2023 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 17–20, 2023, ROME, ITALY

GFLOWNETS FOR SENSOR SELECTION

Spilios Evmorfos[†] *Zhaoyi Xu*[†] *Athina Petropulu*[†]

[†]Rutgers, The State University of New Jersey

ABSTRACT

The efficacy of sensor arrays improves with more elements, yet increased number of elements leads to higher computational demands, cost and power consumption. Sparse arrays offer a cost-effective solution by utilizing only a subset of available elements. Each subset has a different effect on the performance properties of the array. This paper presents an unsupervised learning approach for sensor selection based on a deep generative modeling. The selection process is treated as a deterministic Markov Decision Process, where sensor subarrays arise as terminal states. The Generative Flow Network (GFlowNet) paradigm is employed to learn a distribution over actions based on the current state. Sampling from the aforementioned distribution ensures that the cumulative probability of reaching a terminal state is proportional to the sensing performance of the corresponding subset. The approach is applied for transmit beamforming where the performance of a subset is inversely proportional to the error between its corresponding beampattern and a desired beampattern. The method can generate multiple high-performing subsets by being trained on a small percentage of the possible subsets (less than 0.0001% of the possible subsets for the conducted experiments).

***Index Terms*—** sensor selection, GFlowNets, deep learning, deep generative modeling

1. INTRODUCTION

The size and number of elements in a sensor array greatly affect its performance. However, larger arrays result in increased costs, computational demands, and power consumption. Sparse arrays offer a cost-effective solution by utilizing only a subset of available elements. Since each array element has a different effect on the array beampattern, the main challenge lies in developing efficient methods to select the optimal subset for performance optimization.

Several analytical methods have been proposed for designing sparse sensor arrays. For example, in [1], a greedy algorithm is introduced. The algorithm selects a subarray that maximizes the mutual information between the collected measurements and the far-field array pattern for cognitive radio systems. Another approach, presented in [2], focuses on a Multiple-Input-Multiple-Output (MIMO) radar. It introduces a semidefinite programming method to choose a Tx-Rx pair that maximizes the separation between desired and undesired directions of arrival. For massive MIMO systems, convex optimization frameworks have been proposed in [3, 4]. Furthermore, [5] explores a distributed multiple-radar scenario and presents a greedy approach to sensor selection where the Cramér-Rao Bound (CRB) of the target estimates is optimized. The work in [6] proposes a kernel approach for sensor selection in the context of decentralized detection.

Machine learning approaches have also been considered. Support Vector Machines for sparse antenna array design in wireless communications is proposed in [7]. Similarly, in [8], the antenna selection problem is cast as a classification problem and a Convolutional Neural Network is proposed as a classifier. In [9], a supervised deep learning method is proposed to learn the optimal antenna selection and precoding matrix. The work of [10] proposes a supervised deep learning approach to select antennas based on a channel state information extrapolation metric for massive MIMO systems. In [11], antenna selection in MIMO transmit beamforming is considered, using a classifier that is a neural network composed of elementary operations that resemble self-attention. A similar approach is proposed in [12] for sensor selection in Dual-Function Radar and Communications systems.

The limitation of analytical methods is that they are criterion-dependent. They rely on specific properties of the particular selection objective in order to employ convex relaxations or greedy selection. Additionally, such approaches are prone to getting trapped in local optima. On the other hand, supervised learning methods which treat selection as a multilabel classification problem necessitate extensive apriori annotation and data preprocessing which is challenging to obtain for dynamic sensor selection problems. The current paper proposes a general framework for sensor selection that explicitly addresses the limitations of prior research. The proposed method is criterion-agnostic, eliminating the need for specific assumptions regarding the selection objective (e.g., convexity, differentiability, etc.), and is unsupervised, thereby removing the reliance on annotation.

The novelty of the proposed approach lies in the modeling of the selection process as a deterministic Markov Decision Process (MDP) with a unique root, as opposed to for-

Work supported by ARO under Grants W911NF2110071 and W911NF2320103.

