

# UC Irvine

## UC Irvine Previously Published Works

### Title

Optimization of Sparse Phased Arrays Using Deep Learning

### Permalink

<https://escholarship.org/uc/item/8986691x>

### Journal

2025 INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS (IJCNN), 00

### ISSN

2161-4393

### ISBN

979-8-3315-1043-5

### Authors

Earls, Jackson

Maman, Lior

Lu, David Lin Yi

et al.

### Publication Date

2025-07-05

### DOI

10.1109/ijcnn64981.2025.11229198

### Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

# Optimization of Sparse Phased Arrays Using Deep Learning

Jackson Earls

*Department of Computer Science  
University of California, Irvine  
Irvine, USA  
jearls@uci.edu*

Lior Maman

*Department of Physical Electronics  
Tel Aviv University  
Tel Aviv, Isreal  
maman@mail.tau.ac.il*

David Lin Yi Lu

*Department of Computer Science  
University of California, Irvine  
Irvine, USA  
lud19@uci.edu*

Amir Boag

*Department of Physical Electronics  
Tel Aviv University  
Tel Aviv, Isreal  
boag@tauex.tau.ac.il*

Pierre Baldi

*Department of Computer Science  
University of California, Irvine  
Irvine, USA  
pfbaldi@ics.uci.edu*

**Abstract**—Antenna arrays enhance communication, sensing, and imaging in fields like wireless communications, radar, and radio astronomy by improving signal quality and range through beamforming and spatial multiplexing. However, they face challenges such as mutual coupling, calibration issues, and increased complexity compared to single antennas. These issues require careful engineering to optimize array performance across applications. We present an optimization technique based on deep learning that improves the design of sparse phased arrays by minimizing grating lobes. This method starts with generating configurations for sparse antenna arrays, effectively tackling the non-convex challenges and the high degrees of freedom involved in array design. We utilize neural networks, trained on 70,000 and tested on 30,000 configurations, to estimate a non-convex cost function that evaluates the energy ratio between the main lobe and side lobe levels. This estimation is differentiable, facilitating the minimization of the cost function using gradient descent relative to the coordinates of the antenna elements, resulting in an optimized configuration. Additionally, we incorporate a custom penalty mechanism that integrates various physical and design constraints into the optimization process, enhancing its robustness and effectiveness in practical applications. The efficiency of our approach is resoundingly validated on the ten configurations with the lowest costs, achieving significant cost improvements ranging from 4.36 fold to 6.75 fold improvement, with a remarkable average improvement of 6.06 fold. The code for the method is available at <https://github.com/david5010/Optimization-of-Antenna-Arrays>.

**Index Terms**—Antenna Arrays, Main Lobe, Side Lobe Levels, Machine Learning, Deep Learning, Gradient Descent, Optimization

## I. INTRODUCTION

Antenna arrays consist of multiple antennas, typically arranged in a regular pattern, that work together as a single antenna to transmit or receive radio frequency signals. The configuration and interaction of the antennas in the array allow for improved performance in terms of directionality, gain, and the ability to form beams that can be electronically steered towards a specific direction without physically moving the

antennas. This makes antenna arrays useful in applications such as radar, wireless communications, and broadcasting.

A phased antenna array, also known as a phased array, is a type of antenna array where the relative phases of the signals fed to each antenna element are varied so that the effective radiation pattern of the array is reinforced in a desired direction and suppressed in undesired directions. This technique allows the beam of radio waves to be pointed in different directions without physically moving the antenna array.

The relevance of phased and sparse antenna arrays has grown significantly with the emergence of 6G networks, which aim to unify wireless communication and environmental sensing into a common platform known as Integrated Sensing and Communication (ISAC) [1]–[3]. In such systems, antenna arrays play a dual role: transmitting and receiving data while simultaneously enabling high-resolution sensing. Sparse arrays, in particular, are attractive for ISAC due to their reduced hardware requirements, improved energy efficiency, and potential for high spatial resolution [4]. However, realizing these benefits requires overcoming key design challenges, especially in optimizing sparse array configurations to suppress artifacts such as grating lobes and to maintain beamforming performance across wide scan ranges.

