

# Intelligent Reflecting Surface OFDM Communication with Deep Neural Prior

Tomer Fireaizen<sup>1</sup>      Gal Metzer<sup>2</sup>      Dan Ben-David<sup>1</sup>      Yair Moshe<sup>1</sup>      Israel Cohen<sup>1</sup>      Emil Björnson<sup>3</sup>

<sup>1</sup>Signal and Image Processing Laboratory (SIPL), Faculty of Electrical & Computer Engineering  
Technion – Israel Institute of Technology, Haifa, Israel, <https://sipl.eelabs.technion.ac.il>

<sup>2</sup>Faculty of Engineering, Tel Aviv University, Tel Aviv, Israel

<sup>3</sup>Dept. of Electrical Engineering (ISY), Linköping University and KTH Royal Institute of Technology, Sweden

**Abstract**—An Intelligent Reflecting Surface (IRS) is an emerging technology for improving the data rate over wireless channels by controlling the underlying channel. In this paper, we describe a novel solution for IRS configuration to maximize the data rate over wideband channels. The optimization is obtained by online training of a deep generative neural network. Inspired by related works in image processing, this network is randomly initialized and acts as a regularization term for the optimization process since the structure of the generator is sufficient to capture a great deal of IRS statistics prior to any learning. In contrast to recent deep learning techniques for IRS configuration, the proposed technique does not require an offline training stage and can adapt quickly to any environment. Compared to the previous state-of-the-art algorithm, the proposed method is significantly faster and obtains IRS configurations that achieve higher data transmission rates.

**Index Terms**—Intelligent reflecting surface (IRS), Reconfigurable Intelligent Surface (RIS), passive beamforming, OFDM, deep neural prior.

## I. INTRODUCTION

An Intelligent Reflecting Surface (IRS) [1] is a two-dimensional surface of metamaterial whose interaction with electromagnetic waves can be controlled. It consists of an array of  $N_H \times N_W = N$  discrete passive elements of sub-wavelength size with reflection coefficients having configurable amplitude and/or phase responses, where  $N_H$  and  $N_W$  are the number of elements in each IRS column and row, respectively. By tuning these elements, the flat surface can mimic the reflection properties of a differently shaped object, thereby allowing the array to intelligently reconfigure how an incident wave is reflected [2], [3].

Each IRS element can only have a single configuration, although the channel changes its properties over the frequency domain. Most prior works on IRS configuration optimization assume narrowband communication systems [4], which circumvents this issue by limiting the bandwidth. However, the obtained solutions are not useful in modern communication systems operating over wideband channels. It is therefore essential to develop algorithms that provide good IRS configurations for wideband channels, particularly using orthogonal frequency-division multiplexing (OFDM) [5]–[8]. The physical implementation of such a system encounters

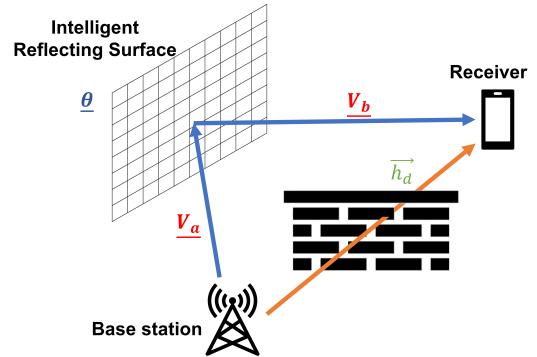


Fig. 1: IRS-assisted wireless communications, where the base station has NLOS to the receiver, but it does have LOS to the IRS, which is configured to improve data transmission rate.

many challenges, such as limited pilot signaling, hardware impairments, low signal-to-noise ratio (SNR), and unknown fading conditions. Therefore, it is important to design a generic algorithmic solution for utilizing IRS technology that is based on minimal prior knowledge of the properties of the specific IRS, and employs as few stages as possible [9], [10]. The goal of this paper is to characterize the behavior of an IRS based on the received signals from over-the-air signaling, i.e., develop an efficient control algorithm to configure the surface to increase communication performance [11].

