Optimizing Wireless Networks with Deep Unfolding: Comparative Study on Two Deep Unfolding Mechanisms

Abuzar B. M. Adam *, *Member, IEEE*, Mohammed A. M. Elhassan, Elhadj Moustapha Diallo

Abstract—In this work, we conduct a comparative study on two deep unfolding mechanisms to efficiently perform power control in the next generation wireless networks. The power control problem is formulated as energy efficiency over multiple interference links. The problem is nonconvex. We employ fractional programming transformation to design two solutions for the problem. The first solution is a numerical solution while the second solution is a closed-form solution. Based on the first solution, we design a semi-unfolding deep learning model where we combine the domain knowledge of the wireless communications and the recent advances in the data-driven deep learning. Moreover, on the highlights of the closed-form solution, fully deep unfolded deep learning model is designed in which we fully leveraged the expressive closed-form power control solution and deep learning advances. In the simulation results, we compare the performance of the proposed deep learning models and the iterative solutions in terms of accuracy and inference speed to show their suitability for the real-time application in next generation networks

Index Terms—Deep unfolding, model-driven model, data-driven model, power allocation, sixth generation (6G), wireless networks, fractional programming.

I. INTRODUCTION

ALTHOUGH the fifth generation (5G) will be deployed in many countries and an intensive research work has been contributed to enhance it, many researchers and developers have pointed their attention into the six generation (6G) [1]. Visions for 6G have been laid down and an initiative has been proposed in the international telecommunication union (ITU) to establish the standards for the 2030 and beyond wireless technologies [2].

The deployment of 6G is expected to meet technical features including high data rate that has the peak rate of 20 Gbps for downlink and 10 Gbps for uplink. Quality of experience in which the user receives a minimum data rate at any given moment regardless of their location within the network. This can be measured in the cell edge to reflect the quality of the network design. Additionally, latency is expected to improve in 6G compared to that of 5G taking into consideration the complication due the large number of entities in the network. For energy efficiency (EE) and spectral efficiency, the power consumption is expected to decrease and EE can improve

100 times compared to 5G while the spectral efficiency is expected to be three times higher [1], [4]. In this context, EE in next-generation networks, including 5G and beyond (6G and future technologies), is a critical area of focus due to the exponentially increasing demand for data and the global emphasis on reducing energy consumption and carbon footprint [5]–[8].

To meet the above requirements, intelligent communications is envisioned as a key enabler for 6G networks [3]. Hence, deep learning (DL) is applied in to improve wireless communication by solving different problems such as channel estimation [9]–[14], optimal resource allocation [15]–[17]. Based on the mechanism of employing DL in solving wireless communications problems, the proposed methods can be categorized into main three categories: data-driven methods, model-driven methods, joint model-data-driven models. In data-driven methods, algorithms are designed to find the patterns and the connections which are hidden in the data.

Recently, model-driven or deep unfolding has been introduced as a key to enhance the performance of communication system. This mechanism combines both domain knowledge with learning ability of DL to overcome the problem of both traditional methods and DL-based methods (i.e., data-driven methods) [19]. Although, this mechanism has been mainly developed signal recovery [20], [21], some studies used it for precoding design [22], [23] and beamforming [24]–[27]. However, very few studies have been dedicated for model-based resource allocation. For instance, the study in [28] investigated unfolding of WMMSE using graph neural network. The idea was to perform energy-efficient power allocation. They have shown the advantages of the deep unfolding mechanism over the data-driven methods in terms of time and computation cost.

In this work, we investigate the employment of deep unfolding to design energy-efficient power control in multicell networks. We also consider modern neural networks architectures in addition to deep unfolding to design a hybrid model. Therefore, this study provides comparative aspects to conceive and incorporate the advantages of both mechanisms. The motivations of this work can be summarized in the following points:

- In the literature, although there are several studies on deep unfolding in precoding and signal recovery, but there are few studies dedicated for power control. We introduce this work to contribute in improving the work in this field.
- Although the study in [23] incorporated DNN to handle part of the problem, however there is a room to explore

*Corresponding author: Abuzar B. M. Adam (abuzar@cqupt.edu.cn)

A. B. M. Adam and E. M. Diallo are with School of Communications and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing, P. R. China, 400065.

