

An Unsupervised Learning-Based Approach for Symbol-Level-Precoding

Abdullahi Mohammad, Christos Masouros, and Yiannis Andreopoulos

Abstract—This paper proposes an unsupervised learning-based precoding framework that trains deep neural networks (DNNs) with no target labels by unfolding an interior point method (IPM) proximal ‘log’ barrier function. The proximal ‘log’ barrier function is derived from the strict power minimization formulation subject to signal-to-interference-plus-noise ratio (SINR) constraint. The proposed scheme exploits the known interference via symbol-level precoding (SLP) to minimize the transmit power and is named strict Symbol-Level-Precoding deep network (SLP-SDNet). The results show that SLP-SDNet outperforms the conventional block-level-precoding (Conventional BLP) scheme while achieving near-optimal performance faster than the SLP optimization-based approach.

I. INTRODUCTION

Recent studies on interference exploitation have shown that known interferences can be effectively managed and transformed into valuable signals to improve the system’s quality-of-service (QoS) [1]. The concept of constructive interference (CI) is first introduced in [2], where instantaneous interference is categorized into constructive and destructive. Suboptimal strategies that exploit CI are first introduced in [3]. Optimal SLP schemes using convex optimization-based CI with strict phase constraints on the received constellation point are proposed in [1], [4]–[6]. Despite the performance benefits offered by the optimization-based SLP schemes, computational complexity is still an issue in their implementation on practical systems.

Due to the low computational cost of online training, there has been an increasing interest in designing ML-based precoding schemes recently for MU-MISO downlink transmission. Alkhateeb *et al.* in [7] propose an ML-based coordinated beamforming technique to improve the link reliability and low latency associated with millimeter-wave (mmWave) communications. A robust deep neural networks (DNNs) precoding method is introduced in [8] for decentralized decision making. In [9], Huang *et al.* propose a fast beamforming design based on unsupervised learning, in which the learning-based solution nearly approximates the solution obtained via the weighted minimum mean-squared error (WMMSE) algorithm with much lower complexity. A convolutional neural networks (CNNs) framework for downlink beamforming optimization using expert knowledge based on the known structure of optimal iterative solutions is proposed in [10]. However, most of the learning-based strategies mentioned above are based on supervised learning, where the constraints are implicitly contained in the training dataset obtained from conventional optimization solutions. This requires solving the optimization problem twice, first by traditional optimization and second by approximating the optimal solution

using DNN. However, if it was difficult to obtain the optimal solutions via conventional optimization methods, the learning-based solutions may be impractical.

This paper proposes a learning-based precoding scheme that requires no target level for power minimization problems under signal-to-interference-noise-ratio (SINR). The learning framework is designed by unfolding an IPM iterative algorithm via IPM proximal log barrier function that considers the convexity of the inequality constraint. A case scenario of strict phase angle rotation is considered under a known perfect channel condition.

II. SYSTEM MODEL

Suppose a MISO downlink channel in a single cell with N transmit antennas at the BS provides signals to K single-antenna users. The channel between users and the BS is assumed to be quasi-static flat-fading and is denoted by $\mathbf{h}_i \in \mathbb{C}^{N \times 1}$. Therefore, the received signal at user i is expressed as [4]

$$\tilde{y}_i = \mathbf{h}_i^T \sum_{k=1}^K \mathbf{v}_k e^{j(\phi_k - \phi_i)} d_k + N_i \quad (1)$$

where \mathbf{h}_i , \mathbf{v}_k , d_k , ϕ_k , N_i denote the channel vector, precoding vector, data symbol, symbol of interest phase angle and received noise for the i -th user.

A. Conventional Power Minimization

Traditionally, the power minimization problem tries to minimize the average transmit power by handling all interference as harmful subject to QoS constraints, as described below [11]

$$\begin{aligned} \min_{\{\mathbf{v}_i\}} \quad & \sum_{i=1}^K \|\mathbf{v}_i\|_2^2 \\ \text{s.t.} \quad & \frac{|\mathbf{h}_i^T \mathbf{v}_i|^2}{\sum_{k=1, k \neq i}^K |\mathbf{h}_i^T \mathbf{v}_k|^2 + N_0} \geq \Gamma_i, \quad \forall i. \end{aligned} \quad (2)$$

where Γ_i is the SINR of the i -th user. From an instantaneous viewpoint, problem (2) does not consider the fact that interference can additively improve the received signal power [12]. Therefore, the solution is sub-optimal.

Fig. 1 shows the generic geometrical representation of the CI, where the received signal is expressed as $\tilde{\mathbf{y}}_i \triangleq \mathbf{h}_i^T \sum_{k=1}^K \mathbf{v}_k e^{j(\phi_k - \phi_1)}$. From the received symbol expression, the real and imaginary parts are respectively given by: $\lambda_{Re} = \text{Re}\left(\mathbf{h}_i^T \sum_{k=1}^K \mathbf{v}_k e^{j(\phi_k - \phi_1)}\right)$ and $\lambda_{Im} = \text{Im}\left(\mathbf{h}_i^T \sum_{k=1}^K \mathbf{v}_k e^{j(\phi_k - \phi_1)}\right)$. The real part of the received symbol (λ_{Re}) gives a measure of the received constellation along the theoretical constellation axis. Likewise, the imaginary part (λ_{Im}) shows the extent of the phase displacement from the primary constellation point.

