Mitra. A multi-implicit neural representation for fonts. *ICML*, 2021. 2

[23] Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. *ECCV*, 2007. 7

[24] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *NeurIPS*, 2020. 2, 3, 5, 6

[25] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15:1929–1958, 2014. 2

[26] Yannick Strümpler, Janis Postels, Ren Yang, Luc Van Gool, and Federico Tombari. Implicit neural representations for image compression. In *ECCV*, 2022. 2

[27] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 2

[28] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. *IJCV*, 128:1867–1888, 2018. 2

[29] Tomas Vaškevičius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. *NeurIPS*, 2019. 7

[30] Krishna Wadhwani and Tamaki Kojima. Squeezenerf: Further factorized fastnerf for memory-efficient inference. *CVPR*, 2022. 2

[31] Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *Communications in Computational Physics*, 2019. 7

[32] Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *ArXiv*, abs/2006.14548, 2020. 3

[33] Yunfan Zhang, Ties van Rozendaal, Johann Brehmer, Markus Nagel, and Taco Cohen. Implicit neural video compression. *ArXiv*, abs/2112.11312, 2021. 2

[34] Peng Zhao, Yun Yang, and Qiao-Chu He. Implicit regularization via hadamard product over-parametrization in high-dimensional linear regression. *arXiv: Statistics Theory*, 2019. 7