979-8-3503-2411-2/23/$31.00 ©2023 IEEE

mulating it as a multiclass classification problem. By doing so, the selection of each sensor subarray corresponds to an MDP trajectory that leads to a specific terminal state. The reward assigned to each non-terminal state is zero, while the reward of each terminal state is determined by evaluating the selection objective for the corresponding sensor subset. By adopting such formulation, the goal becomes to learn a distribution over actions, conditioned on the state, such that the cumulative probability of arriving at a terminal state is proportional to the state's reward. Although Markov Chain Monte Carlo (MCMC) [13] sampling methods can be utilized for this purpose, they primarily explore locally, necessitating a substantial number of transitions to converge. Here, we propose the use of Generative Flow Networks (GFlowNets) [14] framework, a recently proposed deep generative modeling approach, to learn the distribution over actions for the sensor selection MDP. GFlowNets introduce a flow quantity to amortize the learning cost of distributions over composite objects. The approach converts flow-matching equations into a learning objective, shifting the distribution learning problem from the combinatorial object space to the continuous parameter space of a function approximator. Stochastic gradient descent methods, known for their generalization capabilities to unseen examples, can be utilized. GFlowNet employs a neural network to parameterize the flow of the MDP. However, for the sensor selection MDP, the inputs to the flow network consist of binary vectors. Recent learning theory results highlight the limited effectiveness of dense neural networks in handling binary vector inputs, known as Spectral Bias [15]. To overcome this issue, we propose replacing the input layer of the flow network with a Fourier features preprocessing kernel [16]. This modification improves training speed by a factor of 5.

The proposed approach is applied to the problem of antenna selection for MIMO transmit beamforming, where the aim is to select the subarray whose output best matches a predefined beampattern. Despite being trained on a small fraction of the possible subarrays (less than 0.0001% of the total), the proposed method demonstrates superior performance compared to Metropolis-Hastings MCMC and compared to the greedy approach of [5].

2. GENERATIVE FLOW NETWORKS

Consider an MDP, where $\mathcal{S}$ is the set of states and $\mathcal{X} \subset \mathcal{S}$ is the set of terminal states. Let $\mathcal{A}$ be the set of discrete actions and $\mathcal{A}(\mathbf{s})$ the set of permissible actions at state $\mathbf{s}$. The MDP is a Directed Acyclic Graph (DAG), where the leaf nodes possess positive reward, all intermediate states possess zero reward ($\mathcal{R}(\mathbf{s}) = 0 \; \forall \mathbf{s} \notin \mathcal{X}$) and there is a unique root $\mathbf{s}_0$. The DAG is non-injective, implying that different action sequences (starting from the root) can lead to the same state. The objective is to learn an action selection policy, such that the probability of reaching a terminal state is proportionate to the state's reward.

The GFlowNet views the MDP as a flow network. The flow quantity stems from the root and there is a flow sink at each leaf node. Assuming that action $\mathbf{a}$ is performed at state $\mathbf{s}$, the next state $\mathbf{s}'$ can be denoted as $T(\mathbf{s}, \mathbf{a}) = \mathbf{s}'$. The MDP is deterministic, therefore, $T(\mathbf{s}, \mathbf{a})$ is unique for every pair. The flow of edge $(\mathbf{s}, \mathbf{a})$ is denoted as $F(\mathbf{s}, \mathbf{a})$, and the total flow going through state $\mathbf{s}$ as $F(\mathbf{s})$. In order to satisfy the flow balance conditions, the incoming flow of each state should match the outgoing flow. For any node $\mathbf{s}'$, the in-flow is:

$$F(\mathbf{s}') = \sum_{\mathbf{s},\mathbf{a}:T(\mathbf{s},\mathbf{a})=\mathbf{s}'} F(\mathbf{s}, \mathbf{a})$$

On the other hand, the out-flow can be defined as

$$F(\mathbf{s}') = \sum_{\mathbf{a}' \in \mathcal{A}(\mathbf{s}')} F(\mathbf{s}', \mathbf{a}')$$

With $R(\mathbf{s}) = 0$ for non terminal nodes and $\mathcal{A}(\mathbf{s}) = \emptyset$ for leaf nodes, the flow balance equations are:

$$\sum_{\mathbf{s},\mathbf{a}:T(\mathbf{s},\mathbf{a})=\mathbf{s}'} F(\mathbf{s}, \mathbf{a}) = R(\mathbf{s}') + \sum_{\mathbf{a}' \in \mathcal{A}(\mathbf{s}')} F(\mathbf{s}', \mathbf{a}') \tag{1}$$

Assuming that the flow of the MDP is known, the following theorem can be proven.