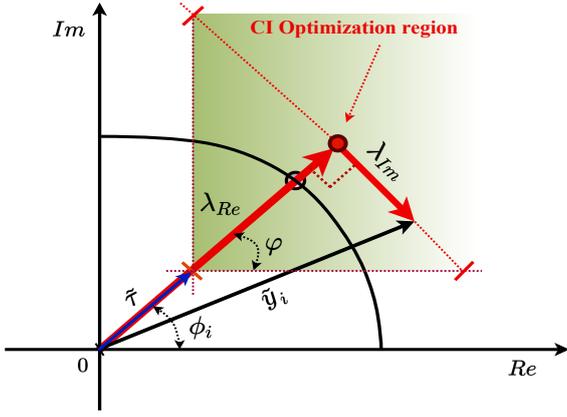


Fig. 1: Generic geometrical optimization regions for interference exploitation for Precoding design in QPSK [4]

B. Power Minimization via Symbol-Level Precoding

The instantaneous interference in a multi-user downlink channel scenario for M-phase shift keying (M-PSK) modulation can be categorized into constructive and destructive based on the known criteria defined in [13]. Therefore, CI is defined as the interference that pushes the received symbols away from the modulated-symbol constellation's decision edges [4]. For further details on SLP and its formulation (3), we refer the reader to [14]. Therefore, the problem in (2) is modeled to incorporate CI in the power minimization formulation. Consequently, the interfering signals align with the symbol of interest constructively by precoding vectors, offering useful signals. Hence, for M-PSK, the power minimization SLP-based optimization can be reformulated based on the classification criteria explained in [3] as

$$\begin{aligned} \min_{\{\mathbf{v}_i\}} & \left\| \sum_{k=1}^K \mathbf{v}_k e^{j(\phi_k - \phi_1)} \right\|_2^2 \\ \text{s.t.} & \angle \left(\mathbf{h}_i^T \sum_{k=1}^K \mathbf{v}_k s_k \right) = \angle(s_i), \forall i \\ & \text{Re} \left(\mathbf{h}_i^T \sum_{k=1}^K \mathbf{v}_k e^{j(\phi_k - \phi_i)} \right) \geq \sqrt{\Gamma_i N_0}, \forall i. \end{aligned} \quad (3)$$

The real part of the received symbol in (1) gives a measure of the received constellation along the theoretical constellation axis. Likewise, the imaginary part shows the extent of the phase displacement from the primary constellation point.

III. LEARNING-BASED SLP FOR POWER MINIMIZATION PROBLEM

This section presents detailed formulations of a learning-based CI power minimization problem for strict phase angle rotation assuming a perfect channel state information (CSI) known at the BS. The power minimization problem for the case where the phase angle of the interfering symbols strictly aligns with the angle of the symbols of interest is obtained from (3). If the maximum angle shift in the constructive interference region is zero, i.e., all the interfering signals completely overlap on the signal

of interest ($\varphi = 0$, see [14] for details), the optimization problem is formulated as [4]

$$\begin{aligned} P_{\text{st}}: \\ \min_{\{\mathbf{v}_i\}} & \left\| \sum_{k=1}^K \mathbf{v}_k e^{j(\phi_k - \phi_1)} \right\|_2^2 \\ \text{s.t.} & \text{Im} \left(\mathbf{h}_i^T \sum_{k=1}^K \mathbf{v}_k e^{j(\phi_k - \phi_i)} \right) = 0, \forall i \\ & \text{Re} \left(\mathbf{h}_i^T \sum_{k=1}^K \mathbf{v}_k e^{j(\phi_k - \phi_i)} \right) \geq \sqrt{\Gamma_i N_0}, \forall i. \end{aligned} \quad (4)$$

It is often difficult to derive the closed-form solution to problem P_{st} due to the in-equality constraints. Therefore, conventional iterative solvers are usually used to find sub-optimal solutions. Motivated by the recent adoption of an IPM for image restoration [15], we propose an unsupervised learning framework that unfolds a constrained optimization problem into a sequence of neural network layers for a multi-user MIMO beamforming. We first convert (4) to a general form of proximal IPM. The measure of the fidelity of the solution to (4) is determined by learning a set of penalty parameters in the form of Lagrange multipliers associated with the constraints. We define the channel vector based on (4) as follows

$$\tilde{\mathbf{h}}_i = \mathbf{h}_i \sum_{k=1}^K e^{j(\phi_k - \phi_i)} \quad (5)$$

$$\mathbf{v} = \sum_{k=1}^K \mathbf{v}_k. \quad (6)$$

The product of complex vectors (5) and (6) can be written as

$$\tilde{\mathbf{h}}_i \mathbf{v} = (\tilde{\mathbf{h}}_{Ri} + j\tilde{\mathbf{h}}_{Ii})(\mathbf{v}_R + j\mathbf{v}_I) \quad (7)$$

where $\tilde{\mathbf{h}}_R = \text{Re}(\tilde{\mathbf{h}}_i)$, $\tilde{\mathbf{h}}_I = \text{Im}(\tilde{\mathbf{h}}_i)$, $\mathbf{v}_R = \text{Re}(\mathbf{v})$ and $\mathbf{v}_I = \text{Im}(\mathbf{v})$.