Phased array systems that use sparse configurations commonly face the challenge of grating lobes, a phenomenon associated with phased antenna arrays, and occur when there are multiple main lobes in the radiation pattern of the array. Grating lobes arise primarily due to the limited number of elements and the beamwidth requirements that determine the size of the antenna aperture. These lobes typically result from the periodic patterns seen in traditional antenna arrays [5]. To counter this, sparse arrays employ non-periodic layouts. Nonetheless, designing these aperiodic arrays introduces considerable challenges, including an increase in the degrees of freedom to be optimized and the complex computational demands needed to evaluate the antenna radiation patterns

across various main beam scan angles. Traditional optimization methods struggle to consistently identify global optima because of the complexity of the problem. The grid search method, though comprehensive, becomes unfeasible due to the sheer volume of possible configurations and the extensive computational effort required. Literature has proposed several methods for optimizing thinned and sub-array configurations using advanced techniques to refine antenna design [6]–[12]. Additionally, earlier methods aimed at sparse arrays have adopted stochastic optimization techniques, such as simulated annealing or genetic algorithms, to design aperiodic linear sparse arrays to minimize side lobe levels and optimize array configurations [13]. While these techniques are adaptable and capable of managing complex, non-linear optimization challenges, they tend to be computationally demanding and slow, particularly for optimizing large arrays.

In recent years, machine learning techniques have become increasingly popular for solving complex problems in the field of electromagnetics (EM). Specifically, neural networks have emerged as an efficient and timely optimization tool. These networks have shown considerable success in handling computationally demanding EM simulations [14], [15] and intricate EM relationships [16], [17] while preserving accuracy. Beamforming, a signal processing technique, has seen extensive application of deep learning [18]–[21]. While traditional beamforming methods suppress side lobe levels (SLLs) by adjusting the excitation of individual antenna elements, our approach preserves uniform excitation and instead optimizes the element positions. Unlike methods that input a desired array pattern into a neural network [21], our model accepts a given sparse array configuration and estimates its associated cost. This estimated cost is then used to iteratively refine the input geometry via gradient descent applied directly to the element coordinates with respect to the cost function.

We tackle the issues of computational speed and complexity in phased array scanning by initially reducing the problem's variables from five (including the observation vector, the scanning vector, and the wave number) to just two [22]–[25]. Subsequently, we utilize a sub-array method to generate sparse array data [26], which simplifies the optimization process by reducing the degrees of freedom. From this data, we identify array configurations that show potential and further refine these through deep learning techniques. By leveraging neural networks, we accurately estimate the complex, non-convex cost function associated with an array. The core novelty of our method lies in applying gradient descent directly to the antenna element coordinates, enabling effective minimization of this cost.

## II. METHODS

### A. Generation of Antenna Array Configurations

In this section, we address the design challenges of the Active Electronically Scanned Array (AESA) by simplifying the characterization of the AESA array factor (AF) from five variables to two and outline the data generation process for optimization. The array factor for a planar phased array

antenna, with uniformly excited elements situated in the  $yz$ -plane, is described as follows:

$$U(k, AZ, EL, AZ_0, EL_0) = \sum_{n=1}^{N_e} e^{jk\mathbf{r}_n \cdot (\hat{\mathbf{r}} - \hat{\mathbf{r}}_0)} \quad (1)$$

In this section,  $k$  denotes the wave-number, and  $\hat{\mathbf{r}} = \cos(EL) \sin(AZ) \hat{\mathbf{y}} + \sin(EL) \hat{\mathbf{z}} + \cos(EL) \cos(AZ) \hat{\mathbf{x}}$  is the unit vector pointing in the direction of observation. The vector  $\mathbf{r}_0 = \cos(EL_0) \sin(AZ_0) \hat{\mathbf{y}} + \sin(EL_0) \hat{\mathbf{z}} + \cos(EL_0) \cos(AZ_0) \hat{\mathbf{x}}$  signifies the primary beam direction. The position of the  $n$ th element is represented by  $\mathbf{r}_n = y_n \hat{\mathbf{y}} + z_n \hat{\mathbf{z}}$ . Here,  $EL$  and  $AZ$  are the elevation and azimuth in the observation direction, respectively, whereas  $EL_0$  and  $AZ_0$  refer to those in the main beam direction. It is noted that  $EL = \pi/2 - \theta$  and  $AZ = \varphi$  in the conventional spherical coordinates system. By introducing the difference vector  $\mathbf{u} = k(\mathbf{r} - \mathbf{r}_0)$ , the array factor can be redefined using just two variables:

$$U(u_y, u_z) = \sum_{n=1}^{N_e} e^{j(y_n u_y + z_n u_z)} \quad (2)$$

The components  $u_y$  and  $u_z$  represent the  $y$  and  $z$  components of the vector  $\mathbf{u}$ . The domain of  $\mathbf{u}$  can be established either numerically or analytically. Determining the support of the array factor within the  $u_y, u_z$  plane simplifies the assessment across the entire domain of interest. This approach decreases the demands on memory and processing resources, thereby facilitating a more straightforward characterization of the entire observation area throughout the full scan sector in a single representation.