Theorem 1 *[14] Let us define a policy π that sequentially generates trajectories starting in state $\mathbf{s}_0$ by sampling actions $\mathbf{a} \in \mathcal{A}(\mathbf{s})$ according to:*

$$\pi(\mathbf{a}|\mathbf{s}) = \frac{F(\mathbf{s}, \mathbf{a})}{F(\mathbf{s})} \tag{2}$$

where $F(\mathbf{s}, \mathbf{a}) > 0$ is the flow through allowed edge $(\mathbf{s}, \mathbf{a})$, $F(\mathbf{s}) = R(\mathbf{s}) + \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{s})} F(\mathbf{s}, \mathbf{a})$ where $R(\mathbf{s}) = 0$ for non terminal nodes $\mathbf{s}$ and $F(\mathbf{x}) = R(\mathbf{x}) > 0$ for terminal nodes $\mathbf{x}$. The flow consistency equation $\sum_{\mathbf{s},\mathbf{a}:T(\mathbf{s},\mathbf{a})=\mathbf{s}'} F(\mathbf{s}, \mathbf{a}) = R(\mathbf{s}') + \sum_{\mathbf{a}' \in \mathcal{A}(\mathbf{s}')} F(\mathbf{s}', \mathbf{a}')$ is satisfied at every node. Let $\pi(\mathbf{s})$ denote the probability of visiting state $\mathbf{s}$ when starting at $\mathbf{s}_0$ and following $\pi(\cdot|\cdot)$. Then

$$\pi(\mathbf{s}) = \frac{F(\mathbf{s})}{F(\mathbf{s}_0)} \tag{3}$$

$$F(\mathbf{s}_0) = \sum_{\mathbf{x} \in \mathcal{X}} R(\mathbf{x}) \tag{4}$$

$$\pi(\mathbf{x}) = \frac{R(\mathbf{x})}{\sum_{\mathbf{x}' \in \mathcal{X}} R(\mathbf{x}')} \tag{5}$$

The above theorem suggests that if $F(\mathbf{s}, \mathbf{a})$ is known for all state-action pairs in the MDP, then by sampling trajectories using the flow-induced policy (eq. (2)), the probability of reaching the terminal state $\mathbf{x}$ is proportional to the reward $R(\mathbf{x})$. The GFlowNet paradigm involves parameterizing the flow F using a function approximator $F_{\mathbf{w}}$. Trajectories of the MDP are sampled, and for each state $\mathbf{s}'$, we minimize the following objective using gradient descent on $\mathbf{w}$:

$$L_{\mathbf{w}}(\mathbf{s}') = \sum_{\mathbf{s},\mathbf{a}:T(\mathbf{s},\mathbf{a})=\mathbf{s}'} F_{\mathbf{w}}(\mathbf{s}, \mathbf{a}) - R(\mathbf{s}') - \sum_{\mathbf{a}' \in \mathcal{A}(\mathbf{s}')} F_{\mathbf{w}}(\mathbf{s}', \mathbf{a}') \tag{6}$$

3. GFLOWNET FOR SENSOR SELECTION

The problem of sensor selection involves selecting a subset of M active sensors from a sensor array with N elements, in order to optimize a performance metric Q. With $\binom{N}{M}$ possible subsets to choose from, the challenge lies in identifying the subset that maximizes (or minimizes) the performance metric.

The sensor selection MDP is defined as follows: each state $\mathbf{s}$ is a binary vector of N elements. The "1" elements of the state represent active sensor elements in the corresponding sensor positions, while the "0" elements represent inactive sensor elements. The initial state $\mathbf{s}_0$ is the zero vector, where all elements are inactive. The action space is also discrete, with permissible actions at each state corresponding to the addition of one extra active element from the set of inactive ones. The terminal states correspond to state vectors with exactly M active elements ($\mathbf{s} \equiv \mathbf{x} \in \mathcal{X}$ iff $\|\mathbf{s}\|_1 = M$).

