

Learning-Based Symbol Level Precoding: A Memory-Efficient Unsupervised Learning Approach

Abdullahi Mohammad*, Christos Masouros* and Yiannis Andreopoulos*

*Department of Electronic and Electrical Engineering, University College London, WC1E 7JE UK
e-mail: (abdullahi.mohammad.16; c.masouros; i.andreopoulos)@ucl.ac.uk

Abstract—Symbol level precoding (SLP) has been proven to be an effective means of managing the interference in a multiuser downlink transmission and also enhancing the received signal power. This paper proposes an unsupervised-learning based SLP that applies to quantized deep neural networks (DNNs). Rather than simply training a DNN in a supervised mode, our proposal unfolds a power minimization SLP formulation in an imperfect channel scenario using the interior point method (IPM) proximal ‘log’ barrier function. We use binary and ternary quantizations to compress the DNN’s weight values. The results show significant memory savings for our proposals compared to the existing full-precision SLP-DNet with significant model compression of $\sim 21\times$ and $\sim 13\times$ for both binary DNN-based SLP (RSLP-BDNet) and ternary DNN-based SLP (RSLP-TDNets), respectively.

I. INTRODUCTION

Multiple-input-multiple-output (MIMO) is one of the essential techniques for fifth-generation (5G) wireless communication and has recently attracted a myriad of research. Conventional block-level precoding (BLP) methods that exploit the spatial multiplexing of the multi-user MIMO system, is employed at the base station (BS) to mitigate the multi-user interference (MUI) have proven to be computationally efficient than the optimal dirty paper coding (DPC) but suffer performance deterioration [1]. The method for classifying instantaneous interference into constructive and destructive was first investigated in [2]. The suboptimal precoding methods that exploit constructive interference (CI) were first introduced in [3].

The optimization-based precoding methods are intriguing because of their propensity to deliver various performance targets. The first optimization-based CI precoding was proposed in the context of vector perturbation strategy in [4]. Additional performance is achieved by applying the precoding coefficients on a symbol-by-symbol basis termed symbol level precoding (SLP) that exploits the multiuser interference via CI with the known channel state information (CSI) and converts it into beneficial power at the receiver. Such precoding strategies have been extensively studied over the last five years. [5]–[8]. More recently, a closed-form optimal precoding design via CI exploitation in the MISO downlink for optimization with both strict and relaxed phase rotations was proposed [9]. Running CI-based precoding methods online on a symbol-by-symbol basis can be computationally taxing despite the outstanding performance they offered.

With relatively low inference complexity, deep learning (DL)-based precoding designs have recently been proposed for MU-MIMO downlink transmission. [10]–[13]. However, learning-based strategies for wireless physical layer designs use DL model as a function approximator in a supervised learning mode, which requires labeled training data. This labeled training data is obtained from the analytical solution of the optimization problem, whose accuracy is bounded by the optimization algorithm. [14].

The DL model contains millions of trained parameters, which are often stored in a 32-bit floating-point (FP32) numerical format. However, this renders the trained DL model computationally inefficient during inference and challenging to deploy to the edged devices due to resource limitations (memory and power constraints). Scalable deep neural network (DNN) models, whose weights are expressed in lower numerical precision, have been recently attracted a lot of attention [15]–[17]. While this idea is not new in computer vision, it has not been adequately explored within the wireless communications domain. A scalable DDN-based MIMO receiver design, where the insignificant neurons were systematically attenuated or removed via monotonically decreasing functions to reduce the network’s size, was first introduced in the work of [18], [19]. However, in this work, we propose an unsupervised, low precision DNN-based SLP framework, where DNN weights are constrained to binary values based on the initial work on scalable learning-based SLP designs [20].

II. SYSTEM MODEL AND SYMBOL LEVEL PRECODING

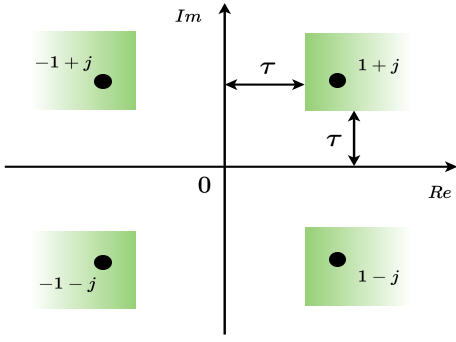
A. System Model

Consider single cell MU-MISO downlink transmission scenario where K single-antenna users are served by an by M BS antennas. Assuming a flat-fading Rayleigh channel $\mathbf{h}_i \in \mathbb{C}^{N_t \times 1}$, the received signal at the i -th user is expressed as

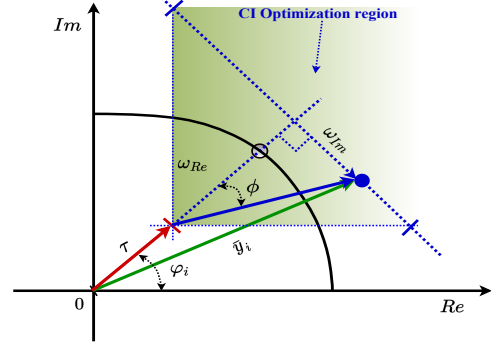
$$y_i = \mathbf{h}_i^T \sum_{k=1}^K \mathbf{w}_k d_k + n_i, \quad (1)$$

where \mathbf{h}_i , \mathbf{w}_i , d_i and n_i represent the channel vector, precoding vector, data symbol, phase rotation and additive white Gaussian noise for the i -th user.