To simplify the optimization process and manage the search domain more effectively, we implemented the concept of local periodicity to generate data [26]. This principle aids in organizing and integrating radiating elements along with their corresponding transmit/receive modules using a sub-array approach. In this strategy, the entire array is composed of multiple sub-arrays, each crafted to display periodicity, thereby streamlining the number of design variables. Practically, the array is arranged into periodic sub-arrays, where each can exhibit unique periodicity along both axes. Additionally, these sub-arrays are designed to be rotatable and repositionable within their specific areas. The array's surface is divided into different subdomains, each accommodating a periodic sub-array. Critical design parameters for optimization encompass the periodicity along each axis and the rotational and positional adjustments of each sub-array. This method reduces the complexity of array design by limiting the vast array of possible configurations to a structured set of variables. Fig. 1 illustrates an array configuration that employs a periodic sub-array structure.

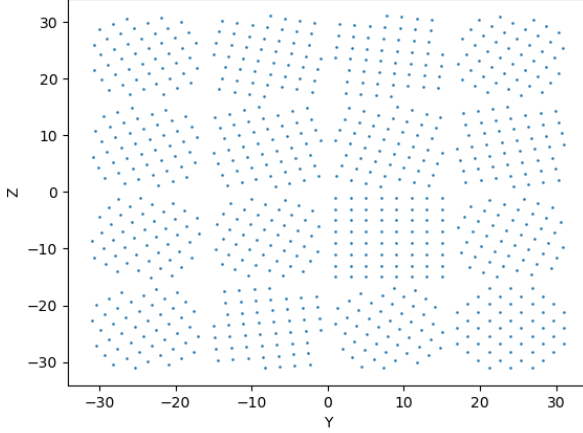


Fig. 1. Periodic sub-array configuration.

### B. Cost Function

The cost function assesses the ratio of the main lobe (ML) to the side lobe level (SLL), complying with the constraints

$$f_{cost} = - \frac{\sum_{i \in ML} |AF(u_y(i), u_z(i))|^{2p}}{\sum_{i \notin ML} |AF(u_y(i), u_z(i))|^{2p}} \quad (3)$$

where  $p$  specifies the norm type. In our data generation process, we choose  $p = 4$  to apply more severe penalties on pronounced peaks, which helps prevent the occurrence of excessively high SLLs, such as grating lobes, within the SLL region.

## III. OPTIMIZATION OF ANTENNA ARRAYS USING NEURAL NETWORK APPROXIMATIONS

Direct optimization of antenna array configurations poses a substantial challenge, especially when aiming to minimize a complex and potentially non-differentiable cost function. The intricacy of the actual cost function requires considerable engineering effort for direct optimization. To navigate these difficulties, we utilize neural networks as a strategic approach to accurately approximate the true cost function, thus acting as a surrogate cost function. Research has demonstrated that neural networks are effective approximation tools [27]–[30]. By their nature, neural networks produce a smooth and continuous function, a feature that facilitates the use of gradient descent methods for optimization purposes.

The selection of a surrogate function offers considerable flexibility, as long as it provides an accurate approximation of the original cost function. This surrogate could adopt various architectural forms, including Feedforward Neural Networks (FNNs), Convolutional Neural Networks (CNNs), or transformer variants, with the primary optimization process commencing once a reliable estimate is secured. In our experiments, we employed both an FNN and a Set Transformer [31] as surrogate functions.

### A. Model Selection

To determine the optimal model architecture and hyperparameters, we used SHERPA [32], a hyperparameter optimization software. We applied a grid search algorithm for each tunable parameter within both the Feedforward Neural Network (FNN) and Set Transformer models.

1) *Feedforward Neural Network (FNN) Model:* The Feedforward Neural Network (FNN) model is structured with four fully connected layers. The input layer has a dimension of 2048, corresponding to the Y and Z coordinates of 1024 array elements. The subsequent two hidden layers have dimensions of 20 and 12, respectively. ReLU activation functions are applied after each of these hidden layers to introduce non-linearity. The output layer consists of a single neuron that predicts the cost associated with the inputted antenna array geometry.

The FNN was trained over 1000 epochs using a learning rate of  $1 \times 10^{-5}$  and a batch size of 128. The Adam optimizer was employed to minimize the loss function during training.

We experimented with dimensions in the range of [4, 120] neurons for the first hidden layer and [2, 40] neurons for the second hidden layer. In addition, we tested learning rates ranging from  $1 \times 10^{-2}$  to  $1 \times 10^{-6}$  and batch sizes between 16 and 256 to determine the optimal parameters for the training process.