## II. SYSTEM MODEL

We consider the practical setup where the IRS is deployed to have a line-of-sight (LOS) channel to the base station [5]. However, the IRS may or may not have LOS to the user's receiver equipment, as illustrated in Fig. 1. We further assume that all users have non-LOS (NLOS) channels to the base station, since this is the setup where the IRS can make the largest difference in the link budget. Users who have LOS to the IRS will be referred to as *LOS users*, while those who have NLOS to the IRS will be referred to as *NLOS users*. The transmitting base station and receiving user have a single antenna each.

We will adopt the system modeling from [2]. More precisely, the relation in the discrete-time domain between the transmitted signal  $\mathbf{x}[k]$  and the received signal  $\mathbf{z}[k]$  is modeled as [2]

$$\mathbf{z}[k] = \sum_{\ell=0}^{M-1} h_{\boldsymbol{\theta}}[\ell] \mathbf{x}[k-\ell] + \mathbf{w}[k] \quad (1)$$

where  $k$  is the time index and  $\{h_{\boldsymbol{\theta}}[\ell] : \ell = 0, \dots, M-1\}$  is the finite impulse response (FIR) representation of the channel from the base station to the receiver with  $M$  taps. Moreover,  $\boldsymbol{\theta} \in \mathbb{C}^{N \times 1}$  with unit-modulus entries representing the IRS configuration and  $\mathbf{w}[k]$  is the receiver noise. The transmission over the communication channel will be carried out using OFDM with  $K$  subcarriers, where  $K > M$ . We can write (1) in the frequency domain as

$$\mathbf{z}^f = \mathbf{h}_{\boldsymbol{\theta}}^f \odot \mathbf{x}^f + \mathbf{w}^f \quad (2)$$

$$\mathbf{h}_{\boldsymbol{\theta}}^f = \mathbf{h}_d^f + \mathbf{V}^f \boldsymbol{\theta}, \quad (3)$$

where  $\mathbf{x}^f$ ,  $\mathbf{z}^f$ ,  $\mathbf{h}_d^f$  and  $\mathbf{w}^f$  are vectors of size  $K \times 1$  representing the frequency-domain transmitted signal, received signal, direct channel and noise, respectively, and  $\odot$  denotes the Hadamard element-wise product. The standard mapping from the time to frequency domain is omitted for brevity, but can be found in [2]. The matrix  $\mathbf{V}^f$  is a  $K \times N$  matrix representing the cascade of the channel from the transmitter to the IRS and the channel from the IRS to the receiver, and  $\boldsymbol{\theta}$  is a vector of size  $N \times 1$  representing the configuration of the IRS. Based on this model, the data rate is computed by [2]:

$$\mathcal{R} = \frac{B}{K+M-1} \sum_{\nu=0}^{K-1} \log_2 \left( 1 + \frac{P |h_{\boldsymbol{\theta}}^f[\nu]|^2}{BN_0} \right) \quad (4)$$

where  $P$  is the transmit power,  $B$  is the bandwidth,  $N_0$  is the noise power spectral density, and  $h_{\boldsymbol{\theta}}^f[\nu]$  is an entry of  $\mathbf{h}_{\boldsymbol{\theta}}^f$  for  $\nu = 0, \dots, K-1$ . It represents each of the subcarrier channels from the base station to the receiver via the IRS for configuration  $\boldsymbol{\theta}$ . This rate expression is a summation over the  $K$  subcarriers, which is then divided by  $K+M-1$  to compensate for the cyclic prefix loss.

#### A. Channel Estimation

The data rate in (4) depends on the IRS configuration  $\boldsymbol{\theta}$ . Acquiring channel state information (CSI) is key to selecting an appropriate configuration that is somehow matched to  $\mathbf{h}_d^f$  and  $\mathbf{V}^f$  in (3). Since the IRS is a passive device, in the sense that the  $N$  elements are not equipped with any active radio frequency (RF) chains, the channel estimation must be carried out at the base station or user, based on received pilot signals. Hence, it is hard (if not impossible) to separately estimate the channel from the base station to the IRS and from the IRS to the receiver. However, the cascaded channel  $\mathbf{V}^f$  from the base station through the IRS and to the receiver can be estimated using pilot signals sent from the base station to the receiver.