Instantaneous interference is categorized into constructive and destructive [21]. As an illustration, Fig. 1(a) shows the QPSK constellations diagram, where the CI area is indicated by the green region with respect to the minimum distance (τ) from the decision boundaries, allowing the interfering signals to be added constructively with the symbol of interest via precoding vectors. The generic geometrical representation of the CI in Fig. 1(b) shows that if the maximum angle shift ($\phi = 0$) in the CI region is zero, the interfering signals overlap completely on the symbol of interest. Hence, the problem becomes a strict phase angle optimization. However, the strict phase formulation is not appealing because it leads to an additional transmission power compared to the corresponding relaxed counterpart [21]. For simplicity, we define the following variables, $\hat{\mathbf{h}}_i = \mathbf{h}_i \sum_{k=1}^K e^{j(\phi_k - \phi_i)} \in \mathbb{C}^{M \times 1}$, $\mathbf{w} = \sum_{k=1}^K \mathbf{w}_k \in \mathbb{C}^{M \times 1}$, $\hat{\mathbf{h}}_{Ri} = \Re\{\hat{\mathbf{h}}_i\}$, $\hat{\mathbf{h}}_{Ii} = \Im\{\hat{\mathbf{h}}_i\}$, $\mathbf{w}_R = \Re\{\mathbf{w}\}$ and $\mathbf{w}_I = \Im\{\mathbf{w}\}$. Similarly, we also let $\Psi = [\hat{\mathbf{h}}_{Ri} \ \hat{\mathbf{h}}_{Ii}]^T$, $\mathbf{w}_1 = [\mathbf{w}_R \ \mathbf{w}_I]^T$, where $\Theta = \begin{bmatrix} \mathbf{O}_M & -\mathbf{I}_M \\ \mathbf{I}_M & \mathbf{O}_M \end{bmatrix} \in \mathbb{R}^{2M \times 2M}$.



(a) Constructive interference constellation



(b) Generic geometrical optimization regions for interference exploitation

Fig. 1: Graphical representation of Interference Exploitation for Precoding design in QPSK [5]

B. Conventional Robust Precoding

In practice, the exact channel state information (CSI) is often unknown; only the estimate is

$$\hat{\mathbf{h}}_i = \mathbf{h}_i + \mathbf{e}_i \quad \forall k, \quad (2)$$

where \mathbf{h}_i is the known CSI estimates at the BS and \mathbf{e}_i denotes the channel error. Given this, the robust traditional recording for the downlink MU-MISO power minimization optimization is [22]

$$\begin{aligned} \min_{\{\bar{\mathbf{w}}_i \geq 0, d_i \geq 0\}} \quad & \sum_{i=1}^K \text{trace}(\bar{\mathbf{W}}_i) \\ \text{s.t.} \quad & \begin{bmatrix} \hat{\mathbf{h}}_i^* \mathbf{T}_i \hat{\mathbf{h}}_i^T - \gamma_i n_0 - d_i \delta_i^2 & \hat{\mathbf{h}}_i^* \mathbf{T}_i \\ \mathbf{T}_i \hat{\mathbf{h}}_i^T & \mathbf{T}_i + \delta_i^2 \mathbf{I} \end{bmatrix} \succeq 0 \quad \forall k \end{aligned} \quad (3)$$

where $\mathbf{T}_i \triangleq \bar{\mathbf{W}}_i - \Gamma_i \sum_{k=1, k \neq i}^K \bar{\mathbf{W}}_k \quad \forall k$ and $\bar{\mathbf{W}}_i = \mathbf{w}_i \mathbf{w}_i^\dagger$.

C. Robust SLP optimization-Based Power Minimization

The multi-cast CI formulation of the power minimization problem for the worst-case scenario is given by [22]

$$\begin{aligned} \min_{\{\mathbf{w}\}} \quad & \|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & \left| \Im\{\hat{\mathbf{h}}_i^T \mathbf{w}\} \right| - (\Re\{\hat{\mathbf{h}}_i^T \mathbf{w}\} - \sqrt{\Gamma_i n_0}) \tan \phi \leq 0, \\ & \forall \|\hat{\mathbf{e}}_i\|^2 \leq \delta_i^2, \quad \forall i. \end{aligned} \quad (4)$$

For simplicity, we drop the subscripts in (4) and slit the real and imaginary parts of the constraint into two separate constraints real-valued functions as follows

$$\Psi^T \mathbf{w}_1 - \Psi^T \mathbf{w}_2 \tan \phi + \delta \|\mathbf{w}_1 - \mathbf{w}_2 \tan \phi\|_2 + \sqrt{\Gamma n_0} \tan \phi \leq 0, \quad (5)$$

$$-\Psi^T \mathbf{w}_1 - \Psi^T \mathbf{w}_2 \tan \phi + \delta \|\mathbf{w}_1 + \mathbf{w}_2 \tan \phi\|_2 + \sqrt{\Gamma n_0} \tan \phi \leq 0, \quad (6)$$

where $\hat{\mathbf{e}} \triangleq [\mathbf{e}_R \quad \mathbf{e}_I]^T$ and $\hat{\mathbf{h}} = \mathbf{h}_R + j\mathbf{h}_I + \mathbf{e}_R + j\mathbf{e}_I$. Then (5) becomes

$$\begin{aligned} \min_{\{\mathbf{w}_1, \mathbf{w}_2\}} \quad & \|\mathbf{w}_1\|_2^2 \\ \text{s.t.} \quad & \text{Constraints (5) and (6)}, \quad \forall i \\ & \text{where } \mathbf{w}_1 = \mathbf{\Pi} \mathbf{w}_2. \end{aligned} \quad (7)$$

