

MULTI-HEAD RELU IMPLICIT NEURAL REPRESENTATION NETWORKS

Arya Aftab^{1,2}, Alireza Morsali³, Shahrokh Ghaemmaghami^{1,2}

¹ Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

² Electronics Research Institute, Sharif University of Technology, Tehran, Iran

³ Department of Electrical and Computer Engineering, McGill University, Montreal, Canada

aftab.arya@ee.sharif.edu, alireza.morsali@mail.mcgill.ca, ghaemmag@sharif.edu

ABSTRACT

In this paper, a novel multi-head multi-layer perceptron (MLP) structure is presented for implicit neural representation (INR). Since conventional rectified linear unit (ReLU) networks are shown to exhibit spectral bias towards learning low-frequency features of the signal, we aim at mitigating this defect by taking advantage of local structure of the signals. To be more specific, an MLP is used to capture the global features of the underlying generator function of the desired signal. Then, several heads are utilized to reconstruct disjoint local features of the signal, and to reduce the computational complexity, sparse layers are deployed for attaching heads to the body. Through various experiments, we show that the proposed model does not suffer from the special bias of conventional ReLU networks and has superior generalization capabilities. Finally, simulation results confirm that the proposed multi-head structure outperforms existing INR methods with considerably less computational cost. The source code is available at <https://github.com/AlirezaMorsali/MH-RELU-INR>

Index Terms— Implicit neural representation, multi-head MLP, ReLU network, spectral bias, multi-layer perceptron

1. INTRODUCTION

Recently, there has been a considerable interest in implicit neural representation (INR) for parameterizing various kinds of signals [1–4]. For example, images and audio signals are conventionally stored as discrete grids of pixels and discrete samples of amplitudes, respectively [5, 6]. On the other hand, with INR, the goal is to find a continuous generator function for the target signal [7–14]. In particular, for images as the target signal of INR, the pixel coordinates are mapped to RGB color values.

Multi-layer perceptron (MLP) networks and, in particular, deep neural networks (DNN) are shown to have an unprecedented capability in learning various input-output transformations (even random input-output mappings) with high accuracy [15, 16]. One of the widely used activation functions

in DNNs is rectified linear unit (ReLU), which brings considerable advantages, such as overcoming gradient vanishing and facilitating the learning process. Furthermore, it is proven that ReLU neural networks can express any continuous piece-wise linear function, which forms bounded convex polytopes. Moreover, every continuous piece-wise linear function can be defined by a specific ReLU neural network [17–19]. As a result, ReLU networks are known as universal approximators. Consequently, ReLU networks could be the prime candidate for INR. However, in [20], the Fourier transform of ReLU neural networks reveals a spectral bias in such networks, i.e., ReLU networks tend to learn lower frequencies faster.

It is shown that an increase in the complexity of the data manifold shapes and low data dimension eases the learning of the high frequencies [2]. Therefore, authors in [2] proposed manipulating the input data by a sine kernel method to facilitate learning high-frequency signals. Positional encoding relieves the spectral bias by kernel regression module, which maps a low-dimension input to a high-dimension one through a set of sine and cosine functions with different frequencies. Furthermore, the use of periodic activation functions for INR is presented in [3], which allows the model (namely, SIREN) to learn high-frequency data more effectively. Nevertheless, a major drawback of the aforementioned methods is the computational cost of such networks during both training and inference. Moreover, in case of SIREN, due to the use of periodic activation function, the network is highly sensitive to initialization and can exhibit unstable behaviours for different types of signals.

In this paper, we present an efficient multi-head INR network structure which is capable of learning high-frequency signals accurately with considerably lower computational cost compared to existing methods. The target signals of INR, for instance image and audio, usually exhibit local structure and neighbourhood dependencies. By exploiting this property of the target signals, we show that ReLU networks can be boosted to be less partial to low-frequency components during the training phase. We introduce a multi-head network architecture in which the body learns the global features of the signal and the output layers, consists of several

This work was supported by DGSculptor (www.dgsculptor.com).

