Regularize implicit neural representation by itself

Zhemin Li¹ Hongxia Wang² Deyu Meng²,³
¹ National University of Defense Technology
² Xi’an Jiaotong University
³ Macau University of Science and Technology
lizhemin@nudt.edu.cn, wanghaixia@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) \( \phi_{\theta}(x) : \mathbb{R}^d \to \mathbb{R}^c \) to approximate the explicit solution of an implicit function \( F(x, \phi_{\theta}, \nabla_x \phi_{\theta}, \nabla_x^2 \phi_{\theta}, \ldots) = 0 \). For an example, we can represent a gray-scale image \( X \in \mathbb{R}^{m \times n} \) with an INR \( \phi_{\theta}(x) : \mathbb{R}^2 \to \mathbb{R} \) which satisfied \( \phi_{\theta}(\frac{x}{m}, \frac{y}{n}) = X_{ij}, i \in \{1, \ldots, m\}, j \in \{1, \ldots, n\} \). Compared with traditional grid representation \( X \), 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 \( 256 \times 256 \) 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 \( \phi_{\theta}(x) \). 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 \( X \). 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-
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 $\phi_{\theta}(x) : \mathbb{R}^d \rightarrow \mathbb{R}^o$ to approximate the explicit representation of an implicit function $F(x, \phi_{\theta}, \nabla_x \phi_{\theta}, \nabla^2_x \phi_{\theta}, \ldots) = 0$, where the FCN has $L$ hidden layers defined as follows,

$$z^{(\ell)} = W^{(\ell)} x^{(\ell-1)} + b^{(\ell)}, \quad x^{(\ell)} = \sigma(z^{(\ell)})$$

$$\phi_{\theta}(x) = z^{(L+1)}, \quad \ell = 1, 2, \ldots, L + 1,$$

with $\sigma(\cdot)$ an element-wise activation function and $x^{(0)} = x$.
\[ \omega_0 = 1 \quad \omega_0 = 10 \quad \omega_0 = 30 \quad \omega_0 = 80 \quad \omega_0 = 200 \]
\[ \delta = 10 \quad \delta = 100 \quad \delta = 200 \quad \delta = 700 \quad \delta = 900 \]

Figure 3. Fitting a 256 × 256 Cameraman which random missing 50% pixels with (a) SIREN and (b) NTK, respectively. \( \omega_0 \) and \( \delta \) are the hyper-parameters of models. (c,d) shows the PSNR value change with \( \omega_0, \delta \) at different random missing rate, respectively.

\[ \theta = \{ W^{(l)}, b^{(l)} \mid l = 1, 2, \ldots, L + 1 \} \sim D \text{ at initialization}, \quad W^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}, \text{ and } n_L \text{ is the width of } l-\text{th layer} \text{ with } n_0 = d, n_{L+1} = o. \]

For simplicity, we focus on a special case of INR with \( F(\theta, x, z) = \| \phi_\theta(x) - z(x) \|_2^2 = 0 \) and \( \phi_\theta(x) : \mathbb{R}^2 \mapsto \mathbb{R} \), where \( z(x) : \mathbb{R}^2 \mapsto \mathbb{R} \) 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

\[
\theta^* = \arg \min_{\theta} \left\{ \mathcal{L}(\theta, \mathcal{X}, \mathcal{Z}) = \sum_{(x_i, z_i) \in \mathcal{X} \times \mathcal{Z}} F(\theta, x_i, z_i) \right\},
\]

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

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

\[ \phi_{\text{NTK}}(x) = \sum_{i=1}^{N} (K^{-1}z)_{i} k_{\text{NTK}}(x_i, x), \]

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

\[ K_{ij} = k_{\text{NTK}}(x_i, x_j) = \mathbb{E}_{\theta \sim D} \left( \frac{\partial \phi_\theta(x_i)}{\partial \theta}, \frac{\partial \phi_\theta(x_j)}{\partial \theta} \right). \]