III. ROBUST LOW-BIT DNN-BASED SLP FOR POWER MINIMIZATION PROBLEM

This section presents robust binary and ternary DNN-based SLP models (RSLP-BDNet and RSLP-TDNet). We begin first by formulating the full-precision DNN-based SLP counterpart (RSLP-DNet). From (7), we define the following: $\mathbf{Q}_1 =$

$(\Theta - \tan \phi \mathbf{I})$ and $\mathbf{Q}_2 = (\Theta + \tan \phi \mathbf{I})$. Therefore, constraints (5) and (6) can be written as

$$\mathbf{\Lambda}^T \mathbf{Q}_1 \mathbf{w}_2 + \delta \|\mathbf{Q}_1 \mathbf{w}_2\|_2 + \sqrt{\Gamma n_0} \tan \phi \leq 0 \quad (8)$$

$$\mathbf{\Lambda}^T \mathbf{Q}_2 \mathbf{w}_2 + \delta \|\mathbf{Q}_2 \mathbf{w}_2\|_2 + \sqrt{\Gamma n_0} \tan \phi \leq 0 \quad (9)$$

Following this, (7) is thus

$$\begin{aligned} \min_{\{\mathbf{w}_2\}} \quad & \|\mathbf{w}_2\|_2^2 \\ \text{s.t.} \quad & \text{Constraints (8) and (9)}, \quad \forall i. \end{aligned} \quad (10)$$

A. SLP using Interior Point Method

We begin by unfolding (10) using an IPM 'log' barrier function and transform it to its equivalent unconstrained sequence of sub-problems per user [14]

$$\min_{\mathbf{w}_2 \in \mathbb{R}^{2M \times 1}} f(\mathbf{w}_2) + vB(\mathbf{w}_2), \quad (11)$$

where $B(\cdot) \triangleq -\sum \ln(\cdot)$ is the logarithmic barrier function, v is the Lagrangian multiplier for inequality constraints. The learning framework is derived by defining a proximity operator of (11) [14]

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

where $\gamma \in \{0, +\infty\}$ is the training step-size, \mathbf{w}_0 is the initial precoding vector and v is the Lagrange multiplier of the inequality constraint.

1) *Euclidean Constraint*: It can be observed that the constraints (8) and (9) are bounded by the ℓ_2 -norm of the form

$$\mathcal{C} = \{\mathbf{z} \in \mathbb{R}^n \mid \|\mathbf{z} - \mathbf{x}\|_2 \leq \alpha\}, \quad (13)$$

where $\alpha > 0$ and $\mathbf{x} \in \mathbb{R}^n$. The 'log barrier function is given by [23]

$$B(\mathbf{z}) = \begin{cases} -\ln(\alpha - \|\mathbf{z} - \mathbf{x}\|_2), & \text{if } \|\mathbf{z} - \mathbf{x}\|_2 < \alpha \\ +\infty, & \text{otherwise.} \end{cases} \quad (14)$$

Based on (14), the barrier function for (8) is expressed at the bottom of this page. Similar expression can also be written for (9). Therefore, the effective barrier function for the two constraints is the sum of the individual barrier functions $B(\mathbf{w}_2) = B_1(\mathbf{w}_2) + B_2(\mathbf{w}_2)$.

It can be seen that the upper bounds of the two constraints (8) and (9) are zeros. Therefore, combining (8) and (9), we obtain

$$(\delta^2 - \Psi^T \Psi) \mathbf{G} \|\mathbf{w}_2\|_2^2 + 4\Psi^T \mathbf{w}_2 \tan \phi \sqrt{\Gamma n_0} \leq 2\Gamma n_0 \tan^2 \phi \quad (16)$$

where $\mathbf{G} = \mathbf{Q}_1^T \mathbf{Q}_1 + \mathbf{Q}_2^T \mathbf{Q}_2$. Consequently, for each \mathbf{w}_2 , the proximity operator of the barrier $\gamma v B$ is

$$\Psi(\mathbf{w}_2, \gamma, v) = \frac{2\Gamma n_0 \tan^2 \phi - \chi(\mathbf{w}_2, \gamma, v)^2}{2\Gamma n_0 \tan^2 \phi - \chi(\mathbf{w}_2, \gamma, v)^2 + 2\gamma v} \mathbf{w}_2 \quad (17)$$

where $\chi(\mathbf{w}_2, \gamma, v)$ is the analytical solution of the cubic equation [14]. The robust deep-unfolded model is derived according to the derivatives of (17) with respect to \mathbf{w}_2 , γ and v as follows

$$\mathcal{J}_\Psi |_{(\mathbf{w}_2)} = \frac{2\Gamma n_0 \tan^2 \phi - \|\Psi(\mathbf{w}_2, \gamma, v)\|_2^2}{2\Gamma n_0 \tan^2 \phi - \|\Psi(\mathbf{w}_2, \gamma, v)\|_2^2 + 2\gamma v} \times M(\mathbf{w}_2, v, \gamma), \quad (18)$$