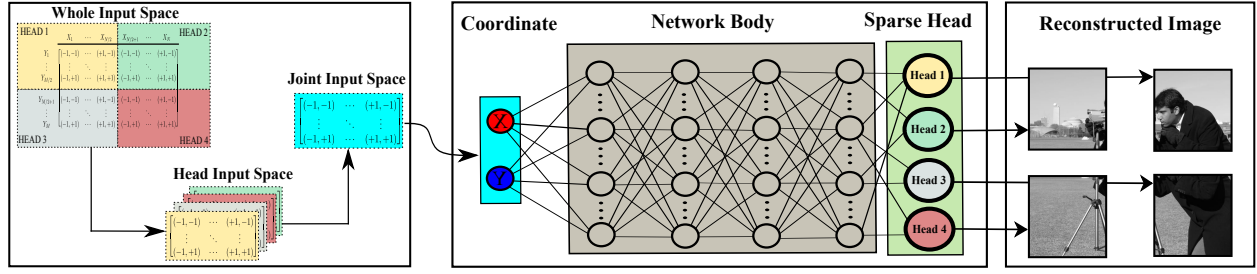


Fig. 1: Network Architecture for 4 Heads

heads, reconstruct separate parts of the signal, and learn local features of the signal. This approach has several promising advantages: I. The network architecture can be tailored to learn high-frequency components without incorporating periodic functions. II. The number of floating point operations (FLOPs) required to generate a signal decreases significantly as the number of heads of the model increases. III. The generalization ability of the model intensifies in the multi-head models compared to the base model.

2. PROBLEM FORMULATION

In this work, we focus on grayscale images for INR. However, the proposed method can be generalized to other multi-dimensional signals. For a given $N_x \times N_y$ grayscale image, i.e., $\mathbf{I} \in \mathbb{R}^{N_x \times N_y}$, the goal of INR is to find an underlying function $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}$ that maps pixel coordinates r and c into the pixel value $\mathbf{I}[r, c]$ for $r = 1, 2, \dots, N_x$ and $c = 1, 2, \dots, N_y$ where $\mathbf{I}[r, c]$ denote the grayscale value of the \mathbf{I} on the r th row and the c th column. Since the function Φ is continuous, we only have its values at discrete points, i.e.,

$$\Phi(x(r), y(c)) = \mathbf{I}[r, c] \quad (1)$$

where $x(r) = 2\frac{r-1}{N_x-1} - 1$, $y(c) = 2\frac{c-1}{N_y-1} - 1$. Note that any arbitrary interval can be selected as the domain of the function; however, for simplicity $x, y \in [-1, 1]$ is usually chosen.

Existing works in the literature [2, 3] use an MLP to approximate $\Phi(x, y)$ and train the MLP using the pixel coordinate and the corresponding pixel values as the dataset:

$$\mathcal{D} = \left\{ \left((x(r), y(c)), \mathbf{I}[r, c] \right) \right\}_{r,c=1}^{N_x, N_y}.$$

3. PROPOSED METHOD

In order to alleviate the spectral bias of the ReLU networks, we take advantage of the local structure of the target signal by dividing the input domain x and y into H_x and H_y equal intervals, respectively, where H_x and H_y are non-negative integers. Let us further assume N_x and N_y are divisible by H_x and H_y , respectively. Consequently, the image \mathbf{I} can be divided into $M = H_x H_y$ equal grid cells, each of which is of size $\hat{N}_x \times \hat{N}_y$ where $\hat{N}_x = \frac{N_x}{H_x}$ and $\hat{N}_y = \frac{N_y}{H_y}$. Therefore,

for $\hat{r} = 1, 2, \dots, \hat{N}_x$ and $\hat{c} = 1, 2, \dots, \hat{N}_y$, we can further explicitly write the image cells as

$$\mathbf{I}_{l,k}[\hat{r}, \hat{c}] = \mathbf{I}[\hat{N}_h(l-1) + \hat{r}, \hat{N}_w(k-1) + \hat{c}] \quad (2)$$

where $l = 1, 2, \dots, H_x$ and $k = 1, 2, \dots, H_y$.