2) *Set Transformer:* The Set Transformer, designed to handle set-structured inputs as outlined by Lee et al. [31], includes an encoder and a decoder. The encoder features two permutation equivariant set attention blocks, which help maintain invariance to the order of inputs. The decoder is composed of a multi-head attention block used as a pooling mechanism, two set-attention blocks, and a fully connected output layer.

Each attention block in both the encoder and decoder utilizes two attention heads. The hidden layers within the network have a dimensionality of 32. The input to this model also has a dimension of 2, corresponding to the Y and Z coordinates of each antenna element. The output layer, similar to the FNN, consists of a single neuron that outputs the cost calculation for the given array geometry.

The Set Transformer was also trained over 1000 epochs but with a higher learning rate of  $1 \times 10^{-3}$ , using a batch size of 64 and the Adam optimizer.

We experimented with hidden layer dimensions of 8 to 64 and attention head counts of 2 to 8. Furthermore, we tested learning rates from  $1 \times 10^{-2}$  to  $1 \times 10^{-6}$  and batch sizes between 16 and 256.

### B. Inputs

The main difficulty in managing inputs stems from the specific neural network architecture utilized, particularly when the architecture is sensitive to the order and length of inputs. This sensitivity is notable in the Feedforward Neural Network (FNN) used in our study. To address this, we ensure a consistent order of antenna coordinates across all arrays and pad each input to match the maximum antenna count in the

dataset, which is 1024. This approach effectively handles the intricacies of input variations.

In contrast, the Set Transformer model is specifically designed to manage inputs of varying order and length without such adjustments. The inherent permutation invariance of the Set Transformer was a substantial factor in its selection for our experiments. This model naturally accommodates the 2D structure of element coordinates  $[(y_1, z_1), (y_2, z_2), \dots, (y_n, z_n)]$ , eliminating the need for padding or a predetermined sequence. The architecture of the Set Transformer is fundamentally based on receiving sets as input, recognizing that the order of elements in a set does not affect its inherent properties or the input length.

The Set Transformer achieves enhanced performance by utilizing specialized components such as Pooling Blocks and Attention Blocks. Pooling Blocks aggregate information across the entire set, encoding it into a fixed-size representation. This aggregation process enables the model to pool relevant features from the entire array, irrespective of its size, ensuring comprehensive feature representation. Subsequently, the encoded information is processed by the Attention Blocks, which leverage the self-attention mechanism. These blocks are crucial as they enable the model to capture both pairwise and higher-order interactions among elements.

### C. Model Cost

To determine the cost associated with different antenna array setups, a neural network is employed to produce a real number output, which can vary between -40,000 and -0.7. A lower numerical output from the network suggests a more optimal configuration cost-wise. Given the broad spectrum of these output values, normalization was necessary during the training phase. Specifically, with a calculated mean ( $\mu_y$ ) of -10,560 and a standard deviation ( $\sigma_y$ ) of 9,301, we utilized (4) for scaling the data. To make the neural network's predictions more understandable, we convert the scaled outputs back to their original cost values through reverse scaling.

$$Y_{\text{scaled}} = \frac{Y - \mu_y}{\sigma_y} \quad (4)$$

$$Y = (Y_{\text{scaled}} \cdot \sigma_y) + \mu_y \quad (5)$$

Since the dataset was generated using consistent parameters, such as the same wavelength and antenna aperture size, the cost of each configuration is purely determined by the coordinates of the array elements. Each array configuration is represented by a set of coordinate pairs, and the number of elements in each array can vary.

Once the neural network provides a reliable estimate of the cost function, it becomes the basis for optimizing antenna configurations. By structuring the configurations in a specific input format, we enable the application of gradient-based optimization techniques to fine-tune the positions of the antennas. This structured approach is crucial for effectively applying optimization methods that rely on gradients to iteratively improve the antenna setup.

To refine a specific antenna array configuration, we modify the positions of the array elements to achieve minimal values of the cost function. Utilizing a neural network surrogate model within a PyTorch framework [33], [34], we apply gradient-based optimization methods such as Stochastic Gradient Descent (SGD) or its variants. In our experiments, we selected the Adam Optimizer due to its computational efficiency and adjustable hyperparameters [35]. Over a predetermined number of iterations, we calculate the gradient of the surrogate model concerning the element coordinates and iteratively adjust these positions to secure a lower estimated cost. This process allows for precise optimization of antenna configurations based on the guidance provided by the surrogate model.

To enhance the optimization process, we implement mechanisms that uphold physical constraints among the array elements. One such mechanism is a minimum distance check. At each iteration, we assess each candidate configuration to verify that all elements maintain a predefined minimum distance from each other. If the configuration satisfies this distance constraint, it is approved, and the optimization progresses to the subsequent iteration. Conversely, if the configuration fails to meet the constraint, it reverts to the previous valid arrangement, and the optimization process is terminated. This approach ensures that the physical limitations of the array elements are consistently respected throughout the optimization.