Let $\Upsilon = [\tilde{\mathbf{h}}_R; \tilde{\mathbf{h}}_I]$, $\mathbf{v}_1 = [\mathbf{v}_R - \mathbf{v}_I]^T$ and $\mathbf{v}_2 = [\mathbf{v}_I \mathbf{v}_R]^T$. To simplify the analysis, we partition the complex vectors into the real and imaginary parts as follows: $\text{Re}(\tilde{\mathbf{h}}_i^T \mathbf{v}) = \Upsilon_i^T \mathbf{v}_1$ and $\text{Im}(\tilde{\mathbf{h}}_i^T \mathbf{v}) = \Upsilon_i^T \Omega \mathbf{v}_1$, where

$$\mathbf{v}_2 = \Omega \mathbf{v}_1 \text{ and } \Omega = \begin{bmatrix} \mathbf{O}_N & -\mathbf{I}_N \\ \mathbf{I}_N & \mathbf{O}_N \end{bmatrix}; \in \mathbb{R}^{2N \times 2N}. \quad (8)$$

From (7), we have $\mathbf{v} = [1 \quad -j1] \begin{bmatrix} \mathbf{v}_R \\ -\mathbf{v}_I \end{bmatrix} = [1 \quad -j1] \mathbf{v}_1$.

Substituting for \mathbf{v} in (4) and simplify, P_{st} finally becomes

$$\begin{aligned} P_{\text{st}1}: \\ \min_{\{\mathbf{v}_1\}} & \|\mathbf{v}_1\|^2 \\ \text{s.t.} & \Upsilon_i^T \Omega \mathbf{v}_1 = 0, \forall i \\ & \Upsilon_i^T \mathbf{v}_1 \geq \sqrt{\Gamma_i N_0}, \forall i. \end{aligned} \quad (9)$$

A. Interior Point Method (IPM)

Consider a general form of a nonlinear constrained optimization of the form [16]:

$$\begin{aligned} \min_{\mathbf{z} \in \mathbb{R}^N} \quad & f(\mathbf{z}) \\ \text{s.t.} \quad & \mathcal{C}(\mathbf{z}) = 0 \\ & \mathbf{z} \geq 0. \end{aligned} \quad (10)$$

The reason for adopting IPM is to replace the initial constrained optimization problem with a chain of unconstrained sub-problems of the form:

$$\min_{\mathbf{z} \in \mathbb{R}^N} f(\mathbf{z}) + \lambda \mathcal{D}(\mathbf{z}) + \mu \mathcal{B}(\mathbf{z}). \quad (11)$$

where \mathcal{B} is the logarithmic barrier function associated with inequality constraint with unbounded derivative at the boundary of the feasible domain, \mathcal{D} is associated with equality constraint, μ and λ are the Lagrangian multipliers for inequality and equality constraints, respectively.

To facilitate the solution of $P_{\text{st}1}$, we introduce additional notations. For every inequality constraint, $\gamma \in \{0, +\infty\}$ and $\mathbf{v}_1 \in \mathbb{R}^{2N \times 1}$, we define the proximity operator as in [16] with respect to (11), which we shall later use to compute the projected gradient descent as

$$\text{prox}_{\gamma \mathcal{G}}(\mathbf{v}_1) = \underset{\mathbf{v}_1 \in \mathbb{R}^{2N \times 1}}{\text{argmin}} \quad \frac{1}{2} \|\mathbf{v}_0 - \mathbf{v}_1\|_2^2 + \gamma \mathcal{G}(\mathbf{v}_1) \quad , \quad (12)$$

where γ is the step-size taken for computing the gradients of the objective function, \mathcal{G} is the function that defines the barrier operator and \mathbf{v}_0 is the initial value of the precoding vector.

To convert (3) into its equivalent barrier function problem, we get rid of the inequality constraint and translate it into a barrier term of the form [17]

$$\begin{aligned} \min_{\mathbf{x}_i \in \mathbb{R}^n} \quad & f(\mathbf{z}) - \mu \sum_{i=1}^p \ln(\mathbf{x}_i) \\ \text{s.t.} \quad & \mathcal{D}(\mathbf{z}) = 0. \end{aligned} \quad (13)$$

1) *Affine Constraints:* Consider a half-space constraint expressed as [15]:

$$\mathcal{C} = \{\mathbf{z} \in \mathbb{R}^N | b^T \mathbf{z} \leq c\} \quad (14)$$

As shown in [15], the \mathcal{B} function associated to (14) is defined as

$$\mathcal{B}(\mathbf{z}) = \begin{cases} -\ln(c - b^T \mathbf{z}), & \text{if } b^T \mathbf{z} < c, \forall \mathbf{z} \in \mathbb{R}^N \\ +\infty, & \text{otherwise} \end{cases} \quad (15)$$

Following (13), we can express (??) as

$P_{\text{st}2}$:

$$\begin{aligned} \min_{\mathbf{v}_1} \quad & f(\mathbf{v}_1) - \mu \sum_{i=1}^p \ln(\mathbf{\Upsilon}_i^T \mathbf{v}_1 - \sqrt{\Gamma_i N_0}) \\ \text{s.t.} \quad & \mathbf{\Upsilon}_i^T \mathbf{\Omega} \mathbf{v}_1 = 0. \end{aligned} \quad (16)$$