The reward for all intermediate states ($\|\mathbf{s}\|_1 < M$) is zero. For all terminal states $\mathbf{x}$, the reward is defined based on the evaluation of the performance objective ($f(Q(\mathbf{x}))$. The general graph corresponding to the sensor selection MDP is illustrated in Fig. 1. Assuming that $Q(\mathbf{x}) > 0 \; \forall \mathbf{x}$, the reward function can be defined to be $R(\mathbf{x}) = Q(\mathbf{x})$ if maximization is required, or as $R(\mathbf{x}) = \frac{1}{Q(\mathbf{x})}$ if minimization is required. The proposed algorithm for sensor selection, denoted

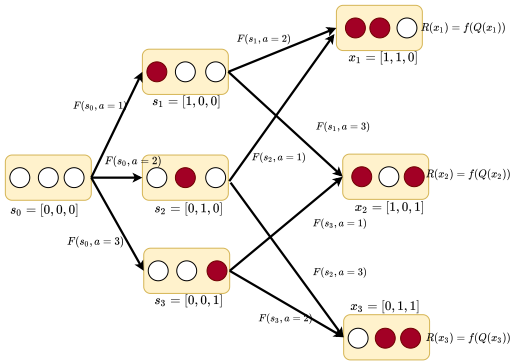

Fig. 1. Sensor selection MDP that corresponds to selecting 2 sensor elements out of 3. The red circles correspond to active elements whereas white circles denote inactive elements.

as GFlowNet for sensor selection (**GFLOW-SS**), is presented in Algorithm 1. **GFLOW-SS** parameterizes the MDP flow with a neural network and trajectories are sampled on-policy. The network's weights are updated using stochastic gradient descent on the trajectory balance loss (eq. (6)). Following training, subsets can be sampled as terminal states using the action policy defined in eq. (2). Importantly, this method is objective-agnostic, as it doesn't require annotations or make assumptions about the smoothness, differentiability, or convexity of the system performance objective. The only requirement is that the objective can be evaluated for a given sensor element subset. This provides an advantage over analytical methods proposed in [1–4], which rely on application-specific properties of the objective to utilize convex relaxations or greedy selection.

Algorithm 1 GFLOW - SS

Initialize $F_{\mathbf{w}}$, $\epsilon \in [0, 1]$ for exploration, learning rate η
for `all root-to-leaf trajectories` **do**
  $\mathbf{s} = \mathbf{s}_0$
  for $M - 1$ transitions **do**
    Sample $z \sim \mathcal{U}(0, 1)$ (Uniform distribution)
    If $z > \epsilon$ choose $\mathbf{a} \in \mathcal{A}(\mathbf{s})$ randomly
    If $z \leq \epsilon$ choose $\mathbf{a} = \arg\max_{\mathbf{a}'} F_{\mathbf{w}}(\mathbf{s}, \mathbf{a}')$
    Apply action $\mathbf{a}$, compute sub. state $\mathbf{s}'$
    $\mathbf{w}' \rightarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} L_{\mathbf{w}}(\mathbf{s}')$ eq. (6) $R(\mathbf{s}') = 0$
    $\mathbf{s} = \mathbf{s}'$
  end for
  Sample $z \sim \mathcal{U}(0, 1)$
  If $z > \epsilon$ choose $\mathbf{a} \in \mathcal{A}(\mathbf{s})$ randomly
  If $z \leq \epsilon$ choose $\mathbf{a} = \arg\max_{\mathbf{a}'} F_{\mathbf{w}}(\mathbf{s}, \mathbf{a}')$
  Apply action $\mathbf{a}$, compute terminal state $\mathbf{x}$
  $\mathbf{w}' \rightarrow \mathbf{w} - \eta \nabla_{\mathbf{w}} L_{\theta}(\mathbf{x})$ $R(\mathbf{x}) = Q(\mathbf{x})$ or $\frac{1}{Q(\mathbf{x})}$
end for