The acquisition of accurate CSI is essential to achieve high-performance gains in IRS-aided wireless systems. Many

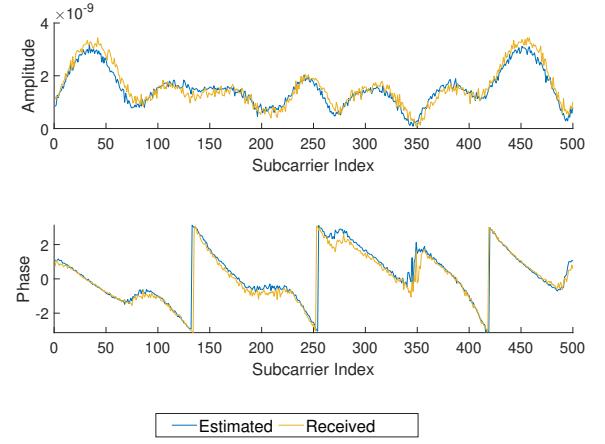


Fig. 2: Verification of our channel estimator. Although using only  $N$  configurations, it follows closely the amplitude and phase of the true received signal.

previous IRS configuration algorithms ignore the channel estimation problem and consider the achievable performance gains assuming perfect CSI for all channels considered [3]. Other works employ rudimentary least squares (LS) estimators [12]. Recent works employ more advanced channel estimation methods, such as IRS element grouping or exploitation of the common channel between the IRS and base station in a multi-user scenario [3]. In practice, assuming perfect CSI is unrealistic. Therefore, in this paper, we employ our novel LS channel estimation method which is described in detail in [12]. Briefly, we use only  $N$  constant pilot signals  $\mathbf{x}[k] \equiv \alpha$  transmitted with  $N$  different IRS configurations  $\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_N$  that form the columns of the  $N \times N$  Hadamard matrix. It is possible to estimate the channel using less than  $N+1$  pilot signals by utilizing information about the physical shape of the IRS. An example of channel estimation using this method is shown in Fig. 2.

#### B. IRS Configuration Selection

The selected IRS configuration  $\boldsymbol{\theta}$  affects all subcarriers because the transmissions are simultaneous. In the narrowband case, we can optimize the IRS in closed form for a single subcarrier, but in the wideband case, we must find a nontrivial tradeoff between all  $K$  subcarriers. The recent literature contains heuristic solutions based on successive convex approximations [6], semidefinite relaxation [7], and strongest tap maximization (STM) in the time domain [8].

Deep neural networks have shown remarkable results in many areas, such as signal and image processing and natural language processing, where it is usually difficult to formulate a mathematical model for feature representation. Similar tools have been recently proposed for wireless communications, where traditional approaches heavily rely on theory based on too simplified models. Deep learning methods for key IRS applications such as signal detection, channel estimation, and beamforming are reviewed in [9], [10]. Most of these methods

do not perform end-to-end learning, are environment-specific, and consume a significant amount of computational resources. In contrast, our proposed technique, as it is not pre-trained offline, can adapt to any environment. It converges quickly and therefore consumes relatively modest resources.

Our configuration optimization algorithm is based on the pioneering work of Deep Image Prior [13] that introduced the use of a *neural prior* in the context of image reconstruction, as well as on following works that extended the neural prior to other problems and representations. We observe that arbitrary IRS phase changes tend to reduce the data transmission rates, whereas configurations subject to a neural prior tend to perform better. This observation is supported by [5], [14], which claim that the IRS configuration should contain strong self-similarity due to spatial correlation. Moreover, it was shown in [15] that the over-parameterization, which is inherently available in neural networks, helps in avoiding local minima in optimization schemes. The design of our algorithm is inspired by these previous works, exploiting the neural prior and the over-parameterization that networks provide to find a configuration that maximizes the rate.

### III. DEEP NEURAL PRIOR

Deep convolution neural networks (CNNs) have been successfully used to learn features and mappings from large amounts of data, including priors about the data. Surprising recent work, Deep Image Prior [13], uses *untrained* CNNs to solve inverse problems like image restoration. In this scheme, a CNN generator is initialized with random weights. These weights are then optimized to make the network produce an output that is as close to the target image as possible. This procedure uses no prior information from other images as the structure of the generator network architecture captures a great deal of image statistics and is sufficient to impose a strong prior to restore the original image from the degraded image. The motivation behind using CNNs is that they share the same kernel/filter across the entire input image. This inductive bias encourages solutions that are highly preferable for image reconstruction, i.e., smooth, contain self-similarity, and lack noisy artifacts. The reconstruction error initially decreases and then plateaus as the network fits natural-looking images. Training should be stopped at this point before the error decreases further as the network starts fitting the noise.