For all $\mu > 0$, $\lambda > 0$ and \mathbf{v}_1 , we define \mathcal{B} as in (15), so that the proximity operator can be defined as follows

$$\Phi(\mathbf{v}_1, \gamma, \mu) = \text{prox}_{\gamma \mu \mathcal{B}}(\mathbf{v}_1). \quad (17)$$

In what follows, we provide the expression of Φ and its corresponding derivatives with respect to the optimization variable \mathbf{v}_1 , the step-size and the barrier parameters

(γ, μ) for affine constraint, which will be used for training the neural network using a gradient backpropagation algorithm. Finally, following the above formulations, the proximal barrier function for the strict phase rotation is reduced to the following expression

$$\mathcal{B}(\mathbf{v}_1) = \begin{cases} -\ln(\mathbf{\Upsilon}_i^T \mathbf{v}_1 - \sqrt{\Gamma_i N_0}), & \text{if } \mathbf{\Upsilon}_i^T \mathbf{v}_1 \geq \sqrt{\Gamma_i N_0} \\ +\infty, & \text{otherwise.} \end{cases} \quad (18)$$

It can be easily shown that for every precoding vector $\mathbf{v}_1 \in \mathbb{R}^{2N \times 1}$, the proximity operator of $\mu \gamma \mathcal{B}$ at \mathbf{v}_1 is given by

$$\begin{aligned} \Phi(\mathbf{v}_1, \mu, \gamma) = \mathbf{v}_1 + \\ \frac{\mathbf{\Upsilon}_i^T \mathbf{v}_1 - \sqrt{\Gamma_i N_0} - \sqrt{(\mathbf{\Upsilon}_i^T \mathbf{v}_1 - \sqrt{\Gamma_i N_0})^2 + 4\gamma\mu \|\mathbf{\Upsilon}_i^T\|_2^2}}{2 \|\mathbf{\Upsilon}_i\|_2^2} \mathbf{\Upsilon}_i. \end{aligned} \quad (19)$$

Furthermore, the Jacobian matrix of Ω with respect to \mathbf{v}_1 , and the derivatives of Ω with respect to γ and μ are as follows

$$\begin{aligned} \mathcal{J}_{\Phi} |_{(\mathbf{v}_1)} = \mathbf{I}_{2N_t} + \frac{1}{2 \|\mathbf{\Upsilon}_i\|_2^2} \times \\ \left(1 - \frac{\mathbf{\Upsilon}_i^T - \sqrt{\Gamma_i N_0}}{\sqrt{(\mathbf{\Upsilon}_i^T \mathbf{v}_1 - \sqrt{\Gamma_i N_0})^2 + 4\gamma\mu \|\mathbf{\Upsilon}_i^T\|_2^2}} \right) \mathbf{\Upsilon}_i \mathbf{\Upsilon}_i^T \end{aligned} \quad (20)$$

$$\Delta_{\Phi} |_{(\mu)} = \frac{-\gamma}{\sqrt{(\mathbf{\Upsilon}_i^T \mathbf{v}_1 - \sqrt{\Gamma_i N_0})^2 + 4\gamma\mu \|\mathbf{\Upsilon}_i^T\|_2^2}} \mathbf{\Upsilon}_i \quad (21)$$

$$\Delta_{\Phi} |_{(\gamma)} = \frac{-\mu}{\sqrt{(\mathbf{\Upsilon}_i^T \mathbf{v}_1 - \sqrt{\Gamma_i N_0})^2 + 4\gamma\mu \|\mathbf{\Upsilon}_i^T\|_2^2}} \mathbf{\Upsilon}_i \quad (22)$$

where $\mathbf{I} \in \mathbb{R}^{2(N_t \times N_t)}$ is identity matrix.

Finally, the learning algorithm for every update rule is thus the unfolded $P_{\text{st}1}$ as a sequence of sub-problems with respect to the constraints as follows

$$\min_{\mathbf{v}_1 \in \mathbb{R}^{2N \times 1}} \|\mathbf{v}_1\|_2^2 + \lambda(\mathbf{\Upsilon}_i^T \mathbf{\Omega} \mathbf{v}_1) + \mu \mathcal{B}(\mathbf{v}_1). \quad (23)$$

Using the proximity operator of the barrier, the update rule for every iteration is given by

$$\mathbf{v}_1^{[r+1]} = \text{prox}_{\gamma^{[r]} \mu^{[r]} \mathcal{B}} \left(\mathbf{v}_1^{[r]} - \gamma^{[r]} \Delta_{\mathcal{E}}(\mathbf{v}_1^{[r]}, \lambda^{[r]}) \right), \quad (24)$$

where

$$\mathcal{E}(\mathbf{v}_1^{[r]}, \lambda^{[r]}) = \|\mathbf{v}_1\|_2^2 + \lambda(\mathbf{\Upsilon}_i^T \mathbf{\Omega} \mathbf{v}_1). \quad (25)$$

The update function can thus be expressed as

$$\begin{aligned} \mathcal{H}(\mathbf{v}_1^{[r]}, \gamma^{[r]}, \mu^{[r]}, \lambda^{[r]}) = \\ \text{prox}_{\gamma^{[r]} \mu^{[r]} \mathcal{B}} \left(\mathbf{v}_1^{[r]} - \gamma^{[r]} \Delta_{\mathcal{E}}(\mathbf{v}_1^{[r]}, \lambda^{[r]}) \right), \end{aligned} \quad (26)$$

and $\Delta = \frac{\partial \mathcal{E}(\mathbf{v}_1^{[r]}, \lambda^{[r]})}{\partial \mathbf{v}_1^{[r]}}$.