The second method to enhance the optimization process includes incorporating a repulsion term into the cost model as a penalty function. This term applies a penalty based on the closeness of element pairs, facilitating uninterrupted optimization. The penalty is designed to escalate sharply as any pair of elements nears the minimum distance threshold, effectively deterring configurations that breach the required spacing. Structured to asymptotically approach infinity as distances approach the threshold, this penalty function can also be tailored to prevent other undesirable behaviors, such as elements excessively spreading apart or converging too closely. This ensures that all configurations adhere to acceptable spatial limits throughout the optimization process.

### D. Penalty Mechanism

In our model, the penalty mechanism is flexible, allowing for the application of customized penalties that align with specific design requirements. These penalties can be tailored to address a variety of constraints or optimization goals. For instance, the penalty function we utilize in our model is designed to adapt to different scenarios by incorporating specific penalties that are crucial for achieving the desired outcomes in antenna array configurations.

This function computes the Euclidean pairwise distance between each pair of elements, represented as  $D_{ij}$  for elements  $i$  and  $j$ , where  $i$  and  $j$  range from 1 to  $N_e$ . Additionally, a minimum threshold is defined as  $\theta$ .

$$P_{ij} = \log \left( \frac{1}{D_{ij} - \theta} \right) \quad (6)$$

The total penalty  $P$  applied during the optimization is calculated as the sum of all individual penalties, each scaled by a factor  $\epsilon$ . This scaling factor  $\epsilon$  determines the influence of the repulsion term on the optimization process, ensuring that the penalties effectively discourage undesirable configurations by adjusting their impact according to specific optimization needs.

$$P = \epsilon \cdot \sum_{\substack{\forall i \in \text{Ant} \\ \forall j \in \text{Ant}}} P_{ij} \quad (7)$$

In this context,  $\text{Ant}$  denotes the set of elements within the configurations. The scalar  $\epsilon$  plays a critical role, as a higher value significantly emphasizes the importance of maintaining distances between elements, effectively preventing them from approaching the minimum threshold too closely. This adjustment ensures that the configuration adheres to the desired spatial constraints, optimizing the overall system performance.

The optimization process is focused on minimizing the custom loss function, which is defined as:

$$\text{loss} = f(\text{Ant}) + P \quad (8)$$

In this equation,  $f(\text{Ant})$  represents the output from the surrogate cost network for a specific antenna configuration, indicating the estimated cost.  $P$  denotes the custom penalty term incorporated to enforce certain constraints and ensure the design adheres to specified requirements.

This approach allows for a systematic optimization of antenna configurations by balancing the estimated costs with penalties for any deviations from desired constraints.

#### IV. RESULTS

In this section, we explore the performance improvements in antenna arrays achieved through the use of the Feedforward Neural Network (FNN) and the Set Transformer, both with the implementation of a penalty mechanism designed to minimize mutual coupling. Fig. 2 displays the optimized sparse array coordinates obtained by the FNN that employs the penalty mechanism. Table 1 shows the results for both the FNN and the Set Transformer without the penalty mechanism. In contrast, Table 2 shows the results when the penalty mechanism is used in conjunction with the FNN and Set Transformer models.

For clarity and simplicity in demonstrating the effectiveness of our methods, we chose the ten antenna array configurations that exhibited the lowest initial costs. Our method is designed to be scalable; multiple optimization processes can be run in parallel to handle larger datasets or more complex configurations efficiently. This scalability ensures that our optimization approach can be adapted to various practical applications and larger-scale problems.

The FNN architecture demonstrates superior performance in optimizing antenna configurations. One effective method to reduce mutual coupling involves increasing the minimum distance between the nearest elements. When the penalty mechanism is not implemented in the array optimization process, the average minimum distance is  $0.5011\lambda$ . However, when the penalty mechanism is implemented, the average distance expands to  $0.5087\lambda$ .

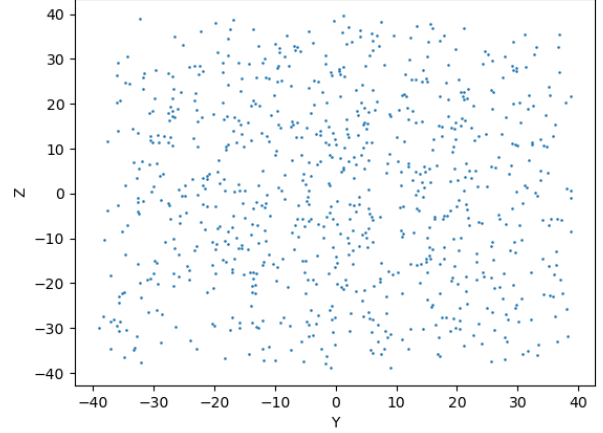


Fig. 2. Example of an optimized array configuration.