Consequently, the INR of a given image \mathbf{I} is broken down into finding M functions. Thus, instead of having a generator function $\Phi(x, y)$ for the whole image \mathbf{I} , we use M functions $\phi_{l,k}(\hat{x}, \hat{y})$ for the corresponding cell of the image $\mathbf{I}_{l,k}$, i.e.,

$$\phi_{l,k}(\hat{x}(\hat{r}), \hat{y}(\hat{c})) = \mathbf{I}_{l,k}[\hat{r}, \hat{c}] \quad (3)$$

where $\hat{x}(\hat{r}) = 2((\hat{r}-1)/\hat{N}_x) - 1$, $\hat{y}(\hat{c}) = 2((\hat{c}-1)/\hat{N}_y) - 1$.

However, training M MLPs is not efficient. Moreover, the desired signals (images in this case) usually contain global features which can be shared by all M functions. Motivated by this fact, we use a function (as body) to produce global features denoted by $\psi(\hat{x}, \hat{y})$. This function, in essence, behaves as an embedding that maps coordinates to a high-dimensional space. Subsequently, M disjoint rendering functions $\tau_{l,k}(\cdot)$ use these mapped coordinates to reconstruct the details of each cell of the image. We can therefore write:

$$\phi_{l,k}(\hat{x}, \hat{y}) = \tau_{l,k}(\psi(\hat{x}, \hat{y})). \quad (4)$$

To find such functions via training, we can write the following optimization problem:

$$\min_{\tau_{l,k}(\cdot), \psi(\cdot)} \sum_{\hat{r}=1}^{\hat{N}_h} \sum_{\hat{c}=1}^{\hat{N}_w} \sum_{l=1}^{H_x} \sum_{k=1}^{H_y} \left(\tau_{l,k}(\psi(\hat{x}(\hat{r}), \hat{y}(\hat{c}))) - \mathbf{I}_{l,k}[\hat{r}, \hat{c}] \right)^2. \quad (5)$$

3.1. Network Architecture

Here, we take advantage of expressive power of MLPs to approximate the rendering functions $\tau_{l,k}(\cdot)$ as well as the body function $\psi(\hat{x}, \hat{y})$. In particular, we propose a multi-head implicit neural representation network consisting of two parts, namely, the body and the rendering heads. The body is a regular MLP with ReLU activation function. While, the rendering networks are a special sparse layer with multiple outputs. Fig. 1 illustrates the structure of the proposed network, which is explained below in details.

3.1.1. Body

A 4-layer MLP with ReLU activation function is used as the body to approximate $\psi(\hat{x}, \hat{y})$. This MLP takes the normalized

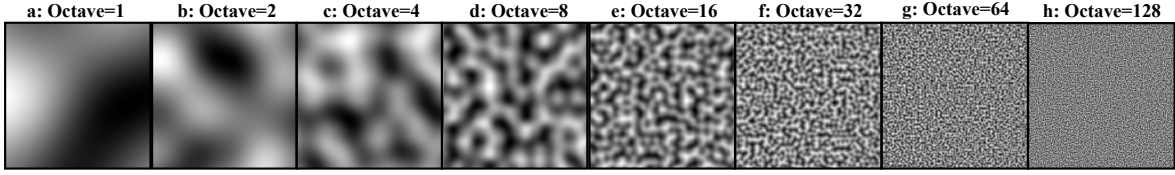


Fig. 2: Perlin noise

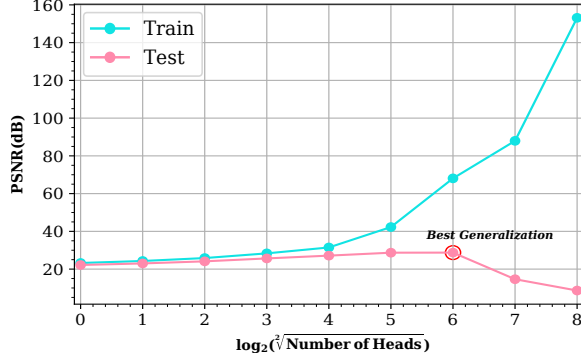


Fig. 3: Generalization ability of the proposed method in different numbers of heads

coordinates of a pixel and creates an intermediate vector that embeds the coordinate into high-dimensional space that is fed into the rendering networks.