2) *Duality and Loss Function for the Strict Phase Formulation*: Since we are interested in learning the optimal solution via unsupervised learning (without target labels), we firstly formulate a primal-dual problem. This formulation is then used to derive the optimization variable (precoding vector) as a function of dual variables (Lagrangian multipliers) associated with the constraints. The Lagrangian function can be expressed as

$$\mathcal{L}_{\text{st}}(\mathbf{v}_1, \lambda, \mu) = \|\mathbf{v}_1\|_2^2 + \sum_{k=1}^K \lambda_k \Upsilon_i^T \Omega \mathbf{v}_1 + \sum_{k=1}^K \mu_k \left(\sqrt{\Gamma_i N_0} - \Upsilon_i^T \mathbf{v}_1 \right). \quad (27)$$

The optimal precoder can be found by minimizing (27) with respect to \mathbf{v}_1 (differentiating $\mathcal{L}_{\text{st}}(\cdot)$ w.r.t \mathbf{v}_1). The optimal precoder is thus

$$\mathbf{v}_1 = \frac{\boldsymbol{\mu}^T \cdot \Upsilon_i - \lambda^T \cdot \Omega \Upsilon_i}{2}. \quad (28)$$

The above expression in (28) is used to generate the training input (precoding vector) by initializing the Lagrangian multipliers (λ and μ) randomly and then train the neural network to learn their best values that minimize the loss function (Lagrangian function). The loss function is modified by adding ℓ_2 -norm regularization over the weights to adjust the learning coefficients to stabilize the learning process. The loss function over B training batches is finally expressed as

$$\mathcal{L}_{\text{st}}(\mathbf{v}_1, \lambda, \mu) = \frac{1}{B} \sum_{i=1}^B (\|\mathbf{v}_1\|_2^2 + \lambda \Upsilon_i^T \Omega \mathbf{v}_1) + \frac{1}{B} \sum_{i=1}^B \left(\mu \left(\sqrt{\Gamma_i N_0} - \Upsilon_i^T \mathbf{v}_1 \right) \right) + \frac{\vartheta}{BL} \sum_{i=1}^B \sum_{l=1}^L \|\boldsymbol{\theta}_i\|_2^2, \quad (29)$$

where $\boldsymbol{\theta}$ is the learning parameter associated with the weights and $\vartheta > 0$ is the penalty parameter that controls the bias and variance of the learning coefficients, B and L are training batch size (number of channel realization) and the number of layers respectively.

B. Deep Proximal Strict Symbol-Level Precoding Network (SLP-SDNet)

The optimization problem is unfolded over r -th iterations, and the Lagrange multiplier associated with the equality constraint is wired across the network to provide additional flexibility [15]. The unfolded neural network is trained in an unsupervised fashion without target labels. We build the structure of the learning framework based on (24) and the algorithm presented in [15], which gives rise to Algorithm 1.

For every r -th iterations (r -th layer) $\mathcal{L}^{[r]}$, there exist three latent structures associated with the learnable parameters (μ , γ and λ) $\mathcal{L}_\mu^{[r]}$, $\mathcal{L}_\gamma^{[r]}$ and $\mathcal{L}_\lambda^{[r]}$. As shown in Fig. 2, each of these structures forms a learning block for computing the barrier parameter (μ) associated with the inequality constraint, the step-size for update rule (γ) and finally (λ), which is related to the equality constraint and all of them must be positive. To impose such constraint,

Algorithm 1 Feed-forward-Backward Proximal IPM

Input: $\mathbf{v}_1^{[0]}$, $\gamma^{[0]}$, $\boldsymbol{\lambda}^{[0]}$ and $\boldsymbol{\mu}^{[0]}$

Output: \mathbf{v}_1

Initialization :

1: randomly initialize $\mathbf{v}_1^{[0]} \in \mathbb{R}^{2N_i \times 1}$, $\boldsymbol{\mu}^{[0]} > 0$, $\boldsymbol{\lambda}^{[0]} > 0$ and $\gamma^{[0]} > 0 \forall i = 1, \dots, K$

Loop over r -th iterations

2: **for** $r = 0$ to L **do**

3: $\mathbf{v}_1^{[r+1]} = \text{prox}_{\gamma^{[r]} \mu^{[r]} \mathcal{B}} \left(\mathbf{v}_1^{[r]} - \gamma^{[r]} \Delta \boldsymbol{\varepsilon}(\mathbf{v}_1^{[r]}, \lambda^{[r]}) \right)$.

4: **end for**

5: **return** \mathbf{v}_1

Algorithm 2 Proximity Barrier Operator for Strict phase rotation

Input: \mathbf{h}_R , \mathbf{h}_I , Γ_i and N_0 (noise power)

Output: \mathbf{v}_1 , γ , $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$

Initialization :

1: randomly initialize $\mathbf{v}_0 \in \mathbb{R}^{2N \times 1}$, $\boldsymbol{\mu}^{[0]} > 0$, $\boldsymbol{\lambda}^{[0]} > 0$ and $\gamma^{[0]} > 0 \forall i = 1, \dots, K$.

2: Compute the Barrier function $\mathcal{B}(\mathbf{v}_1)$ using function (15).