Generators used for Deep Neural Prior are typically over-parameterized, i.e., the number of network weights is much larger than the output dimension. Choosing a neural network as the parametric model over-parameterizes the problem as typical neural networks have more parameters than required to reconstruct the target. This over-parameterization has been shown to be powerful at avoiding local minima [15] as well as crucial to allow smoother and Lipschitz limited solutions [16], [17].

In contrast to direct optimization techniques that usually perform some type of gradient-descent optimization on the model parameters directly, network-based optimization techniques try to minimize the objective by optimizing network

weights. Although a theoretical explanation has yet to be proposed, and with only initial mathematical interpretations, deep neural networks have been empirically shown to be helpful in solving various optimization problems and to reaching better solutions than state-of-the-art direct optimization algorithms. The idea of [13] has been extended to other fields, such as consolidation of point clouds [18], reconstruction of 3D meshes [19], and various audio applications [20]. Furthermore, deep neural networks have been shown to be successful at improving optimization processes in general, leading to better solutions than direct optimization [21], [22]. Therefore, assuming the solution lies on the manifold spanned by neural networks and employing a deep neural prior:

- 1) encourages a regularized result;
- 2) exploits the over-parameterization provided by neural networks to find a better local optimum.

Previous works [5], [14] claim that a favorable solution to the IRS configuration problem should contain strong self-similarity since the channel is spatially correlated over the IRS surface, supporting the use of a deep neural prior for IRS configuration search. Moreover, it was shown [15] that over-parameterization, which is inherently available in neural networks, helps to avoid local minima during the optimization process and reach a better final solution, even for simple problems such as linear kernel estimation. Consequently, we hypothesize that a deep neural prior may also be favorable for solving the IRS configuration problem as it is subject to a neural network that produces self-similar results and is able to avoid local optimization minima better than direct optimization.

We define our loss function as minus the data rate and optimize a network to overfit a phase configuration for a single specific user, so our optimization problem is defined as

$$\underset{\theta}{\text{minimize}} (-\mathcal{R}) \quad \text{s.t.} \quad |\theta| = 1, \quad (5)$$

where  $\mathcal{R}$  is defined in (4) and  $|\theta| = 1$  means that every entry should have a unit magnitude. This is a non-convex optimization problem because of its constraint. Therefore, convex optimization algorithms cannot be used to reach a global minimum and indeed, gradient-descent based algorithms have shown inferior results when compared to known heuristic solutions and also compared to our solution.

### IV. NETWORK ARCHITECTURE

Since the users are mainly distributed in the azimuth plane, we have observed that LOS users achieve high data rates with very smooth, column-like, configurations. We conclude that the configuration's azimuth components are highly important, especially for LOS users. Therefore, we designed a network architecture that accounts for the azimuth and elevation components separately and learns how to combine them in order to maximize the data rate. The separation into two branches facilitates the training of the network and allows it to converge faster. As depicted in Fig. 3, our network architecture consists of two branches with identical architecture: the top branch