##### A. Impact of Gradient Descent without Penalty

Examining the 10 most economical configurations without applying the custom penalty showed an average cost reduction of 62%, with the highest reduction reaching 76%. Further analysis of the smallest distances between elements after optimization indicates that these distances frequently neared the threshold value ( $\theta = 0.5$ ). This tendency to approach the threshold implies that while the optimization aims to lower costs, it risks the physical feasibility and operational integrity of the antenna arrays by pushing the configurations to the boundaries of the minimum distance constraint.

##### B. Effectiveness of the Penalty Function

The use of gradient descent, coupled with a penalty for adhering to the minimum distance constraint, has shown significant improvements in reducing the costs of antenna array configurations. By incorporating our custom penalty mechanism, we noted an average enhancement of 606% across the examined configurations, with the most substantial improvement recorded at 675% for the FNN model. These findings

TABLE I  
COMPARISON OF COST BY OPTIMIZATION WITHOUT PENALTY

| Config. | Cost Bef. | FNN       |         | SetTrans.  |          |
|---------|-----------|-----------|---------|------------|----------|
|         |           | Cost Aft. | %Chg    | Cost Aft.  | %Chg     |
| 0       | -40463.98 | -60435.75 | -49.35% | -55972.50  | -38.32%  |
| 1       | -39931.30 | -67853.06 | -69.92% | -54156.19  | -35.62%  |
| 2       | -39767.27 | -65105.89 | -63.71% | -65431.58  | -64.53%  |
| 3       | -39472.41 | -66652.71 | -68.85% | -58543.00  | -48.31%  |
| 4       | -39431.64 | -69775.05 | -76.95% | -83596.91  | -112.00% |
| 5       | -39383.22 | -64393.63 | -63.50% | -57158.75  | -45.13%  |
| 6       | -39036.94 | -60825.42 | -55.81% | -56469.61  | -44.65%  |
| 7       | -38702.13 | -60464.29 | -56.22% | -91310.48  | -135.93% |
| 8       | -38690.57 | -63633.52 | -64.46% | -101049.47 | -161.17% |
| 9       | -38600.70 | -60984.63 | -57.98% | -70421.51  | -82.43%  |

highlight the capability of gradient descent to significantly refine and enhance the performance of antenna array designs. Incorporating a penalty function into the optimization process

TABLE II  
COMPARISON OF COST BY OPTIMIZATION WITH PENALTY

| Config. | Cost Bef. | FNN        |          | SetTrans. |          |
|---------|-----------|------------|----------|-----------|----------|
|         |           | Cost Aft.  | %Chg     | Cost Aft. | %Chg     |
| 0       | -40463.98 | -217088.64 | -436.49% | 100749.35 | -148.98% |
| 1       | -39931.30 | -274389.90 | -587.15% | -56144.42 | -40.60%  |
| 2       | -39767.27 | -286250.78 | -619.81% | -78089.15 | -96.36%  |
| 3       | -39472.41 | -278903.59 | -606.57% | -76152.99 | -92.92%  |
| 4       | -39431.64 | -258909.26 | -556.60% | -86334.79 | -118.94% |
| 5       | -39383.22 | -303772.37 | -671.32% | -65641.45 | -66.67%  |
| 6       | -39036.94 | -290405.96 | -643.92% | -98108.46 | -151.32% |
| 7       | -38702.13 | -288678.65 | -645.89% | -87371.46 | -125.75% |
| 8       | -38690.57 | -278607.12 | -620.09% | -91047.03 | -135.32% |
| 9       | -38600.70 | -299506.68 | -675.91% | -85766.29 | -122.18% |

helps maintain operational feasibility while maximizing cost efficiency. This approach ensures that enhancements in antenna array configurations do not sacrifice their practical utility. Additionally, the ability to adjust the  $\epsilon$  value introduces a level of precise control provided by the penalty function over the optimization process. This precise control facilitates a customized balance between aggressive cost reduction and compliance with physical constraints. Optimal results were achieved with  $\epsilon$  values of 12.5 for the FNN architecture and 1 for the Set Transformer architecture, determined after testing  $\epsilon$  values ranging from 0.1 to 100.

### C. Comparative Analysis

When comparing the outcomes of optimization processes, the importance of our surrogate cost function becomes evident, particularly in understanding the relationship between cost reduction and the minimum distance between elements. The surrogate cost function plays a crucial role in guiding the optimization process to achieve the desired trade-offs between reducing costs and maintaining practical distances between elements.