3: Compute the Proximity Operator of the Barrier at \mathbf{v}_0 using (12), where $\mathcal{G} = \mu \mathcal{B}(\mathbf{v}_1)$.

4: Compute the derivatives of the Proximity Operator w.r.t \mathbf{v}_1 , μ and γ using (20), (21) and (22).

5: Update the training variables as follows:

(a) $\mu^{[r+1]} = \mu^{[r]} - \eta \frac{\partial \Phi}{\partial \mu}$

(b) $\gamma^{[r+1]} = \gamma^{[r]} - \eta \frac{\partial \Phi}{\partial \gamma}$

(c) $\lambda^{[r+1]} = \lambda^{[r]} - \eta \frac{\partial \boldsymbol{\varepsilon}(\mathbf{v}_1^{[r]}, \lambda^{[r]})}{\partial \lambda^{[r]}}$ using (25)

where η is the learning rate.

6: Use the results in step 5 and the Algorithm 1 to obtain the optimal precoding tensor.

a ‘Softplus sign’ function is used. Hence, the step-size and the parameters associated with the constraints can all be estimated as $\gamma^{[r]} = \mathcal{L}_\gamma^{[r]} = \text{Softplus}(\mathbf{z}^{[r]})$. The output of the last three hidden structures is connected to an auxiliary processing module (APM) to convert it into the required transmit precoding vector. The APM consists of 4 convolution layers and 3 activation layers, a ‘Batch Normalization’ layer placed between them. Therefore, the Proximal Barrier function for a strict phase formulation is summarized in Algorithm 2.

Finally, the output from the auxiliary processing block is the precoding vector in the real domain. The relation: $\mathbf{v}_1 = [\mathbf{v}_R \quad -\mathbf{v}_I]^T$ is used to convert it to its equivalent complex domain for every SINR value of the i -th user.

C. The proposed Learning Structure and the general NN Architecture

Using (24) and Algorithm 1, we show a startling correlation between our scheme and the universal feed-forward DNN. Generally, an open-chained neural network (NN) structure can be derived from (24) as follows

$$\mathbf{v}_1^{[r+1]} = \text{prox}_{\gamma^{[r]} \mu^{[r]} \mathcal{B}} \left[\left(\mathbf{I}_N - 2\gamma^{[r]} \right) \mathbf{v}_1^{[r]} + \lambda^{[r]} \Upsilon_i \Omega \right]. \quad (30)$$

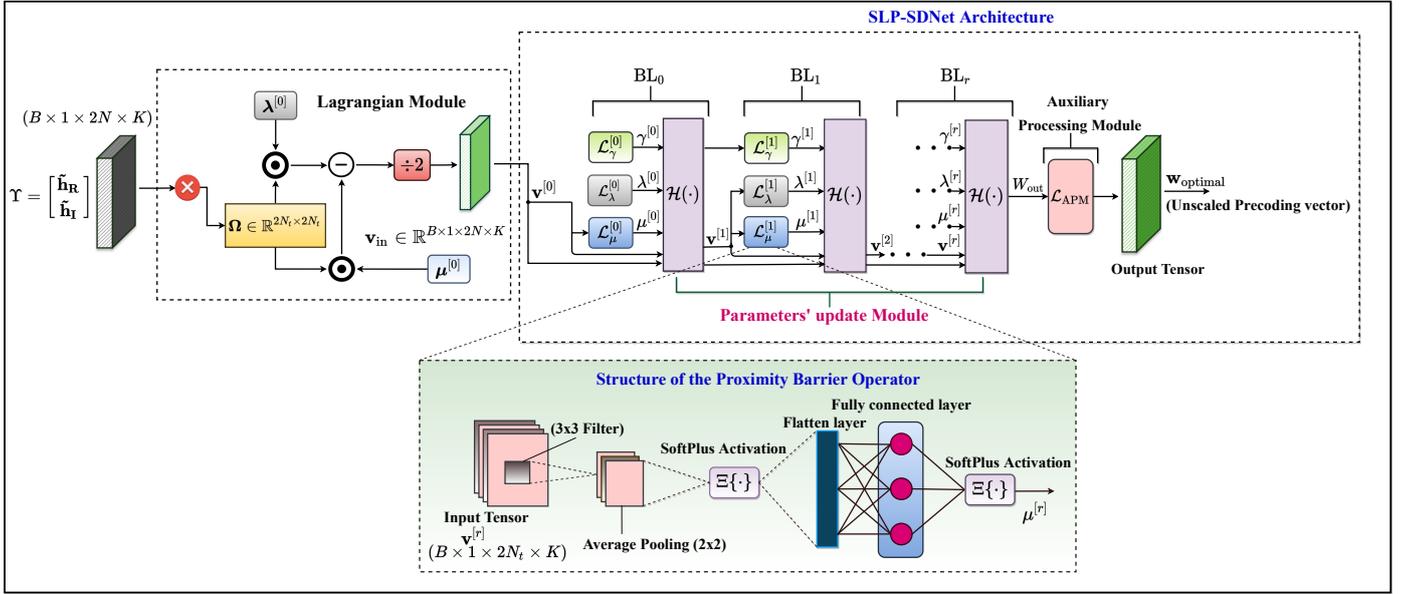


Fig. 2: Complete SLP-SDNet Architecture showing the internal structure of the Barrier Operator.