3.1.2. Head

One neuron is used for each rendering head $\tau_{l,k}(\cdot)$ to approximate the pixel values of the corresponding cell of the image. The main advantage of using multiple heads to reconstruct different parts is the ability of the network to reconstruct several pixels in each forward pass, which leads to a significant reduction in model computations to reconstruct the whole image.

Alternatively, all of the single neurons can be considered as a fully connected layer, namely, the rendering multi-head layer, which is attached to the body network. The main issue with a fully-connected rendering multi-head network is the large number of required parameters, which is proportional to the number of heads connected to the model body. In particular, the number of parameters grows significantly larger as the number of heads increase. For example, if 256^2 heads are used for a 256×256 grayscale image and the last layer of the model body has 256 neurons, then approximately 16.7 million (M) parameters are required.

To solve the problem, we use a sparse layer instead of the fully-connected one. In the sparse layer, the number of trainable parameters is significantly reduced due to omission of most of the connections with the previous layer. Let us denote the number of connections of each neuron with the previous layer by α . Indeed, the minimum value of α is 1, and by increasing this parameter to the number of neurons in the previous layer, the sparse layer becomes a fully-connected layer.

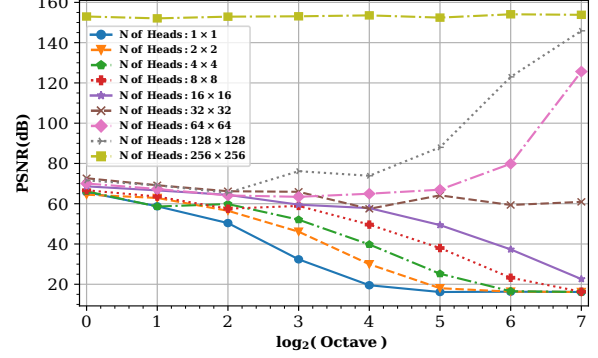


Fig. 4: Spectral bias in the proposed method with different number of heads

3.2. Model Configuration

To further analyse the model and tuning the hyper-parameters, a base body network with 4-hidden layers and 256 neurons for each layer is used. Each rendering head (output neuron) is connected to the body network with a partially-connected linear layer (sparse layer) that has only α connections to the output layer of the body. The active connections are randomly selected with a uniform distribution at the beginning of the training and do not change. For all the experiments and simulations, the weights and biases of the networks are initialized with uniform distribution, and training is performed for 2000 epochs.

3.3. Spectral Bias

Here, we present experimental results to show the effect of number of heads on the spectral bias of the model. Specifically, we use 2D Perlin noise [21] as the target image. Perlin noise is a pseudo-random pattern of float values generated across an N-dimensional plane which allows for controlled high-frequency features in each dimension using a parameter, namely, Octave. In this experiment, Perlin noises with different frequencies are generated in two dimensions with a size of 256×256 , shown in Fig. 2.

We trained the proposed model with the different number of heads on this set of synthetic noises. The base model presented in Subsection 3.2 is used for this experiment with $\alpha = 32$. In Fig. 4, peak signal-to-noise ratio (PSNR) of the reconstructed image with different network configurations for various Octaves of Perline noise are presented.

Note that, the higher the Octave the higher the frequency of the target image. The solid blue curve which corresponds to the network with 1 head (which is in essence the regular