The surrogate cost function effectively quantifies the trade-offs involved, allowing for a more informed decision-making process that balances cost efficiency with operational feasibility. This relationship highlights the critical role that mathematical modeling and optimization techniques play in the refinement of complex systems such as antenna arrays.

The optimization approach that incorporates the penalty function maintains a safer distance from the threshold, indicating a more cautious strategy that prioritizes compliance with physical constraints. This method ensures that antenna configurations remain within acceptable bounds, enhancing the practical viability of the optimized antenna arrays.

Optimization of the array using the penalty function not only kept a safer distance from the threshold but also outperformed the optimization without the penalty. This enhancement was evident in both the FNN and Set Transformer models. By ensuring that the configurations stay within acceptable limits, this method of optimization improves both the efficiency and the practicality of the optimized antenna arrays.

This comparative study highlights the significance of adhering to crucial design constraints. Integrating a custom

penalty term into the optimization process was markedly more effective than optimization without such a term. The use of this custom penalty term led to both larger cost reductions and increased spacing between array elements.

Employing a penalty function becomes essential when maintaining physical and operational integrity is crucial, especially in sensitive applications. This method not only reduces costs but also heightens pragmatic feasibility. The scaling term within the penalty function provides considerable flexibility. When aggressive cost reduction is the priority, the scaling term can be adjusted to optimize costs for a specific array geometry. Conversely, when adherence to physical constraints is paramount, increasing the scaling term can promote greater spacing between array elements, albeit at the potential expense of performance.

Understanding these trade-offs is crucial for engineers and designers as they select the most appropriate optimization strategy. Each method has its advantages and limitations, which must be carefully weighed against the project's specific goals and constraints.

## V. CONCLUSION

A novel design methodology has been developed for antenna arrays that utilizes deep learning to efficiently optimize large antenna configurations. This strategy begins with the creation of sub-arrays, which serve as the foundational elements for configuring the complete array. At the heart of this methodology is the use of deep learning, using neural networks to facilitate further refinements. This deep learning method has proven to be an effective means of optimizing large antenna arrays, significantly reducing costs, and providing the flexibility to manage both design and physical constraints. Applying deep learning, we achieved an average cost reduction of 606% with penalty and 62% without penalty in the ten lowest cost antenna array configurations. This innovative approach, which employs neural networks to both help estimate and minimize the cost function, holds promise for future applications in various areas, including phase tapering, beam shaping methods, and thinned array design, among others.

## REFERENCES

- [1] H. Luo, T. Zhang, C. Zhao, Y. Wang, B. Lin, Y. Jiang, D. Luo, and F. Gao, "Integrated sensing and communications framework for 6g networks," *arXiv preprint arXiv:2405.19925*, 2024.
- [2] T. Wild, A. Grudnitsky, S. Mandelli, M. Henninger, J. Guan, and F. Schaich, "6g integrated sensing and communication: From vision to realization," *arXiv preprint arXiv:2305.01978*, 2023.
- [3] S. Lu, F. Liu, Y. Li, K. Zhang, H. Huang, J. Zou, X. Li, Y. Dong, F. Dong, J. Zhu, Y. Xiong, W. Yuan, Y. Cui, and L. Hanzo, "Integrated sensing and communications: Recent advances and ten open challenges," *IEEE Internet of Things Journal*, vol. 11, no. 11, pp. 19094–19120, 2024.
- [4] J. Li, X. Shao, F. Chen, S. Wan, C. Liu, Z. Wei, and D. Wing Kwan Ng, "Networked integrated sensing and communications for 6g wireless systems," *IEEE Internet of Things Journal*, vol. 11, no. 17, pp. 29062–29075, 2024.
- [5] N. Amitay, V. Galindo, and C. P. Wu, *Theory and Analysis of Phased Array Antennas*. New York: Wiley-Interscience, 1972.
- [6] P. Rocca, G. Oliveri, R. J. Mailloux, and A. Massa, "Unconventional phased array architectures and design methodologies—a review," *Proceedings of the IEEE*, vol. 104, no. 3, pp. 544–560, 2016.