$$\Delta_\Psi |_{(v)} = \frac{-2\gamma}{2\Gamma n_0 \tan^2 \phi - \|\Psi(\mathbf{w}_2, \gamma, v)\|_2^2 + 2\gamma v} \times M(\mathbf{w}_2, v, \gamma) (\Psi(\mathbf{w}_2, v, \gamma)), \quad (19)$$

$$\Delta_\Psi |_{(\gamma)} = \frac{-2v}{2\Gamma n_0 \tan^2 \phi - \|\Psi(\mathbf{w}_2, \gamma, v)\|_2^2 + 2\gamma v} \times M(\mathbf{w}_2, v, \gamma) (\Psi(\mathbf{w}_2, v, \gamma)), \quad (20)$$

where $M(\mathbf{w}_2, v, \gamma)$ is as defined in [14].

We use the proximity operator of the barrier to obtain the variable update function as follows

$$\mathbf{w}_2^{[r+1]} = \text{prox}_{\gamma^{[r]} v^{[r]} B_{\text{robust}}} \left(\mathbf{w}_2^{[r]} - \gamma^{[r]} \Delta f(\mathbf{w}_2^{[r]}, \lambda^{[r]}) \right) \quad (21)$$

where $f(\mathbf{w}_2^{[r]}, \lambda^{[r]}) = \|\mathbf{w}_1^{[r]}\|_2^2 + \lambda \mathbf{w}_2$. We define the update function \mathbb{D} as

$$\mathbb{D}(\mathbf{w}_2^{[r]}, \gamma^{[r]}, v^{[r]}, \lambda^{[r]}) = \text{prox}_{\gamma^{[r]} v^{[r]} B} \left(\mathbf{w}_2^{[r]} - \gamma^{[r]} \Delta f(\mathbf{w}_2^{[r]}, \lambda^{[r]}) \right), \quad (22)$$

and $\Delta = \frac{\partial f(\mathbf{w}_2^{[r]}, \lambda^{[r]})}{\partial \mathbf{w}_2^{[r]}}$.

2) *Loss Function*: The training loss function is the Lagrangian function of (10) obtained as

$$\begin{aligned} \mathcal{L}(\mathbf{w}_2, \mathbf{v}_1, \mathbf{v}_2) &= \frac{1}{N} \sum_{i=1}^N \|\mathbf{w}_2\|_2^2 \\ &+ \frac{\mathbf{v}_1}{N} \sum_{i=1}^N \left[\delta^2 \|\mathbf{Q}_1 \mathbf{w}_2\|_2^2 - (\sqrt{\Gamma n_0} \tan \phi - \Psi^T \mathbf{Q}_1 \mathbf{w}_2)^2 \right] \\ &+ \frac{\mathbf{v}_2}{N} \sum_{i=1}^N \left[\delta^2 \|\mathbf{Q}_2 \mathbf{w}_2\|_2^2 - (\sqrt{\Gamma n_0} \tan \phi - \Psi^T \mathbf{Q}_2 \mathbf{w}_2)^2 \right] \\ &+ \frac{\mu}{NL} \sum_{i=1}^N \sum_{i=1}^L \|\Omega_i\|_2^2, \quad (23) \end{aligned}$$

where \mathbf{v}_1 and \mathbf{v}_2 are the Lagrangian multipliers of the two inequality constraints. The $\Omega_i(s)$ are the trainable parameters of the i -th layers and $\mu > 0$ is the penalty parameter that controls the bias and variance of the learnable parameters. Note that are associated with the barrier term and are randomly initialized from a uniform distribution. The model is trained in an unsupervised mode to update \mathbf{v} , λ , γ and \mathbf{w}_2 such that the

loss function is minimized. By minimizing (23) with respect to \mathbf{w}_2 , we obtain the optimal precoder

$$\begin{aligned} (1 + (\mathbf{v}_1 \|\mathbf{Q}_1\|_2^2 + \mathbf{v}_2 \|\mathbf{Q}_2\|_2^2) (\delta^2 - \Psi^T \Psi)) \mathbf{w}_2 = \\ - (\mathbf{v}_1 \mathbf{Q}_1 + \mathbf{v}_2 \mathbf{Q}_2) \Psi \sqrt{\Gamma w_0 \tan \phi}. \quad (24) \end{aligned}$$

For clarity, we let $[\|\mathbf{Q}_1\|_2^2 \ \|\mathbf{Q}_2\|_2^2] = \bar{\mathbf{Q}}_n$, $[\mathbf{Q}_1 \ \mathbf{Q}_2] = \mathbf{Q}'$ and $[\mathbf{v}_1 \ \mathbf{v}_2] = \bar{\mathbf{v}}$. Hence, (24) is reduced to

$$(\mathbf{I}_{2M} + \bar{\mathbf{Q}}_n \bar{\mathbf{v}}^T (\delta^2 - \Psi^T \Psi)) \mathbf{w}_2 = -\Psi \mathbf{Q}' \bar{\mathbf{v}}^T \sqrt{\Gamma n_0} \tan \phi \quad (25)$$

The optimal transmit precoder is finally obtained as

$$\mathbf{w}_2 = -\Psi \mathbf{Q}' \bar{\mathbf{v}}^T \mathbf{P}^{-1} \sqrt{\Gamma n_0} \tan \phi, \quad (26)$$

where $\mathbf{P} = (\mathbf{I}_{2M} + \bar{\mathbf{Q}}_n \bar{\mathbf{v}}^T (\delta^2 \mathbf{I}_{2M} - \Psi^T \Psi))$.