3.1. Antenna Selection for MIMO Radar

We consider a uniform linear sensor array consisting of N antenna elements. The antennas are equally spaced, with the distance between two consecutive elements denoted as d. Each antenna transmits narrowband signals with a carrier wavelength of λ. The array output at angle θ is:

$$y(t; \theta) = \mathbf{a}(\theta)^H \mathbf{v}(t), \tag{7}$$

where $\mathbf{a}(\theta)$ is the steering vector at direction θ and $\mathbf{v}(t) \in \mathbb{C}^N$ is the transmitted array snapshot at time t. Let $\mathbf{v}(t) = \mathbf{C}\mathbf{e}(t)$ where $\mathbf{e}(t) \in \mathbb{C}^N$ is a white signal vector and $\mathbf{C} \in \mathbb{C}^{N \times N}$ is a precoding matrix. The array output vector at K different angles is:

$$\mathbf{y}(t) = \big[y(t; \theta_1), \ldots, y(t; \theta_K)\big] = \mathbf{A}^H \mathbf{v}(t) \tag{8}$$

with $\mathbf{A} = \big[\mathbf{a}(\theta_1), \ldots, \mathbf{a}(\theta_K)\big] \in \mathbb{C}^{N \times K}$ being the steering matrix. Assuming that a binary selection vector $\mathbf{x}$ is provided ($\|\mathbf{x}\|_1 = M$), the sparse array output:

$$\mathbf{y}_{sparse}(t) = \mathbf{A}^H \mathbf{S}(\mathbf{x}) \mathbf{v}(t) \tag{9}$$

The matrix $\mathbf{S}(\mathbf{x}) = \text{diag}(\mathbf{x}) \in \mathbb{R}^{N \times N}$ is a selection matrix that corresponds to the selection vector $\mathbf{x}$. Assuming that the selection vector $\mathbf{x}$ and the precoding matrix $\mathbf{C}$ have been provided, the output power of the sparse array at angle θ_i is:

$$\hat{p}_i = \mathbf{a}^H(\theta_i) \mathbf{S}(\mathbf{x}) \mathbf{C}\mathbf{C}^H \mathbf{S}(\mathbf{x}) \mathbf{a}(\theta_i) \tag{10}$$

To solve the selection problem, we start with the desired signal power for all K angles, denoted by the vector $\mathbf{p}_d = [p_1, p_2, \ldots, p_K]^T$. The objective is to find the selection vector $\mathbf{x}$ and the precoding matrix $\mathbf{C}$ that minimize the difference between the desired power and the sparse array output power (beampattern error):

$$Q(\mathbf{x}, \mathbf{C}) = \|\mathbf{p} - \text{diag}\{\mathbf{A}^H \mathbf{S}(\mathbf{x}) \mathbf{C}\mathbf{C}^H \mathbf{S}(\mathbf{x}) \mathbf{A}\}\|_2 \tag{11}$$

The **GFLOW-SS** algorithm utilizes the terminal state reward $R(\mathbf{x})$, which is chosen as the inverse of the beampattern error $(\frac{1}{Q(\mathbf{x},\mathbf{C})})$. However, the beampattern error depends not only on the selection vector $\mathbf{x}$ but also on the precoding matrix $\mathbf{C}$ (this is the reason that the notation of the performance objective in (11) is not only a function of the subset but of the precoding matrix as well). Therefore, upon obtaining the selection vector $\mathbf{x}$ as the terminal state of the MDP, the optimal precoding matrix is computed through a fixed number of gradient descent steps, minimizing the beampattern error (eq. (11)). Since this objective is a function of a complex matrix $\mathbf{C}$, we implement the Wirtinger conjugate gradient method [17].

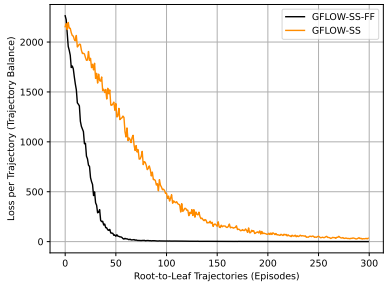

Fig. 2. The trajectory loss for 300 trajectories of the sensor selection MDP for **GFLOW-SS** and **GFLOW-SS-FF**.

3.2. Fourier features

One critical question is how to parameterize the flow network $F_{\mathbf{w}}$. The basic **GFLOW-SS** algorithm employs a rectified linear (ReLU) multilayer perceptron (MLP) as a function approximator. However, recent graphics research [16] suggests that ReLU MLPs are ineffective for binarized inputs due to *Spectral Bias* [15]. To tackle this issue, we propose a modified version of **GFLOW-SS** called **GFLOW-SS-FF**. In **GFLOW-SS-FF**, we replace the input layer of the flow network with a learnable Fourier features kernel, which includes a linear transformation and a sinusoidal activation [18]. For kernel element initialization, we adopt a zero-mean Gaussian distribution with a variance of 0.001.