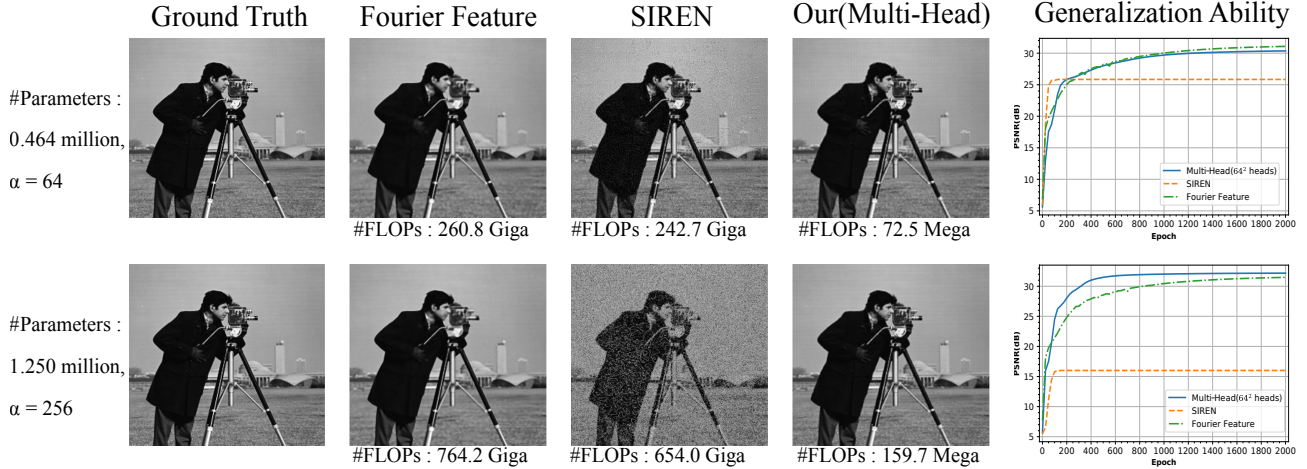


Fig. 5: Comparison of the proposed method(64^2 heads) with the SIREN and Fourier feature in terms of generalization ability and computational cost. The computational cost is calculated for the evaluation phase (image’s size of 512×512).

ReLU network) clearly illustrates the spectral bias of such ReLU networks. However, as the number of heads increase, the model can more effectively reconstruct higher frequencies. In particular, by increasing the number of heads, each rendering unit (head) reconstructs a smaller area of the image. The amount of variation in the neighbouring pixels decreases dramatically with decreasing size of each cell, and in practice, each head reconstructs a low-frequency signal.

Since each rendering head represents a hyperplane, if the number of model heads is equal to the number of image pixels, the hyperplane associated with each head can approximate that single pixel perfectly. In fact, for the case of 256^2 where each head only reconstructs one pixel and the whole image is reconstructed with one forward pass, the error-free approximation of the network is shown in Fig. 4 regardless of frequency variations.

3.4. Generalization ability

In INR tasks, only memorizing the training data is not enough. Specifically, in addition to the ability of the proposed model to alleviate the frequency bias of ReLU networks, the generalization ability must be also considered. For example, in neural radiance fields (NERF) [4], continuous scenes are represented as 5D neural radiance fields, parameterized as MLP networks; if the model fails to create new scenes, it becomes practically useless.

Consequently, we presented an experiment to evaluate the generalization ability of the proposed model. For this experiment, we first resizes a 512×512 grayscale image to a 256×256 image. Then, the proposed model with different heads is trained on the resized image. Finally, the trained model’s performance is evaluated on the original 512×512 image. Fig. 3 shows the results of this experiment. We can observe that although the model’s performance on the training data improves by increasing the number of heads, from

the number of 64^2 heads on-wards, we encounter a decrease in the generalization ability. The base model used in this experiment has 256^2 heads and the $\alpha = 32$. Consequently, in the rest of the paper, we focus on a model with 64^2 as it exhibits the best generalization characteristics.

4. EXPERIMENTAL RESULTS

In this section, in order to compare our method with the state-of-the-art methods, we investigate the accuracy of different models with the same number of parameters. We also report number of floating point operations (FLOPs) for training each model. Fig. 5 illustrates the reconstructed image of the proposed model as well as SIREN [3] and Fourier feature [2]. To ensure a fair comparison, SIREN and Fourier feature are configured (the best possible configurations are selected based on [2,3]) to have the same number of parameter as the multi-head network with $\alpha = 64$ and 256 , i.e., 0.464 and 1.250 million, respectively. With 64^2 heads, the proposed model with 0.464M parameters requires **3347** times fewer FLOPs compared to competitors to reconstruct the whole image, which shows its outstanding performance. It should be noted that by increasing the model’s size from 0.464M to 1.250M, the reduction of the required FLOPs becomes more pronounced (**4095** times).

5. CONCLUSION

In this paper, a novel multi-head INR was proposed to improve the spectral bias of ReLU networks. In particular, tackle this issue by exporting the local structure of the signals. Specifically, an MLP is used as the body to capture the global features and several heads are used to reconstruct local features. The proposed structure requires considerably less computational cost while achieving superior or similar results to the state-of-the-art methods.