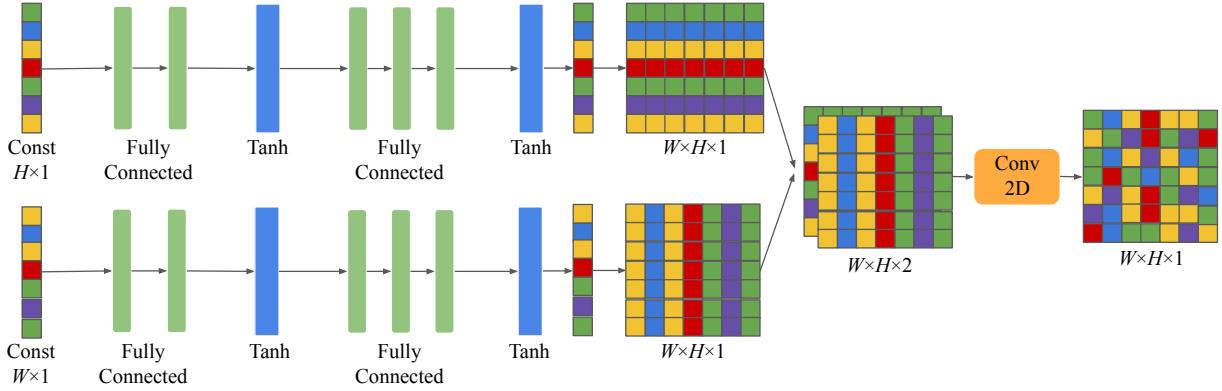


Fig. 3: The proposed network consists of two branches: the top branch generates the elevation component, and the bottom branch generates the azimuth component. The outputs of these two branches are combined to generate an IRS configuration matrix.

generates the elevation component and the bottom branch generates the azimuth component. The outputs of the two branches are used together to generate a final IRS configuration matrix. Each branch consists of five fully connected layers interspersed by one hyperbolic tangent activation function, and a final hyperbolic tangent activation layer. For optimization purposes, using multiple consecutive fully-connected layers is not equivalent to having a single larger fully-connected layer, as shown in [15]. In fact, this creates the desired over-parameterization by decomposing a single linear layer into equivalent multiple linear layers, which helps in avoiding local minima throughout the optimization.

The fully connected layers in the top branch are of size  $N_H \times 1$ . The resulting vector of this branch is of size  $N_H \times 1$  and represents the columns of an azimuth configuration. It is replicated to form a  $N_H \times N_W$  matrix. Similarly, the fully connected layers in the bottom branch are of size  $N_W \times 1$ . The resulting vector of this branch is of size  $N_W \times 1$  and represents the rows of an elevation configuration. It is also replicated to form a  $N_H \times N_W$  matrix. The outputs of the top and bottom branches are fed as separate input features into a convolutional layer that outputs a final IRS configuration matrix. An example of this combination process is depicted in Fig. 4. For optimizing the network, we used gradient descent with cosine annealing learning.

## V. RESULTS

We compare the performance of the proposed algorithm to the baseline case with no IRS, random IRS configurations, a gradient descent optimizer, and STM [8]. We use the Signal Processing Cup 2021 simulator [11] to create our validation and test sets. In the IRS simulator, we set the carrier frequency to  $f_c = 4$  GHz, the bandwidth to  $B = 10$  MHz, and the number of sub-carriers to  $K = 500$ . In our simulations, we assume the IRS is square-shaped, i.e.,  $N_H = N_W$ . Each element size is  $0.4\lambda \times 0.4\lambda$  where  $\lambda$  is the wavelength. To simulate typical physical conditions, we place the base station at a height of 25 m, away from the IRS and receivers, the IRS

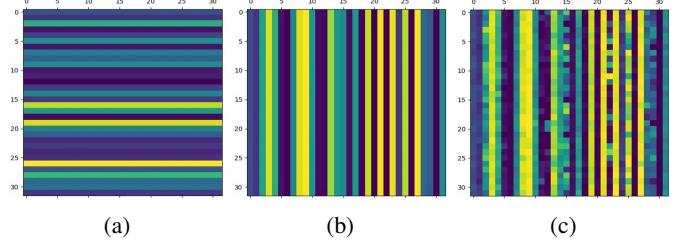


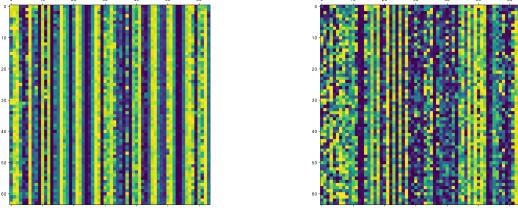
Fig. 4: Network outputs. A lighter color represents a higher phase value. (a) The output of the top branch is the elevation component, and (b) the output of the bottom branch is the azimuth component. They are combined (c) to form an IRS configuration matrix that tends to have a strong bias towards the azimuth component.

TABLE I: Average data rate obtained for 51 LOS and 51 NLOS users in an equal distance of 15 m from the center of the IRS.