In this paper, we consider INR \( \phi' : \mathbb{R}^{2D} \mapsto \mathbb{R} \) with a feature map \( \gamma(x) = \frac{1}{\sqrt{D}}[\cos Bx^T, \sin Bx^T]^T : \mathbb{R}^d \mapsto \mathbb{R}^{2D} \) as its input, where \( x \in \mathbb{R}^d, B \in \mathbb{R}^{D \times d}, \) and \( B_{ij} \sim N(0, \delta) \). Then \( \phi_{\text{NTK}}(\gamma(x)) \) 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 \( \delta \) globally. Especially when \( \delta \) 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 \( \phi_\theta(\cdot) : \mathbb{R}^{2D} \mapsto \mathbb{R} \) with \( x \in \mathbb{R}^d, \gamma(x) \in \mathbb{R}^{2D}, \) and the feature map \( \gamma(x) = \frac{1}{\sqrt{D}}[\cos Bx^T, \sin Bx^T]^T \) with \( B \in \mathbb{R}^{D \times d} \) and \( B_{ij} \sim N(0, \delta) \). Denote the corresponding composed NTK as \( k_D(x_i, x_j) = h_{\text{NTK}}(\frac{1}{\sqrt{D}}[\cos (B(x_i - x_j))]) \), then we have

\[
\lim_{D \to \infty} k_D(x_j, x_j) = h_{\text{NTK}}(e^{-\delta^2\|x_i - x_j\|^2}).
\]

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

\[
\lim_{\delta \to \infty} \phi_{\text{NTK}}'(\gamma(x)) = \left\{ \begin{array}{ll}
\begin{array}{l}
z_{l-1} \\
\frac{h(0)1_x}{(N-1)h(0)+h(1)} x_i \in \{x_i\}_{i=1}^N,
\end{array}
\end{array} \right. \quad x \notin \{x_i\}_{i=1}^N.
\]

INR needs to be regularized. We validate Corollary 1 by exploring the performance of \( \phi_{\text{NTK}}'(\gamma(x)) \) in image inpainting task with different missing rates. As Figure 3(b) shows, when \( \delta = 900, \phi_{\text{NTK}}(\gamma(x)) \) 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 \( \omega_0 \) in the first layer as \( \sin(\omega_0 Wx + b) \).

Based on the numerical result, the optimal \( \omega_0 \) or \( \delta \) is required so that INR generalizes the best. However, finding an optimal \( \omega_0 \) or \( \delta \) with non-uniformly sampled training data
is impossible. Figure 3(c,d) illustrates that the optimal \( \omega_0 \) or \( \delta \) 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 \( L = \sum_{(x_i, z_i) \in \mathcal{X} \times \mathcal{Z}} \| \phi_\theta(x_i) - z_i \|_2^2 \), 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 \( \mathcal{L}(\theta, \mathcal{X}, \mathcal{Z}) + \lambda \mathcal{R}(\theta, \mathcal{X}, \mathcal{Z}) \), where \( \lambda \) 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

\[
\mathcal{R}_\text{DE} = \text{tr} (\mathbf{X}^\top \mathbf{L} \mathbf{X}) = \sum_{1 \leq i,j \leq m} \mathbf{A}_{ij} \| \mathbf{X}_{i,:} - \mathbf{X}_{j,:} \|_2^2,
\]

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 \( \phi_\theta(x) \).

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:

\[
\begin{align*}
\mathcal{R}(\theta) &= \text{tr} \left( [\mathcal{T}(\mathbf{X})]^\top \mathbf{L}(\theta) \mathcal{T}(\mathbf{X}) \right) \\
\mathbf{L}(\theta) &= \mathbf{A}(\theta) \cdot \mathbf{1}_{m' \times m'} \odot \mathbf{1}_{m'} - \mathbf{A}(\theta) \\
\mathbf{A}(\theta) &= \exp(g(\theta; \mathbf{u})g(\theta; \mathbf{v}))1_{m'}
\end{align*}
\]