M. A. M. Elhassan is with School of Mathematics and Computer Science, Zhejiang Normal University (e-mail: Mohammedac29@zjnu.edu.cn).

the advantages of both data-driven deep learning and model-driven to deep unfolding methods. Our goal is to incorporate the advantages of other DL models when designing deep unfolding based models.

- The emphasis on specific patterns in multivariate system is one of domain knowledge nonnegligible trait in communications systems. The impact of several parameters should be incorporated during the design of DL models which can enhance the prediction [18]. In this work, we want to show the advantage of multivariate patterns prediction in improving the model accuracy.

Our contributions can be summarized as follows:

- We investigate power control in multicell networks. The problem is formulated as EE maximization. The formulated problem is sum-of-ratios problem (SoRP). Obtaining the global optimal solution in the polynomial time for this problem is difficult. Therefore, we propose multidimensional fractional programming-based power control frameworks. The proposed frameworks convert the problem into a sequence of convex problems using auxiliary variables. To perform this conversion, the auxiliary variables are used to separate the rate function from the total power consumption and the signal from the interference. Based on the above procedure, we design two solutions to handle the problem.
- Despite that the proposed fractional programming solutions have the advantage of exploring most of the possible solutions in the solution space, However, they are time consuming and require several iterations to converge which make them unsuitable for real-time application. Hence, we resort to DL to handle the power control.
- First, we propose deep unfolding model based on the closed-from solution of power control i.e., we unfold the iterative solution into layers where we determine the learnable parameters from the unfolded solutions.
- We design another an end-to-end model on the highlights of the numerical solution and the deep unfolding while incorporating multiple attention sub-neural networks to compensate the loss. The attention blocks are used to perform multivariate prediction. Hence, we incorporate the advantages of modern DL architectures and data-driven mechanism.
- An ablation study is conducted to investigate the optimal design architectures for both models and the impact of each part of the proposed models.
- The simulation results have shown the efficiency of the proposed deep learning models in terms of accuracy and the inference speed.

The remainder of this paper is organized as follows: In Section II, we discuss the system model and problem formulation. In Section III, we introduce the fractional programming based energy-efficient power allocation where we investigate the numerical solution and closed-form solution. Section IV, investigates both model-driven designs energy efficient power control. In Section V, we present our simulation results and discussion. In Section VI, we give the conclusions and the future works.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a multi-cell interference network includes M base stations (BSs) serving K users. $\mathcal{M}$ and $\mathcal{K}$ represent the set of BS and users and denoted as $\mathcal{M} = \{1, 2, ..., M\}$ and $\mathcal{K} = \{1, 2, ..., K\}$. We assume that each BS is equipped with N antenna while each user is equipped with single antenna. The BS m transmits a signal to its associated users. Then, the received signal at the user k associated with the m is given as

$$y_{m,k} = \sqrt{\rho_{m,k}P_{max}}\mathbf{h}_{m,k}^H\mathbf{x}_m + \sum_{n\in\mathcal{M},n\neq m}\sum_{k'\in\mathcal{K}}\sqrt{\rho_{n,k'}P_{max}}\mathbf{h}_{n,k}^H\mathbf{x}_n + \mathbf{n}_{m,k},$$
(1)

where $\rho_{m,k}$ with $0 < \rho_{m,k} < 1$ is the power allocation coefficient of the user k associated with BS m, and P_{max} represents the maximum transmit power of the BS m, respectively. $\mathbf{x}_m$ is the transmit signal at the BS m where $E\left[\|\mathbf{x}_m\|^2\right] = 1$. $\mathbf{h}_{m,k} \in \mathbb{C}^{N\times 1}$ is the channel coefficient between the BS m the user k. The second term in (1) represents the intercell interference. $\mathbf{n}_{m,k}$ is the additive white Gaussian noise (AWGN) at the user k with variance σ^2. Hence, the signal-to-interference-plus-noise ratio (SINR) of the user k is expressed as,

$$\gamma_{m,k} = \frac{\left|\mathbf{h}_{m,k}^H\right|^2\rho_{m,k}P_{max}}{\sum_{j\neq k}\left|\mathbf{h}_{m,k}^H\right|^2\rho_{m,j}P_{max} + \sum_{n\in M,n\neq m}\sum_{k'\in K}\left|\mathbf{h}_{n,k}^H\right|^2\rho_{n,k'}P_{max} + \sigma^2_{m,k}},$$
(2)