B. RSLP-DNet and the Generic NN Architecture

Intuitively, we can form NN cascade layers from (21) as follows

$$\mathbf{w}_2^{[l+1]} = \text{prox}_{\gamma^{[l]} v^{[l]} B} \left[(\mathbf{I}_{2M} - 2\gamma^{[l]}) \mathbf{w}_2^{[r]} + \gamma^{[l]} \lambda^{[l]} \mathbf{1}^T \right], \quad (27)$$

where $\mathbf{1} \in \mathbb{R}^{1 \times 2M}$ is a vector of ones. By letting $\mathbf{W}_l = \mathbf{I}_{2M} - 2\gamma^{[l]}$, $\mathbf{b}_l = \gamma^{[l]} \lambda^{[l]} \mathbf{1}^T$ and $\Xi_l = \text{prox}_{\gamma^{[l]} v^{[l]} B}$, the l -layer network $\mathcal{L}^{[l-1]} \dots \mathcal{L}^{[0]}$ will correspond to the following

$$\Xi_0(\mathbf{W}_0 + \mathbf{b}_0), \dots, \Xi_l(\mathbf{W}_l + \mathbf{b}_l), \quad (28)$$

\mathbf{W}_l and \mathbf{b}_l present weight and bias parameters, respectively, and Ξ_l describes the nonlinear activation functions. Finally, based on this formulation, RSLP-DNet is built as shown in Fig. 2 and its internal DNN designs are summarised in Tables I and II.

TABLE I: Proximity Barrier Term DNN Design

Layer	Parameter, kernel size = 3×3
Input Layer	Input size (B, 1, $2M$, K)
Layer 1: Convolutional	Size (B, 20, $2M$, K^2); zero padding
Layer 2: Average Pooling	Size ((1, 1), stride = (1, 1))
Layer 3: Activation	Soft-Plus
Layer 4: Flat	Size (B \times 40 \times K^2)
Layer : Fully-connected	Size (B \times 40 \times K^2 , 1)
Layer 5: Activation	Soft-Plus function

TABLE II: A PPU DNN Design

Layer	Parameter, kernel size = 3×3
Input Layer	Input size (B, 1, $2M$, K)
Layer 1: Convolutional	Size (B, 16, $2M$, K), dilation = 1 and unit padding
Layer 2: Batch Normalization	eps = 10^{-6} , momentum = 0.1
Layer 3: Activation	PReLU/k-bit function
Layer 4: Convolutional	Size (B, 8, K , $2KM$), dilation = 1 and unit padding
Layer 5: Batch Normalization	eps = 10^{-6} , momentum = 0.1
Layer 6: Activation	PReLU/k-bit function
Layer 7: Convolutional	Size (B, 1, $2KM$, 1), dilation = 1 and unit padding

$$B_1(\mathbf{w}_2) = \begin{cases} -\ln(-\sqrt{\Gamma n_0} \tan \phi - (\Psi^T \mathbf{Q}_1 \mathbf{w}_2 + \delta \|\mathbf{Q}_1 \mathbf{w}_2\|_2)), & \text{if } \Psi^T \mathbf{Q}_1 \mathbf{w}_2 + \delta \|\mathbf{Q}_1 \mathbf{w}_2\|_2 < -\sqrt{\Gamma n_0} \tan \phi \\ +\infty, & \text{otherwise} \end{cases} \quad (15)$$

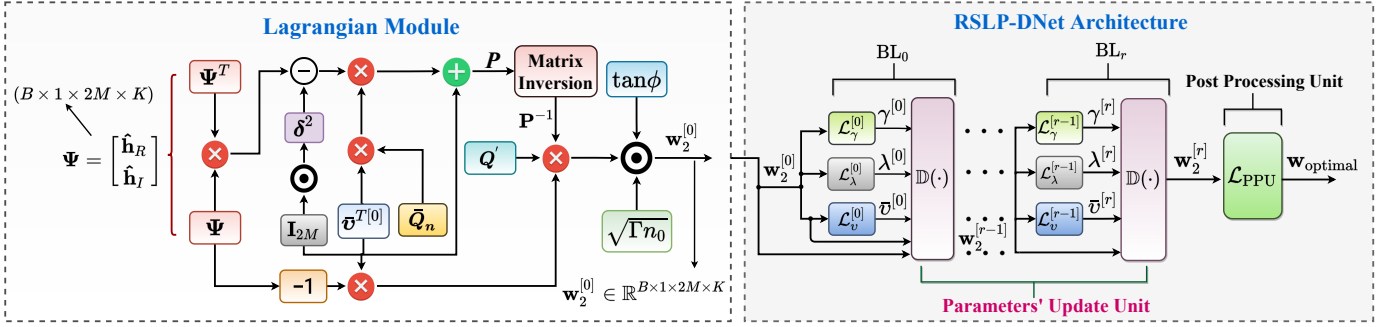


Fig. 2: Complete RSLP-DNet Architecture

C. Low-bit DNN Weights

Traditionally, DNN is designed with full-precision weights and activations. The quantization schemes have been proposed to design low-bit DNN models to address the problems of limited storage capacity and reduce hardware requirements during model deployment.