Algorithm	LOS [Mbit/s]	NLOS [Mbit/s]
No IRS	3.48	3.49
Random phase	3.44	3.50
STM	111.16	69.45
Proposed	<b>115.87</b>	<b>70.98</b>

at a height of 10 m, and the users at varying heights ranging from 0 to 20 m, such that the base station, the IRS and the users are not co-planar. A quantitative comparison of the average data rate obtained for 51 users under the conditions described above is given in Table I. The data rate obtained using the proposed method is the highest, both for LOS users and for NLOS users. It should be noted that the small difference in values between LOS and NLOS in the first two rows of the table is due to randomness and is not statistically significant.

Fig. 5 depicts a typical configuration obtained by the proposed method and a typical configuration obtained by STM. The phases obtained by the proposed method are spatially smoother since the neural prior and the network architecture



(a) Proposed (b) STM

Fig. 5: Typical configurations obtained for one NLOS user with IRS of size  $64 \times 64$ . A lighter color represents higher phase values.

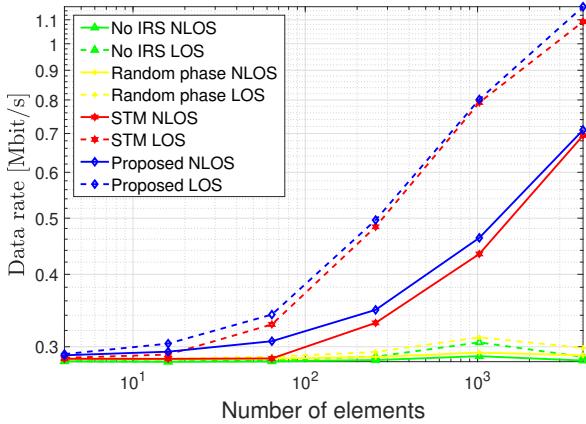


Fig. 6: Mean data rate for LOS and NLOS users as a function of the IRS size. The rate is averaged over 51 different users in equal distance from the center of the IRS.

encourage configurations with this property. Fig. 6 depicts the data rate as a function of the IRS size. As expected, using a well-configured IRS allows for higher data rates that also increase rapidly with the number of IRS elements. The proposed method achieves higher data rates compared to STM across all IRS sizes.

One of the notable features of our network is its ability to converge quickly. The number of optimization epochs until the network converges increases with the IRS size since the network size is  $5N_H + 5N_W + 9 \propto \sqrt{N}$ . We define the network convergence time as the number of epochs required to reach 95% of the rate obtained after  $10^5$  epochs. Fig. 7 shows the number of epochs required by the network to converge as a function of IRS size.

Another important advantage of our solution is its robustness to continuous changes in the channels. Such changes can occur as a result of a user moving around in the room, causing changes in the direct and controllable channels. Fig. 8 shows the number of epochs required by a pre-trained network to re-converge after a change in the channels caused by a shift in the receiver location. We simulated a user moving smoothly through the room with changing  $(x, y)$  coordinates but a constant  $z$  coordinate. We then measured the number of epochs required for the network to re-converge as a function

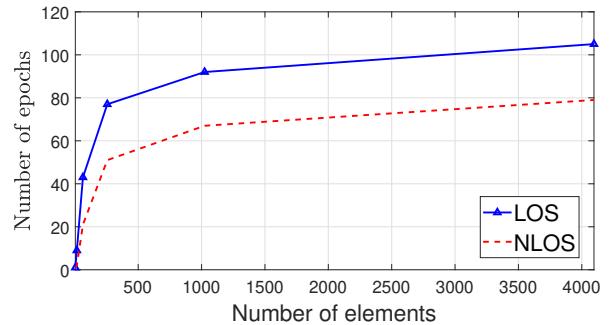


Fig. 7: Number of epochs required by the network to converge with a learning rate  $lr = \frac{10^{-2}}{N}$  as a function of IRS size.