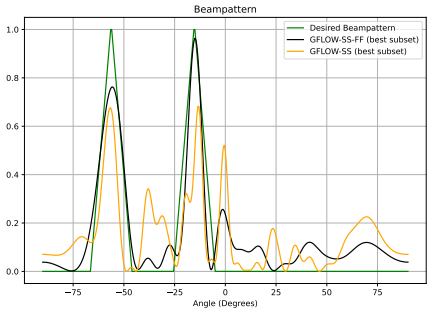

Fig. 3. The beampatterns achieved by the best subsets sampled by **GFLOW-SS** and **GFLOW-SS-FF** in comparison to the desired beampattern. The selected subsets are comprised by $M = 10$ elements and the array size is $N = 40$.

3.3. Experiments

The goal is to choose 12 out of 42 antenna elements, spaced at a distance of $d = \frac{\lambda}{2}$, where $\lambda = 1.25$ cm represents the wavelength. To maintain a fixed aperture, the subarray always includes the first and last sensors, resulting in the selection of $M = 10$ out of $N = 40$ elements. The desired beampattern consists of two main lobes centered at $-50°$ and $-30°$ ($0°$ denotes the direction perpendicular to the array face) and a width of 20 degrees for each lobe. To model the MDP flow of **GFLOW-SS**, we utilize a 3-layer dense neural network with ReLU activations, incorporating 512 neurons per layer. The Adam optimizer [19] with a learning rate of 0.0001 is employed for optimization. In the case of **GFLOW-SS-FF**, the first layer is substituted with the Fourier kernel, ensuring that both variations maintain an equivalent number of parameters.

The flow network was trained using 300 root-to-leaf-node trajectories for both variations. Each trajectory's leaf corresponds to a specific subset out of the total of $847,660,528$ possible subsets resulting from choosing 10 elements out of 40 ($\binom{40}{10} = 847,660,528$). As depicted in Fig. 2, the trajectory balance loss (eq. (6)) for both variations approaches values close to 0 by the end of training. Notably, **GFLOW-SS-FF** exhibits a significantly faster loss reduction rate compared to **GFLOW-SS**. Furthermore the loss of the **GFLOW-SS-FF** is lower than that of **GFLOW-SS** at the end of training. A smaller loss indicates a more accurate estimation of the flow.

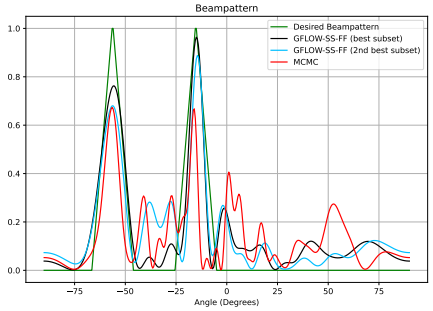

Fig. 4. The beampatterns corresponding to the best and 2nd best subsets generated by **GFLOW-SS-FF** and the beampattern that corresponds to the subset generated by **MCMC**. $M = 10$ and $N = 40$.

To sample the optimal subset using the trained flow network, the greedy sampling strategy is employed by selecting the action that maximizes the flow at each state. Consequently, the leaf node of the trajectory corresponds to the subset that minimizes the beampattern error. Fig.3 presents the beampatterns of the best subsets generated by **GFLOW-SS** and **GFLOW-SS-FF** respectively. Notably, the beampattern associated with **GFLOW-SS-FF** is closer to the desired one compared to the beampattern of the best **GFLOW-SS** subset. This disparity can be attributed to the superior capability of the function approximator of **GFLOW-SS-FF** in capturing the MDP flow within the given number of trajectories (as evidenced by the lower loss at the end of training).

We compare the beampattern produced by the best subset generated by **GFLOW-SS-FF** with the beampattern that corresponds to the subset generated by Metropolis-Hastings **MCMC** [13], another unsupervised method for sampling from distributions over complex structures. To conduct

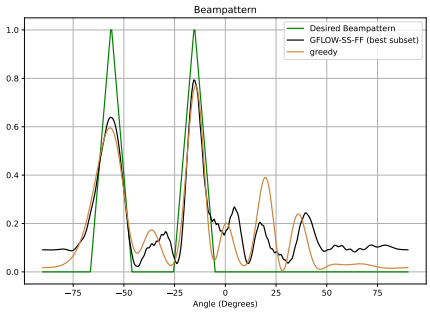

Fig. 5. The beampatterns corresponding to the best subset generated by **GFLOW-SS-FF** and the one generated by the greedy approach of [5]. The number of antenna elements is $N = 5$ from which $M = 100$ are selected.