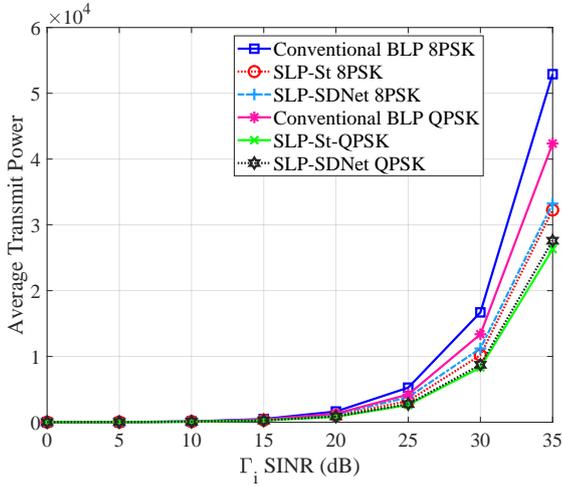


Fig. 3: Transmit Power vs SINR averaged over 2000 test samples for Conventional BLP, SLP-St optimization-based and SLP-SDNet schemes, $N = 4$, $K = 4$.

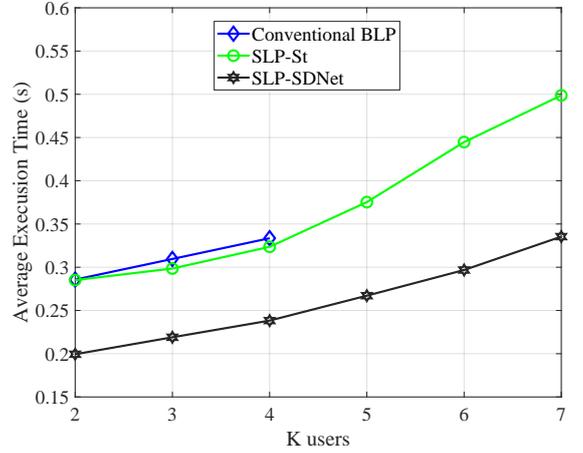


Fig. 4: Comparison of average execution time per sample averaged over 2000 test samples for Conventional BLP, SLP-St optimization-based and SLP-SDNet schemes.

By letting $\mathbf{v}_r = \mathbf{I}_{2N} - 2\gamma^{[r]}$, $\mathbf{b}_r = \lambda^{[r]}\Upsilon_i\Omega$ and $\mathbf{\Pi}_r = \text{prox}_{\gamma^{[r]}\mu^{[r]}\mathcal{B}}$, the r -layer network $\mathcal{L}^{[r-1]} \circ \dots \circ \mathcal{L}^{[0]}$ will correspond to the following

$$\mathbf{\Pi}_{R-1} \circ (\mathbf{v}_{R-1} + \mathbf{b}_{R-1}) \circ \dots \circ \mathbf{\Pi}_0 \circ (\mathbf{v}_0 + \mathbf{b}_0) \quad \forall 0 \leq r \leq R-1 \in \text{R-layers}, \quad (31)$$

where $[\mathbf{v}_r]_{0 \leq r \leq R-1}$ and $[\mathbf{b}_r]_{0 \leq r \leq R-1}$ are described as weight and bias parameters respectively. The identity square matrix is defined as $\mathbf{I}_{2N} \in \mathbb{R}^{2(N \times N)}$. The nonlinear activation functions are defined by $[\mathbf{\Pi}_r]_{0 \leq r \leq R-1}$ and can be obtained from the proximal operator. Furthermore, $\mathbf{\Pi}_r$ can be expressed as sum of a bias and a proximal activation operator.

D. SLP-SDNet Training and Testing

The SLP-SDNet has two modules; the parameter module and the auxiliary module. The parameter module consists of three structures associated with Lagrangian multipliers (equality and inequality constraints) and the

training step-size. The proximity barrier function is related to the inequality constraint and forms the parameter module. It is constructed with one convolutional layer, an average pooling layer, a fully connected layer, and a softPlus layer so that the output is constrained to a positive real value. The parameter update module contains r -th blocks and is trained block-wise for l -th number of iterations. Similarly, the auxiliary unit is trained for k -th iterations. It is important to note that the number of training iterations of the parameter update module may not necessarily be equal to that of the auxiliary unit. We train the parameter update unit with 15 iterations and the auxiliary unit for 10 iterations. During the inference, a feed-forward pass is performed over the whole layers using the learned Lagrangian multipliers to calculate the precoding vector using (28). The trained model is run over different SINR values to output the optimal precoding matrix.

IV. RESULTS AND DISCUSSION

A. Simulation Setup

We consider a downlink scenario, where the BS has four antennas ($N = 4$) that serve K single users, assuming a perfect known CSI. We generate 50,000 training samples and 2000 test samples of the channel coefficients randomly drawn from a normal distribution with zero mean and unit variance using (5). The datasets are normalized by the data symbol so that data entries are within the nominal range. The transmit data symbols are modulated using a QPSK and 8PSK modulation schemes; and the SINR is randomly generated from uniform distribution $\Gamma_{\text{train}} \sim \mathcal{U}(\Gamma_{\text{low}}, \Gamma_{\text{high}})$. A stochastic gradient descent algorithm with Adam optimizer is used to minimize the Lagrangian function (loss function). For every training iteration, the learning rate is reduced by $\beta = 0.65$ to help the learning algorithm converge faster. The implementation is done on Pytorch 1.7.1 and Python 3.7.8 on a computer with the following specifications: Intel(R) Core (TM) i7-6700 CPU Core, 32.0GB RAM.