In presence of intercell interference, the order of SINR is considered for power allocation and successive interference cancelation (SIC) in both BS and user[]. without loss of generality, we can assume that $\gamma_{m,1} > \gamma_{m,2} > ... > \gamma_{m,K}$. Using SINR information, power allocation coefficients are computed. After SIC is carried out, SINR of the user k associated with BS m can be expressed as,

$$\gamma_{m,k} = \frac{\left|\mathbf{h}_{m,k}^H\right|^2\rho_{m,k}P_{max}}{\sum_{j=1}^{k-1}\left|\mathbf{h}_{m,k}^H\right|^2\rho_{m,j}P_{max} + \sum_{n\in M,n\neq m}\sum_{k'\in K}\left|\mathbf{h}_{n,k}^H\right|^2\rho_{n,k'}P_{max} + \sigma^2_{m,k}},$$
(3)

And the achievable rate of the user k is given as

$$R_{m,k} = B\log_2\left(1 + \gamma_{m,k}\right)$$
(4)

Then, EE of the user k is given as follows

$$\eta_{m,k} = \frac{R_{m,k}}{\rho_{m,k}P_{max} + p_{k,c}}$$
(5)

where $p_{k,c}$ is the circuit power consumption.

We aim at maximize the weighted sum energy efficiency (WSEE) of the network. Unlike the global energy efficiency (GEE) where some links and users are favored, WSEE gives the opportunity to do the balance by explicitly prioritizing some links via different weight assignment [29]. The optimization problem can be formulated as,

$$\underset{\forall\rho_{m,k}}{\text{maximize}} \quad \sum_{m=1}^{M}\sum_{k=1}^{K}\omega_{m,k}\eta_{m,k}$$
(6)

$$\text{subject to} \quad \sum_{m\in\mathcal{K}}\rho_{m,k} \leq 1,$$
(6a)

where $\omega_{m,k}$ is the weight. The problem is nonconvex and sum of ratio problem (SoRP). Therefore, it is difficult to obtain the global solution with limited complexity [29], [30]. Nevertheless, we can recast it as a tractable convex problem as in the Section III. Toward the goal of improving the network performance, Section IV includes design of deep learning model to perform EE maximization in the real-time. This is achieved by projecting the traditional solution in Section III to a deep learning model.

III. Fractional Programming-based Energy-Efficient Power Allocation

In this section, we consider a low complexity algorithms to suboptimally solve problem (6). The proposed solutions is a quadratic transform based fractional programming. For similar problems as (6), some studies such as [29], [31] employed successive convex approximation (SCA) to obtain the lower bound. But the result of the this approximation on (6) will be sum of pseudo-concave functions which is not guaranteed to be pseudo-concave [18], [29]. To obtain a transformed function that fulfills the desired properties, we resort to the fractional programming method in [30]. Let us consider the following problem

$$\underset{x}{\text{maximize}} \quad \sum_{i=1}^{F} f\left(\frac{g_i(x)}{h_i(x)}\right)$$
$$\text{subject to} \quad x \in \mathcal{X}, \tag{7}$$

According to this method, the transformed function should satisfy the following technical conditions:

- The transformed objective function $\tilde{f}(x,\lambda)$ can be decoupled as $\tilde{f}(x,\lambda) = f_1(\lambda)A(g_i(x)) + B(h_i(x))f_2(\lambda)$ where λ is an auxiliary variable.
- The function $\tilde{f}(x,\lambda)$ is concave over λ when x is fixed.
- x^* maximizes $f(x)$ if and only if x^* with some λ^* maximize $\tilde{f}(x,\lambda)$.
- For $\lambda^* = \arg\max_y \tilde{f}(x,\lambda)$ given some x, then $\tilde{f}(x,\lambda^*) = f(x)$ for this x.