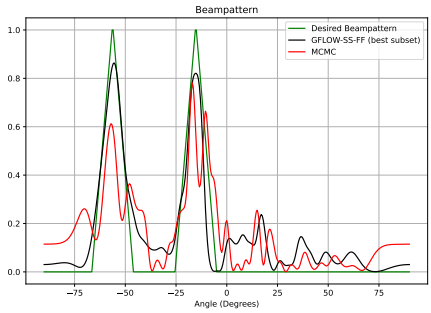

Fig. 6. The beampatterns corresponding to the best subset generated by **GFLOW-SS-FF** and the one generated by **MCMC**. The number of antenna elements is $N = 500$ from which $M = 8$ are selected.

MCMC, we process 3000 root-to-leaf trajectories of the MDP, which is 10 times more than the number of trajectories used for the GFlowNet-based methods. However, despite the increased number of trajectories, the proposed **GFLOW-SS-FF** paradigm outperforms **MCMC**, as evidenced by the significantly closer beampattern to the desired one (Fig. 4).

Fig. 5 illustrates the comparison between the best subset generated by **GFLOW-SS-FF** and the greedy approach proposed in [5]. The selection problem involves choosing $N = 5$ elements out of $M = 100$. The **GFLOW-SS-FF** approach is trained using only 300 trajectories. The beampatterns of the two approaches are similar. Notably, the **GFLOW-SS-FF** approach achieves its beampattern within approximately 3 minutes, while the method in [5] requires 36 minutes.

GFlowNet, trained to represent a distribution over sampling policies rather than maximizing reward, enables the generation of not only the best subset but also subsequent subsets without additional training. For instance, to sample the second-best subset, we select the action corresponding to the second-highest flow value at the current state. The beampattern of the second best subset produced by **GFLOW-SS-FF** is also included in Fig. 4. The beampattern of the second best subset surpasses the one that corresponds to the **MCMC** subset . Training the flow network for each one of the GFlowNet-based methods took approximately 4 minutes on a Quadro RTX8000 NVIDIA GPU with 48GB memory.

Fig.7 illustrates the beampattern error attained by **GFLOW-SS-FF**, **GFLOW-SS**, and **MCMC** for varying values of M when $N = 40$. Each method is trained independently for each value of M. Fig. 6 demonstrates the beampatterns of the subset generated by **GFLOW-SS-FF** and the one generated by **MCMC** for a large example where $N = 500$ and $M = 8$. Regarding the large example, **GFLOW-SS-FF** was trained for 1000 trajectories whereas **MCMC** was trained for 5000.

3.4. Discussion

Both GFlowNet variations outperform MCMC sampling, producing array subsets with lower beampattern error within the given trajectory budget. The superiority of GFlowNet lies in Theorem 1 and its reformulation of the problem of learning a distribution over composite objects. By shifting from the combinatorial space of subsets to the continuous parameter space of the flow network, gradient descent optimization can be employed, enabling generalization to unseen inputs.

For $M = 10$ and $N = 40$, both **GFLOW-SS** and **GFLOW-SS-FF** recover well-performing subsets by processing only 300 out of approximately 850 million possible configurations (less than 0.0001% of all subarrays). Each variation requires $10 \times 300 = 3000$ gradient descent steps, without preprocessing or annotation. **GFLOW-SS-FF** has an advantage over **GFLOW-SS** that can be attributed to the Spectral Bias of the function approximator of the latter. The use of the Fourier preprocessing kernel results in convergence with fewer gradient steps and therefore fewer trajectories. The superior performance of **GFLOW-SS-FF** over **MCMC** remains in the large setting where $M = 8$ elements are selected out of $N = 500$.