B. Performance Evaluation of SLP-SDNet

We consider a SLP-SDNet for strict phase angle rotation problems (29). Our proposed unsupervised learning framework's performance is evaluated against the benchmark precoding designs in [4], [11]. We compare the average transmit power of the conventional BLP approach (2), the SLP-based problems (4), the proposed SLP learning-based precoding scheme based on Algorithm 2. Fig. 3 shows that the SLP-SDNet gives less transmit power than the conventional BLP scheme because, for a 4×4 system, there is inadequately available transmit power at the BS. It is also essential to note that the transmit power given by an SLP-SDNet is the same as for an SLP optimization-based solution at SINR values below 30dB. However, the transmit power increases by 8% for an SLP-SDNet solution over SLP optimization-based approach at SINR greater than 30dB.

C. Complexity Evaluation

For a fair comparison, we measure the complexities of our proposals and the benchmark optimization-based precoding schemes in terms of the optimization algorithms' average execution time, as shown in Figs. 4. We observe that the average execution time of the SLP-SDNet scheme per symbol averaged over 2000 test samples offers $2 \times$ decrease in execution time per data symbol because the predominant operations in SLP-SDNet during online training are matrix-matrix or vector-matrix convolution. This shows that the proposed unsupervised learning-based precoding scheme offers a desirable trade-off between performance and computational complexity.

V. CONCLUSION

This paper proposes a fast unsupervised learning-based precoding framework for a multi-user downlink MISO system. The proposed learning technique exploits the constructive interference for the power minimization problem so that for given QoS constraints, the transmit power available for transmission is minimized. We use domain

knowledge to develop an unsupervised learning architecture by unfolding the proximal interior point method barrier 'log function. Proximal barrier function for strict phase rotation is derived based on the nature and characteristics of the inequality constraints.

REFERENCES

- [1] C. Masouros, "Harvesting signal power from constructive interference in multiuser downlinks," in *Wireless Information and Power Transfer: A New Paradigm for Green Communications*. Springer, 2018, pp. 87–122.
- [2] C. Masouros and E. Alsusa, "A novel transmitter-based selective-precoding technique for ds/cdma systems," in *2007 IEEE International Conference on Communications*. IEEE, 2007, pp. 2829–2834.
- [3] C. Masouros and E. Alsusa, "Dynamic linear precoding for the exploitation of known interference in mimo broadcast systems," *IEEE Transactions on Wireless Communications*, vol. 8, no. 3, pp. 1396–1404, 2009.
- [4] C. Masouros and G. Zheng, "Exploiting known interference as green signal power for downlink beamforming optimization," *IEEE Transactions on Signal processing*, vol. 63, no. 14, pp. 3628–3640, 2015.
- [5] M. Alodeh, S. Chatzinotas, and B. Ottersten, "Constructive multiuser interference in symbol level precoding for the miso downlink channel," *IEEE Transactions on Signal processing*, vol. 63, no. 9, pp. 2239–2252, 2015.
- [6] A. Li and C. Masouros, "Interference exploitation precoding made practical: Optimal closed-form solutions for psk modulations," *IEEE Transactions on Wireless Communications*, vol. 17, no. 11, pp. 7661–7676, 2018.
- [7] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37328–37348, 2018.
- [8] P. de Kerret and D. Gesbert, "Robust decentralized joint precoding using team deep neural network," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*. IEEE, 2018, pp. 1–5.
- [9] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, "Unsupervised learning-based fast beamforming design for downlink mimo," *IEEE Access*, vol. 7, pp. 7599–7605, 2018.
- [10] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of miso downlink beamforming," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1866–1880, 2019.
- [11] E. Björnson, M. Bengtsson, and B. Ottersten, "Optimal multiuser transmit beamforming: A difficult problem with a simple solution structure [lecture notes]," *IEEE Signal Processing Magazine*, vol. 31, no. 4, pp. 142–148, 2014.
- [12] C. Masouros, "Correlation rotation linear precoding for mimo broadcast communications," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 252–262, 2010.
- [13] C. Masouros, T. Ratnarajah, M. Sellathurai, C. Papadias, and A. Shukla, "Known interference in wireless communications: a limiting factor or a potential source of green signal power?" *IEEE Comms. Mag.*, vol. 51, no. 10, pp. 162–171, 2013.
- [14] A. Li, D. Spano, J. Krivochiza, S. Domouchtsidis, C. G. Tsinos, C. Masouros, S. Chatzinotas, Y. Li, B. Vucetic, and B. Ottersten, "A tutorial on interference exploitation via symbol-level precoding: Overview, state-of-the-art and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 796–839, 2020.
- [15] C. Bertocchi, E. Chouzenoux, M.-C. Corbineau, J.-C. Pesquet, and M. Prato, "Deep unfolding of a proximal interior point method for image restoration," *Inverse Problems*, vol. 36, no. 3, p. 034005, 2020.
- [16] R. Hauser, "Interior-point methods for inequality constrained optimization," 2007.
- [17] N. Pustelnik and L. Condat, "Proximity operator of a sum of functions; application to depth map estimation," *IEEE Signal Processing Letters*, vol. 24, no. 12, pp. 1827–1831, 2017.