1) *1 Binary Weights*:: The real-valued weights are converted to $(\mathbf{W}_b \in \{+1, -1\}^n)$. A full-precision 32-bitweight matrix is binarized such that the weights \mathbf{W} are converted to their equivalent binary by the following function

$$\mathbf{W}_b = \text{sign}(\mathbf{W}) = \begin{cases} +1 & \text{if } \mathbf{W} \geq 0 \\ -1 & \text{otherwise,} \end{cases} \quad (29)$$

A more robust binarized network ‘‘BWN’’ is proposed in [15] as an extension of a straightforward binary network (Binary Connect) by introducing a real scaling factor $\beta \in \mathbb{R}^+$ such that $\mathbf{W} \approx \beta \mathbf{W}_b$ by solving an optimization problem

$$J(\mathbf{W}_b, \beta) = \underset{(\mathbf{W}_b, \beta)}{\text{argmin}} \|\mathbf{W} - \beta \mathbf{W}_b\|_2^2, \quad (30)$$

and this yields

$$\begin{aligned} \mathbf{W}_b^* &= \text{sign}(\mathbf{W}) \\ \beta^* &= \frac{1}{n} \|\mathbf{W}\|_1 \end{aligned} \quad (31)$$

2) *2 Weighted Ternary Weights*:: A ternary weighted network (TWN) is the one in which an extra 0 state is introduced into BWN to solve the following optimization problem

$$\begin{cases} \beta^*, \mathbf{W}_t^* = \underset{\beta, \mathbf{W}_t}{\text{argmin}} J(\beta, \mathbf{W}_t) = \|\mathbf{W} - \beta \mathbf{W}_t\|_2^2 \\ \text{s.t.} \quad \beta \geq 0, \mathbf{W}_t \in \{-1, 0, +1\}^n, \end{cases} \quad (32)$$

and as shown in [24], solving (32) gives

$$\mathbf{W}_t^* = \begin{cases} +1 & , \text{ if } \mathbf{W} > \rho \\ 0 & , \text{ if } |\mathbf{W}| \leq \rho \\ -1 & , \text{ if } \mathbf{W} < -\rho, \end{cases} \quad (33)$$

where $\rho = \frac{0.7}{n} \sum_{i=1}^n |\mathbf{W}|$ and $\beta^* = \frac{1}{\mathbf{I}_\rho} \sum_{i \in \mathbf{I}_\rho} |\mathbf{W}|$,

$\mathbf{I}_\rho = \{|\mathbf{W}| > \rho\}$ is the cardinality of set \mathbf{I}_ρ . As an illustration, Fig. 3 depicts how the weight matrices are quantised based on (31) and (33).

D. RSLP-BDNet Training and Inference

The RSLP-BDNet has two central units; the parameter update unit (PUU) and the post-processing unit (PPU). The parameter unit has three core components; \mathbf{v} (associated with the barrier term), λ and γ that are wired across the network (see Fig. 2). The barrier term is formed with one convolutional layer, an average pooling layer, a fully connected layer, and a softPlus layer to satisfy the positive inequality constraint. The PUU has

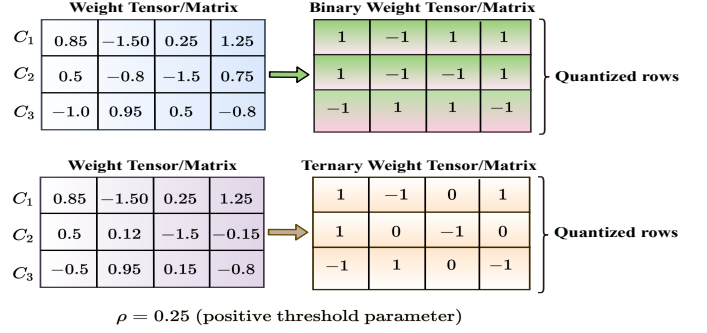


Fig. 3: Binary and Ternary DNN weights

r -th blocks, each representing a layer, and is trained block-wise for l -th iterations. Similarly, the PPU is made up three convolutional layers with batch normalisation layers in between them except for the last layer, and is trained for k -th iterations. The number of training iterations for the PUU may not necessarily be the same as that of the PPU. We train the PUU unit for 15 iterations and the PPU 10 iterations with Adam optimizer [25]. We also adjust the learning rate by a factor $\vartheta \in \mathbb{R}^+$ for every training step to enhance the training convergence. Because the learning is done unsupervised, we use regularised Lagrangian function as a loss function. During the inference, forward pass is performed over the entire layers with the learned Lagrangian multipliers to compute the precoding vector using (26). The inference is performed with different SINR values to calculate the required optimal precoding matrix.

IV. SIMULATION, RESULTS AND DISCUSSION

A. Simulation Set-up

We assume a single cell in a downlink scenario where the BS having four antennas ($M = 4$) serves K , single users. We generate 50000 training and 2000 test samples of channel coefficients, respectively. The transmit symbols are modulated using a QPSK modulation. The training SINR is randomly generated from uniform distribution $\Gamma_{\text{train}} \sim \mathcal{U}(\Gamma_{\text{low}}, \Gamma_{\text{high}})$ to allow training over wide range of SINR values. parametric rectified linear unit (**PReLU**) activation function is used in RSLP-DNet instead of the traditional **ReLU** function to mitigate the effect of dying gradient due to the saturation of neurons. After every iteration, the learning rate is reduced by a factor $\alpha = 0.65$ to facilitate learning convergence. The simulation parameters are summarized in Table 1. We implement the models in 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 of RAM.