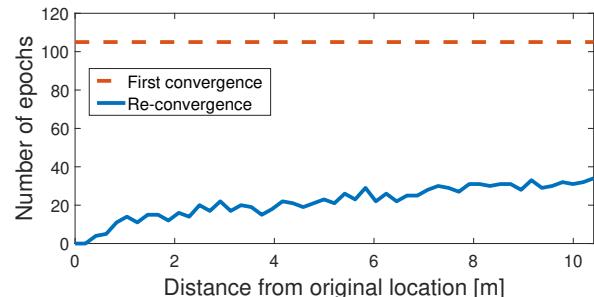


Fig. 8: Number of epochs required by a pre-trained network to re-converge to 99% of the new maximal data rate as a function of the user's distance from his/her original location with IRS of size  $64 \times 64$ . The result shown is an average for 10 users who start walking from different randomly selected starting points and walk in a straight line.

of the user's distance from his/her original location. The initial number of epochs was about 105, while the number of epochs required to re-converge was significantly lower. These results imply that the proposed network fits the system parameters rather than a specific channel or user. We also examined the influence of having a limited number of retraining epochs on the transmission rate of a user in motion. We measured the user's data transmission rate, and then we simulated his/her movement in a room. Fig. 9 shows the user's transmission rate after a limited number of retraining epochs as a fraction of the rate at the origin location. The results are compared to the case where the IRS retains the same configuration as in the original location.

As mentioned before, the STM method serves as a baseline for our comparisons. Therefore, it is important to compare the two algorithms not only in terms of the obtained transmission rate, but also in terms of their run times. Complexity metrics are not commonly used in deep learning since they are implementation specific. Instead, we measured the average run time of both methods on 51 users, 25 of whom were LOS and 26 of whom were NLOS. Both methods, STM and the proposed, were tested on a computer equipped with an Intel Core i7 CPU and NVIDIA RTX A5000 GPU, with our

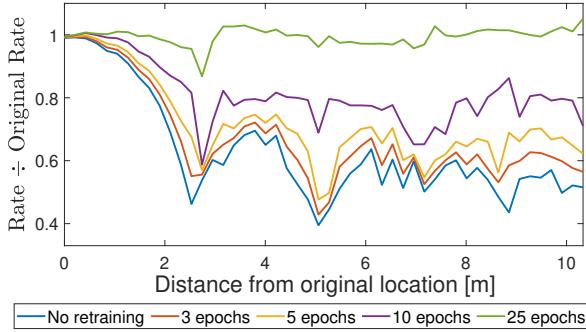


Fig. 9: User transmission rate as a function of the distance from the original location, with different limit values on the number of epochs for retraining. The rate is normalized by dividing by the transmission rate at the original location. It is possible to obtain values greater than 1 due to channel changes.

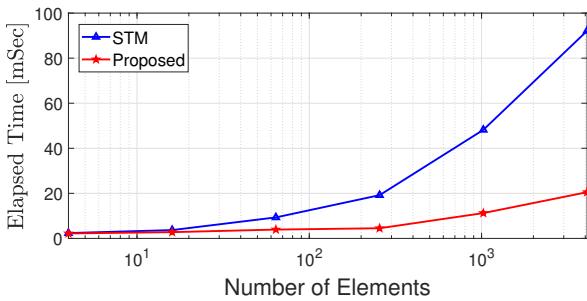


Fig. 10: Average run time for STM and the proposed method as a function of IRS size.

implementation exploiting the GPU’s parallel capabilities. The run times are depicted in Fig. 10. As shown in the figure, the proposed method is significantly faster, especially when the IRS contains a large number of elements.

## VI. CONCLUSIONS

We have addressed the IRS configuration optimization problem for wideband channels using a novel *deep neural prior* approach. We showed that the recently proposed ideas of neural prior and over-parametrization in generative neural networks can be applied to the world of wireless communications, far from their original use in computer vision. This approach, compared to the previous state-of-the-art algorithm, is significantly faster, and allows us to better utilize IRS technology to achieve higher data transmission rates between the base station and users. The code of our solution is available at <https://github.com/tofi98/DeepIRS>

## VII. ACKNOWLEDGMENT

The authors thank Prof. David Malah and Nimrod Peleg from the Signal and Image Processing Lab (SIPL) at the Technion for their continuous assistance and support.