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

\[
\text{minimize}_{\theta, \theta', \theta''} \{ \mathcal{L}(\theta, \mathcal{X}, \mathcal{Z}) + \lambda_r \mathcal{R}(\theta_r) + \lambda_c \mathcal{R}(\theta_c) \},
\]

where \( \mathcal{R}(\theta_r) \) and \( \mathcal{R}(\theta_c) \) are row and column regularizers respectively. \( \mathcal{T}(\mathbf{X}) = \mathbf{X} \) in \( \mathcal{R}(\theta_r) \), and \( \mathcal{T}(\mathbf{X}) = \mathbf{X}^\top \) in \( \mathcal{R}(\theta_c) \). \( \lambda_r, \lambda_c \) are used to balance the fidelity and regularization terms.

As the self-similarity which is represented by \( g(\theta; \cdot; \cdot) \) or \( g(\theta''; \cdot; \cdot) \) are much simpler than the image, so the parameter number of \( g(\theta; \cdot; \cdot) \) or \( g(\theta''; \cdot; \cdot) \) are much lesser than the one of \( \phi_\theta(\cdot) \), 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 patterns include random missing, patch missing, and textural missing. Four images are tested including Baboon, Barbara, Man and Boats.

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

Figure 4. Residual of image inpainting, i.e., $|X - X^*|$ 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: $\mathcal{R}_{TV} = \sum_{(x_i, y_i) \in \mathcal{G}} \| \nabla_x \phi(\mathbf{X}_i) \|_1$, $\mathcal{G} = \{ \frac{1}{m}, \ldots, \frac{m}{m} \} \times \{ \frac{1}{n}, \ldots, \frac{n}{n} \}$, which is the discrete version on $\mathcal{G}$.
2. $L_2$: $\mathcal{R}_{L_2} = \sum_{\ell=1}^{L+1} \| \mathbf{W}^{(\ell)} \|_2$ 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 $h(\cdot) : \mathbb{R}^{N_0+2} \rightarrow \mathbb{R}$ 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, INR, 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 $|X - X^*|$ 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
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 \( L \) 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 \( L \) learned by AIR and INRR, respectively. The \( L \) learned by AIR (Figure 5(a)) is discontinuous with high probability at those locations that are not sampled, while the \( L \) learned by INRR (Figure 5(b)) is much more continuous. The continuous \( L \) introduced by the tiny INR is more consistent with practice.

![Figure 5](image)

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 \( X(t+1) \) involves both \( X(t) \) and \( L(t) \), and the update of \( L(t) \) depends on \( X(t-1) \). To understand the training dynamics, we consider the following simplified model:

\[
\text{minimize}_{X,L} \{ \mathcal{L} + \lambda \text{tr}(X^T LX) \},
\]

and we have

\[
\nabla_{X(t)} L = \nabla_{X(t)} \mathcal{L} + 2\lambda L(t)X(t),
\]

where \( \mathcal{L} \) is the fidelity term, \( L(t) \) is the function of \( \{X(t_0) | t_0 < t \} \) as \( L(t) \) is updated based on \( X(t-1) \).
Therefore, every iteration step of INRR leverages all the previously learned information \( \{X(t_0) \mid t_0 < t\} \). Note that the update of both vanilla INR, INR+TV, and INR+\( L_2 \) only depend on \( X(t) \). 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 \( \{(x_i, y_j) \mid i, j\} \) is a uniform \( 256 \times 256 \) 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 \( L_r(t) \) and \( L_c(t) \) for Baboon are shown in Figure 8. A few large blocks appear in \( L_r(500) \) and \( L_c(500) \) 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
Figure 8. Learned $L_r(t)$ and $L_c(t)$ during training. (a): first and second rows depict the Baboon image and its rotation. (b)-(d): first/second row shows the heatmap of $L_r/L_c$ at different $t$. A darker color indicates a stronger similarity captured by the adaptive regularizer. The $(i, j)$-th element in the heatmap of $L_r(t)$ 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.

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

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


[3] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Ding-Xuan Zhou. Deep image prior powered by red. IJCV, 2019. 2