6. REFERENCES

- [1] J. J. Park, P. Florence, R. N. J. Straub, and S. Lovegrove, “DeepSDF: Learning continuous signed distance functions for shape representation,” in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2019, pp. 165–174.
- [2] M. Tancik, P. P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. T. Barron, and R. Ng, “Fourier features let networks learn high frequency functions in low dimensional domains,” in *Conf. Neural Inf. Process. Syst. (NeurIPS)*, June 2020.
- [3] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, “Implicit neural representations with periodic activation functions,” in *Conf. Neural Inf. Process. Syst. (NeurIPS)*, June 2020.
- [4] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NERF: Representing scenes as neural radiance fields for view synthesis,” in *European Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 405–421.
- [5] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, “Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision,” *arXiv preprint arXiv:1612.00814*, 2016.
- [6] S. Lombardi, T. Simon, J. Saragih, G. Schwartz, A. Lehrmann, and Y. Sheikh, “Neural volumes: Learning dynamic renderable volumes from images,” *arXiv preprint arXiv:1906.07751*, 2019.
- [7] M. Meshry, D. B. Goldman, S. Khamis, H. Hoppe, R. Pandey, N. Snavely, and R. Martin-Brualla, “Neural rerendering in the wild,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2019, pp. 6871–6880.
- [8] K.-A. Aliev, A. Sevastopolsky, M. Kolos, D. Ulyanov, and V. Lempitsky, “Neural point-based graphics,” in *European Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 696–712.
- [9] J. Thies, M. Zollhöfer, and M. Nießner, “Deferred neural rendering: Image synthesis using neural textures,” *ACM Trans. Graph. (TOG)*, vol. 38, no. 4, pp. 1–12, July 2019.
- [10] L. Liu, W. Xu, M. Zollhoefer, H. Kim, F. Bernard, M. Habermann, W. Wang, and C. Theobalt, “Neural rendering and reenactment of human actor videos,” *ACM Trans. Graph. (TOG)*, vol. 38, no. 5, pp. 1–14, Nov. 2019.
- [11] M. Atzmon and Y. Lipman, “SAL: Sign agnostic learning of shapes from raw data,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2020, pp. 2562–2571.
- [12] I. Mehta, M. Gharbi, C. Barnes, E. Shechtman, R. Ramamoorthi, and M. Chandraker, “Modulated periodic activations for generalizable local functional representations,” *arXiv preprint arXiv:2104.03960*, 2021.
- [13] V. Sitzmann, J. Thies, F. Heide, M. Niessner, G. Wetzstein, and M. Zollhofer, “Deepvoxels: Learning persistent 3d feature embeddings,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, June 2019, pp. 2432–2441.
- [14] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutional occupancy networks,” in *European Conf. Comput. Vis. (ECCV)*, Aug. 2020, pp. 523–540.
- [15] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio *et al.*, “A closer look at memorization in deep networks,” in *Int. Conf. Mach. Learn. (ICML)*. PMLR, July 2017, pp. 233–242.
- [16] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” *Commun. ACM*, vol. 64, no. 3, pp. 107–115, Mar. 2021.
- [17] S. Arora, N. Cohen, and E. Hazan, “On the optimization of deep networks: Implicit acceleration by overparameterization,” in *Int. Conf. Mach. Learn. (ICML)*. PMLR, July 2018, pp. 244–253.
- [18] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, “Understanding deep neural networks with rectified linear units,” *arXiv preprint arXiv:1611.01491*, 2016.
- [19] B. Hanin, “Universal function approximation by deep neural nets with bounded width and relu activations,” *Math.*, vol. 7, no. 10, p. 992, Oct. 2019.
- [20] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, and A. Courville, “On the spectral bias of neural networks,” in *Int. Conf. Mach. Learn. (ICML)*. PMLR, July 2019, pp. 5301–5310.
- [21] K. Perlin, “Improving noise,” in *Proc. Annu. Conf. Comput. Graph. interactive Techn.*, July 2002, pp. 681–682.