## REFERENCES

- [1] M. Di Renzo, A. Zappone, M. Debbah, M.-S. Alouini, C. Yuen, J. de Rosny, and S. Tretyakov, “Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and road ahead,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 11, pp. 2450–2525, 2020.
- [2] E. Björnson, H. Wymeersch, B. Matthiesen, P. Popovski, L. Sangolini, and E. de Carvalho, “Reconfigurable intelligent surfaces: A signal processing perspective with wireless applications,” *arXiv preprint arXiv:2102.00742*, 2021.
- [3] Q. Wu, S. Zhang, B. Zheng, C. You, and R. Zhang, “Intelligent reflecting surface aided wireless communications: A tutorial,” *IEEE Transactions on Communications*, 2021.
- [4] J. Yuan, E. De Carvalho, R. J. Williams, E. Björnson, and P. Popovski, “Frequency-mixing intelligent reflecting surfaces for nonlinear wireless propagation,” *IEEE Wireless Communications Letters*, 2021.
- [5] E. Björnson, “Optimizing a binary intelligent reflecting surface for OFDM communications under mutual coupling,” in *ITG Workshop of Smart Antennas*, 2021.
- [6] Y. Yang, B. Zheng, S. Zhang, and R. Zhang, “Intelligent reflecting surface meets OFDM: Protocol design and rate maximization,” *IEEE Transactions on Communications*, vol. 68, no. 7, pp. 4522–4535, 2020.
- [7] B. Zheng and R. Zhang, “Intelligent reflecting surface-enhanced OFDM: Channel estimation and reflection optimization,” *IEEE Wireless Communications Letters*, vol. 9, no. 4, pp. 518–522, 2019.
- [8] S. Lin, B. Zheng, G. C. Alexandropoulos, M. Wen, F. Chen *et al.*, “Adaptive transmission for reconfigurable intelligent surface-assisted OFDM wireless communications,” *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2653–2665, 2020.
- [9] A. M. Elbir and K. V. Mishra, “A survey of deep learning architectures for intelligent reflecting surfaces,” *arXiv preprint arXiv:2009.02540*, 2020.
- [10] J. Wang, W. Tang, Y. Han, S. Jin, X. Li, C.-K. Wen, Q. Cheng, and T. J. Cui, “Interplay between RIS and AI in wireless communications: Fundamentals, architectures, applications, and open research problems,” *IEEE Journal on Selected Areas in Communications*, 2021.
- [11] E. Björnson, “IEEE Signal Processing Cup 2021 - Configuring an Intelligent Reflecting Surface for wireless communications,” 2021. [Online]. Available: [https://github.com/emilbjornson/SP\\_Cup\\_2021](https://github.com/emilbjornson/SP_Cup_2021)
- [12] T. Fireaizen, D. Ban-David, S. Hadad, G. Metzer, N. Kurland, S. Etkind, P. Lifshits, Y. Moshe, and I. Cohen, “Intelligent reflecting surface configuration using adaptive quantization and neural prior,” in *2021 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)*. IEEE, 2021.
- [13] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep image prior,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 9446–9454.
- [14] S. Lin, B. Zheng, G. C. Alexandropoulos, M. Wen, and F. Chen, “Progressive channel estimation and passive beamforming for ris-assisted ofdm systems,” in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 01–06.
- [15] S. Bell-Kligler, A. Shocher, and M. Irani, “Blind super-resolution kernel estimation using an internal-gan,” *arXiv preprint arXiv:1909.06581*, 2019.
- [16] S. Bubeck, Y. Li, and D. Nagaraj, “A law of robustness for two-layers neural networks,” *arXiv preprint arXiv:2009.14444*, 2020.
- [17] S. Bubeck and M. Sellke, “A universal law of robustness via isoperimetry,” *arXiv preprint arXiv:2105.12806*, 2021.
- [18] G. Metzer, R. Hanocka, R. Giryes, and D. Cohen-Or, “Self-sampling for neural point cloud consolidation,” *arXiv preprint arXiv:2008.06471*, 2020.
- [19] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or, “Point2mesh: A self-prior for deformable meshes,” *ACM Trans. Graph.*, vol. 39, no. 4, 2020.
- [20] Y. Tian, C. Xu, and D. Li, “Deep audio prior,” *arXiv preprint arXiv:1912.10292*, 2019.
- [21] S. Arora, N. Cohen, and E. Hazan, “On the optimization of deep networks: Implicit acceleration by overparameterization,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 244–253.
- [22] A. M. Saxe, J. L. McClelland, and S. Ganguli, “Exact solutions to the nonlinear dynamics of learning in deep linear neural networks,” *arXiv preprint arXiv:1312.6120*, 2013.