- [7] P. Rocca, R. Mailloux, and G. Toso, "Ga-based optimization of irregular subarray layouts for wideband phased arrays design," *IEEE Antennas and Wireless Propagation Letters*, vol. 14, pp. 131–134, 2014.
- [8] P. Rocca, R. L. Haupt, and A. Massa, "Sidelobe reduction through element phase control in uniform subarrayed array antennas," *IEEE Antennas and Wireless Propagation Letters*, vol. 8, pp. 437–440, 2009.
- [9] P. Rocca, L. Manica, and A. Massa, "An improved excitation matching method based on an ant colony optimization for suboptimal-free clustering in sum-difference compromise synthesis," *IEEE Transactions on Antennas and Propagation*, vol. 57, no. 8, pp. 2297–2306, 2009.
- [10] R. L. Haupt, *Antenna arrays: a computational approach*. John Wiley & Sons, 2010.
- [11] R. L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 7, pp. 993–999, 1994.
- [12] R. L. Haupt, "Interleaved thinned linear arrays," *IEEE Transactions on Antennas and Propagation*, vol. 53, no. 9, pp. 2858–2864, 2005.
- [13] A. Trucco and V. Murino, "Stochastic optimization of linear sparse arrays," *IEEE Journal of Oceanic Engineering*, vol. 24, no. 3, pp. 291–299, 1999.
- [14] Y. Kim, S. Keely, J. Ghosh, and H. Ling, "Application of artificial neural networks to broadband antenna design based on a parametric frequency model," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 669–674, 2007.
- [15] X. Ding, V. Devabhaktuni, B. Chattaraj, M. Yagoub, M. Deo, J. Xu, and Q. J. Zhang, "Neural-network approaches to electromagnetic-based modeling of passive components and their applications to high-frequency and high-speed nonlinear circuit optimization," *IEEE Transactions on Microwave Theory and Techniques*, vol. 52, no. 1, pp. 436–449, 2004.
- [16] K.-C. Lee, "Application of neural network and its extension of derivative to scattering from a nonlinearly loaded antenna," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 990–993, 2007.
- [17] J. Liu, Z.-X. Chen, W.-H. Dong, X. Wang, J. Shi, H.-L. Teng, X.-W. Dai, S. S. T. Yau, C.-H. Liang, and P.-F. Feng, "Microwave integrated circuits design with relational induction neural network," *arXiv preprint arXiv:1901.02069*, 2019.
- [18] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1065–1069, 2020.
- [19] T. Lin and Y. Zhu, "Beamforming design for large-scale antenna arrays using deep learning," *IEEE Wireless Communications Letters*, vol. 9, p. 103–107, Jan. 2020.
- [20] S. Komeylian, "Deep neural network modeling of different antenna arrays; analysis, evaluation, and application," *IEEE Canadian Journal of Electrical and Computer Engineering*, vol. 44, no. 3, pp. 261–274, 2021.
- [21] X. Yang, D. Yang, Y. Zhao, J. Pan, and Y. Chen, "Synthesis of linear sparse array using dnn-based machine-learning method," *IEEE Transactions on Antennas and Propagation*, vol. 71, no. 8, pp. 6513–6522, 2023.
- [22] L. Maman, S. Zach, and A. Boag, "Phase-only beam broadening for array antennas," in *2023 17th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–3, IEEE, 2023.
- [23] L. Maman, S. Zach, and A. Boag, "Beam broadening by phase tapering in wideband electronically scanned arrays," in *2021 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)*, (Tel Aviv, Israel), Nov. 2021.
- [24] L. Maman, S. Zach, and A. Boag, "Beam broadening by phase tapering in wideband electronically scanned arrays," in *2023 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)*, (Tel Aviv, Israel), Jul. 2024.
- [25] L. Maman, S. Zach, and A. Boag, "Beam shaping by phase-only wave-form encoding for transmitting array antennas in radar applications," *IEEE Open Journal of Antennas and Propagation, Early Access*, 2025.
- [26] A. Boag, S. Zach, Y. Gazit, G. Zwirn, and A. Boag, "Grating lobe mitigation in ultra-wideband phased arrays," in *2018 International Conference on Electromagnetics in Advanced Applications (ICEAA)*, pp. 85–86, IEEE, 2018.
- [27] P. Baldi, *Deep learning in science*. Cambridge University Press, 2021.
- [28] F. Scarselli and A. C. Tsoi, "Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results," *Neural networks*, vol. 11, no. 1, pp. 15–37, 1998.
- [29] Y. Patel, "Neural network training and non-differentiable objective functions," *arXiv preprint arXiv:2305.02024*, 2023.
- [30] R. DeVore, B. Hanin, and G. Petrova, "Neural network approximation," *Acta Numerica*, vol. 30, p. 327–444, 2021.
- [31] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh, "Set transformer: A framework for attention-based permutation-invariant neural networks," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 3744–3753, PMLR, 09–15 Jun 2019.
- [32] L. Hertel, J. Collado, P. Sadowski, J. Ott, and P. Baldi, "Sherpa: Robust hyperparameter optimization for machine learning," *SoftwareX*, 2020. In press.
- [33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.
- [34] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Workshop on Machine Learning Systems*, 2017.
- [35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.