TABLE III: Simulation settings

Parameters	Values
Training Samples	50000
Batch Size (B)	200
Test Samples	2000
Training SINR range	0.0dB - 45.0dB
Test SINR range (i -th user SINR)	0.0dB - 35.0dB
Initial Learning Rate η	0.001
Learning Rate decay factor ϑ	0.65
Number of blocks in the PUU	$B_r = 2$
Training Iterations for each block of the PUU	15
Training iterations for PPU	10

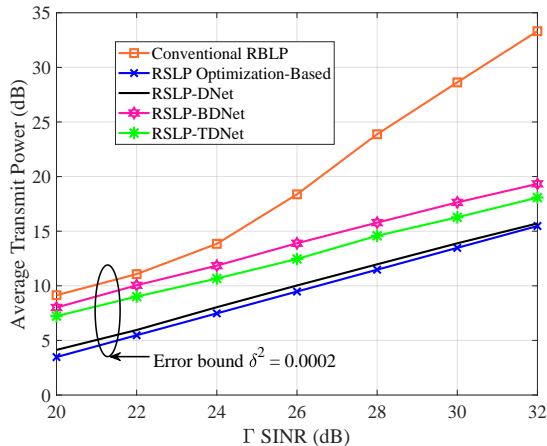


Fig. 4: Transmit Power vs SINR averaged over 2000 test samples for robust conventional RBLP, RSLP optimization-based, binary and ternary DNN-based SLP solutions under $M = 4$, $K = 4$ and $\delta^2 = 0.0002$

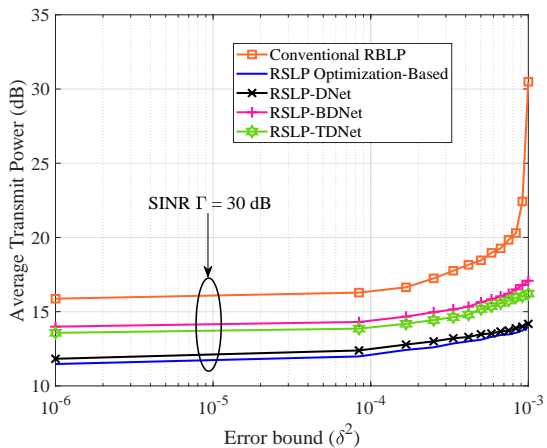


Fig. 5: Transmit Power vs Error-bound for Conventional RBLP, robust RSLP optimization-based, binary and ternary DNN-based SLP solutions under $M = 4$, $K = 4$

B. Performance Evaluation of and SLP-DNet RSLP-DNet

In this subsection, we consider a full-precision RSLP-DNet and its quantized counterparts (RSLP-BDNet and RSLP-TDNet). We use 4×4 MISO system with CSI error bounds $\delta^2 = 10^{-4}$, and QPSK modulation scheme. We compare the average transmit power of RSLP-BDNet and RSLP-TDNet against R-SLP-DNet [14] robust SLP optimization-based [5] and conventional [22] BLP methods. Fig. 4 depicts how the average transmit power increases with the SNR thresholds.

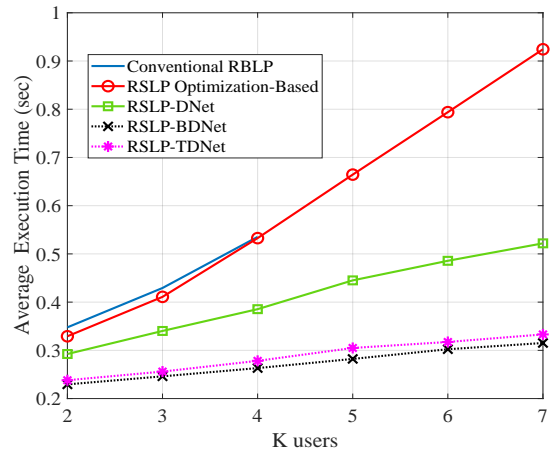


Fig. 6: Comparison of average execution time per sample averaged over 2000 test samples for Nonrobust and Robust precoding schemes, i.e, conventional BLP, SLP optimization-based and SLP learning-based models under $M = 4$, $K = 4$

The RSLP optimization-based is observed to show a significant power savings of more than 60% compared to the conventional RBPL solution. Similarly, the proposed RSLP-BDNet and RSLP-TDNet show considerable power savings of 40% – 58% against the conventional RBLP but lower lower than the RSLP optimization-based solution.

Furthermore, we study the effect of the CSI error bounds on the transmit power at 30dB. Fig. 5 depicts the variation of the transmit power with increasing CSI error bounds. A significant increase in transmit power can be observed where the channel uncertainty lies within the region of CSI error bounds of $\delta^2 = 10^{-3}$. Interestingly, like the RSLP optimization-based, by exploiting the CI, the proposed methods show a descent or moderate increase in transmit power.

C. Computational Complexity Evaluation

The computational costs of the proposed learning methods are obtained from the PUU and the feed-forward convolutions of the CNN that makes up the PPU. For the PUU, the dominant computation is in computing the proximal barrier functions, which requires computing the ‘log’ of the barrier and the shallow CNN structure. It can be seen that both RSLP optimization-based and the proposed schemes are feasible for all sets of M BS antenna and K mobile users. However, for conventional RBLP, the solution is only feasible for $M \geq K$.