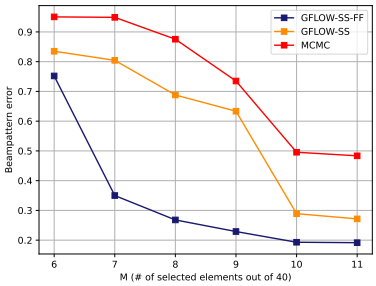

Fig. 7. The beampattern error achieved by **GFLOW-SS-FF**, **GFLOW-SS** and **MCMC** for different values of M when $N = 40$

4. CONCLUSION

We have proposed a novel deep generative modeling framework for sensor selection. Instead of posing the selection problem as a multilabel classification, we have modeled the selection process as a deterministic MDP, where the sensor subsets arise as terminal states. This formulation enables the use of GFlowNets in order to learn a distribution over the actions, conditioned on the state, such that the probability of arriving at each terminal state is proportional to the performance of the corresponding subset. The subset selection can be performed by sampling from the policy defined by the trained GFlowNet. The proposed approach is unsupervised

and therefore it requires no annotations. The method's efficacy has been demonstrated in antenna selection for MIMO radar transmit beamforming, effectively identifying high-performing subarrays by being trained on a small fraction of the possible subsets and MDP trajectories.

5. REFERENCES

[1] Gal Shulkind, Stefanie Jegelka, and Gregory W Wornell, "Multiple wavelength sensing array design," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3424–3428.

[2] Hamed Nosrati, Elias Aboutanios, and David B Smith, "Receiver-transmitter pair selection in MIMO phased array radar," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 3206–3210.

[3] Siddharth Joshi and Stephen Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2008.

[4] Venkat Roy, Sundeep Prabhakar Chepuri, and Geert Leus, "Sparsity-enforcing sensor selection for DOA estimation," in *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. IEEE, 2013, pp. 340–343.

[5] Hana Godrich, Athina P Petropulu, and H Vincent Poor, "Sensor selection in distributed multiple-radar architectures for localization: A knapsack problem formulation," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 247–260, 2011.

[6] Weiguang Wang, Yingbin Liang, Eric P. Xing, and Lixin Shen, "Sparse sensor selection for nonparametric decentralized detection via l1 regularization," in *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2014, pp. 1–6.

[7] Jingon Joung, "Machine learning-based antenna selection in wireless communications," *IEEE Communications Letters*, vol. 20, no. 11, pp. 2241–2244, 2016.

[8] Ahmet M Elbir, Kumar Vijay Mishra, and Yonina C Eldar, "Cognitive radar antenna selection via deep learning," *IET Radar, Sonar & Navigation*, vol. 13, no. 6, pp. 871–880, 2019.

[9] Thang X Vu, Symeon Chatzinotas, Van-Dinh Nguyen, Dinh Thai Hoang, Diep N Nguyen, Marco Di Renzo, and Björn Ottersten, "Machine learning-enabled joint antenna selection and precoding design: From offline complexity to online performance," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3710–3722, 2021.

[10] Bo Lin, Feifei Gao, Shun Zhang, Ting Zhou, and Ahmed Alkhateeb, "Deep learning-based antenna selection and CSI extrapolation in massive MIMO systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7669–7681, 2021.

[11] Konstantinos Diamantaras, Zhaoyi Xu, and Athina Petropulu, "Sparse antenna array design for MIMO radar using softmax selection," *arXiv preprint arXiv:2102.05092*, 2021.

[12] Zhaoyi Xu, Fan Liu, and Athina Petropulu, "Cramér-Rao bound and antenna selection optimization for dual radar-communication design," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 5168–5172.

[13] Christian Robert, George Casella, Christian P Robert, and George Casella, "Metropolis–hastings algorithms," *Introducing Monte Carlo Methods with R*, pp. 167–197, 2010.

[14] Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio, "Flow network based generative models for non-iterative diverse candidate generation," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27381–27394, 2021.

[15] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville, "On the spectral bias of neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5301–5310.

[16] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7537–7547, 2020.

[17] Zhun Wei, Wen Chen, Cheng-Wei Qiu, and Xudong Chen, "Conjugate gradient method for phase retrieval based on the wirtinger derivative.," *Journal of the Optical Society of America. A, Optics, Image Science, and Vision*, vol. 34, no. 5, pp. 708–712, 2017.

[18] Spilios Evmorfos, Konstantinos I. Diamantaras, and Athina P. Petropulu, "Reinforcement learning for motion policies in mobile relaying networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 850–861, 2022.

[19] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.