Fig. 6(a) shows the average execution time of the proposed learning solutions per symbol averaged over 2000 test samples. We observe that while RSLP-DNet is faster than both RSLP optimization-based and conventional RBLP, RSLP-BDNet and RSLP-TDNet offer much less average execution time by $\sim 3\times$ and $\sim 5\times$ compared to RSLP-DNet, respectively. This because most of the MACs operations are replaced by binary bit-wise operations. RSLP-TDNet is slightly slower than RSLP-BDNet due to the additional ‘0’ binary state introduced in the former. Accordingly, we binarized DNN could offer significant training and inference accelerations while offering good trade-off between the performance and computational complexity.

The size of the DNN is often bounded by the available memory. Therefore, it is beneficial to estimate the memory requirements of the DNN at the inference. We examine and analyze the memory utilization of full-precision RSLP-DNet and its corresponding quantized versions at inference. We adopt the approach presented in [26] to calculate the inference memory utilization as the summation of 32-bit times the number of floating-point parameters and 1-bit times the number of binary

TABLE IV: Inference memory utilization comparison

Models	Weights	Activations	Memory usage (MB)	Memory saving
RSLP-DNet	$(32 - \text{bit}) \in \mathbb{R}$	$(32 - \text{bit}) \in \mathbb{R}$	0.1898	—
RSLP-BDNet	$\{-1, +1\}$	$\{-1, +1\}$	0.0089	21.33×
RSLP-TDNet	$\{-1, 0, +1\}$	$\{-1, +1\}$	0.0146	13×

parameters, expressed as $\frac{1}{32}W_b + W_f$, where W_b and W_f are the binary and floating-point weights, respectively. Table IV presents the summary of the inference memory requirements, where we observe that RSLP-BDNet and RSLP-TDNet provide considerable memory savings up to $\sim 21\times$ and $\sim 13\times$ compared to the RSLP-DNet, respectively.

V. CONCLUSION

This paper proposes robust binary and ternary unsupervised learning-based SLP designs for downlink power minimization optimization. The real-valued NN weights are converted to binary values, allowing the operations between the inputs and weights tensors to be performed in binary operations. We use domain knowledge to design unsupervised learning architectures by unfolding the proximal interior point method barrier 'log function for a relaxed phase rotation. The performance is within the range of 89% – 95% of the RSLP optimization-based solution with a substantial computational complexity reduction. Therefore, our proposals demonstrate an indispensable balance between the performance and the computational complexity involved.

REFERENCES

- [1] C. Windpassinger, R. F. Fischer, T. Vencel, and J. B. Huber, "Precoding in multiantenna and multiuser communications," *IEEE Transactions on Wireless Communications*, vol. 3, no. 4, pp. 1305–1316, 2004.
- [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, "Correlation rotation linear precoding for mimo broadcast communications," *IEEE Transactions on Signal Processing*, vol. 59, no. 1, pp. 252–262, 2010.
- [4] C. Masouros, M. Sellathurai, and T. Ratnarajah, "Vector perturbation based on symbol scaling for limited feedback mimo downlinks," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 562–571, 2014.
- [5] 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.
- [6] P. V. Amadori and C. Masouros, "Constant envelope precoding by interference exploitation in phase shift keying-modulated multiuser transmission," *IEEE Transactions on Wireless Communications*, vol. 16, no. 1, pp. 538–550, 2016.
- [7] D. Spano, M. Alodeh, S. Chatzinotas, and B. Ottersten, "Symbol-level precoding for the nonlinear multiuser mimo downlink channel," *IEEE Transactions on Signal Processing*, vol. 66, no. 5, pp. 1331–1345, 2017.
- [8] 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.
- [9] 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.
- [10] 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. 37 328–37 348, 2018.
- [11] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of mimo downlink beamforming," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1866–1880, 2019.
- [12] 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.
- [13] 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, 2019.
- [14] A. Mohammad, C. Masouros, and Y. Andreopoulos, "An unsupervised deep unfolding framework for robust symbol level precoding," *submitted to IEEE Transactions on Signal Processing*, 2021.
- [15] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 525–542.
- [16] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.
- [17] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1389–1397.
- [18] A. Mohammad, C. Masouros, and Y. Andreopoulos, "Complexity-scalable neural-network-based mimo detection with learnable weight scaling," *IEEE Transactions on Communications*, vol. 68, no. 10, pp. 6101–6113, 2020.
- [19] —, "Accelerated learning-based mimo detection through weighted neural network design," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.
- [20] —, "A memory-efficient learning framework for symbollevel precoding with quantized nn weights," *submitted to IEEE Transactions on Signal Processing*, 2021.
- [21] 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.
- [22] G. Zheng, K.-K. Wong, and T.-S. Ng, "Robust linear mimo in the downlink: A worst-case optimization with ellipsoidal uncertainty regions," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, pp. 1–15, 2008.
- [23] 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.
- [24] H. Alemdar, V. Leroy, A. Prost-Boucle, and F. Pétrot, "Ternary neural networks for resource-efficient ai applications," in *2017 international joint conference on neural networks (IJCNN)*. IEEE, 2017, pp. 2547–2554.
- [25] S. Shalev-Shwartz and S. Ben-David, *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [26] J. Bethge, H. Yang, M. Bornstein, and C. Meinel, "Binary-densenet: developing an architecture for binary neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.