A. Numerical Solution for Power Allocation

On the highlights of the above conditions, we are going to transform the objective function in (6). First applying quadratic transform, we obtain the following equivalent problem

$$\underset{\forall \rho_{m,k}, y}{\text{maximize}} \quad \sum_{m=1}^{M}\sum_{k=1}^{K} \omega_{m,k}\left(2y(R_{m,k})^{\frac{1}{2}} - y^2\left(\rho_{m,k}P_{max} + p_{k,c}\right)\right) \tag{8}$$

$$\text{subject to} \quad \text{(6a)}$$
$$y_{m,k} \in \mathbb{R}, \tag{8a}$$

where y is the set of variables $\{y_{1,1}, ..., y_{M,K}\}$. The optimal value of $y_{m,k}^*$ is given when $\rho_{m,k}$ is fixed as

$$y_{m,k}^* = \frac{(R_{m,k})^{\frac{1}{2}}}{\rho_{m,k}P_{max} + p_{k,c}} \tag{9}$$

Problem (8) is equivalent to (6) and first technical condition is met. The term $(R_{m,k})^{\frac{1}{2}}$ is a non-decreasing and its concavity

can be established if $R_{m,k}$ is linear or concave while $y_{m,k}$ can be seen as a maximizer point. Obviously, when $\rho_{m,k}$ is fixed, the objective function is concave over y. However, due to the interference, $R_{m,k}$ is nonconcave. We employ multidimensional quadratic transform to linearize $R_{m,k}$. This is done by separating the signal from the interference. Thus, we have the transformed function in (10) on top of next page.

z represents the set $\{z_{1,1}, ..., z_{M,K}\}$. Where the optimal $z_{m,k}^*$ is given when other variables are fixed as follows

$$z_{m,k}^* = \frac{\left(\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,k}P_{max}\right)^{\frac{1}{2}}}{\sum_{j=1}^{k-1}\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,j}P_{max} + \sum_{n\in M, n\neq m}\sum_{k'\in K}\left|\mathbf{h}_{n,k}^H\right|^2 \rho_{n,k'}P_{max} + \sigma_{m,k}^2} \tag{11}$$

The objective function is concave over $z_{m,k}$ when the variables $\rho_{m,k}$ and $y_{m,k}$ are fixed. Hence, the equivalent problem can be written as,

$$\underset{\rho, y, z}{\text{maximize}} \quad \sum_{m=1}^{M}\sum_{k=1}^{K} \tilde{f}_{m,k}^{FP} \tag{12}$$
$$\text{subject to} \quad \text{(6a), (8a)}$$
$$z_{m,k} \in \mathbb{R}, \tag{12a}$$

where $\tilde{f}_{m,k}^{FP}$ is given as follows

$$\tilde{f}_{m,k}^{FP} = \omega_{m,k}\left(2y_{m,k}\left(\tilde{R}_{m,k}(\rho, z)\right)^{\frac{1}{2}} - y_{m,k}^2\left(\rho_{m,k}P_{max} + p_{k,c}\right)\right) \tag{13}$$

The problem in (12) is convex on $\rho_{m,k}$ when $z_{m,k}$ and $y_{m,k}$ are fixed and it can be solved using CVX [32]. Algorithm 1 illustrates the logical procedure to solve (12).

Algorithm 1 Proposed Iterative Numerical Solution for Problem (**??**)

Initialization: $\rho_{m,k}$, ϵ, $t = 0$.
1: **while** $|\tilde{f}_{m,k}^{FP}(t) - \tilde{f}_{m,k}^{FP}(t-1)| \geq \epsilon$ **do**
2:     Update $z_{m,k}$ according to (11).
3:     Update $y_{m,k}$ according to (9).
4:     Update $\rho_{m,k}$ by solving (12) with fixed $z_{m,k}$ and $y_{m,k}$.
5:     $t = t + 1$.
6: **end while**

B. Closed-form Solution for Power Allocation

Obtaining a closed-form solution for the power allocation is desirable to design a deep unfolding model. However, the solution of power allocation problem (12) can not be expressed in a closed-form. To circumvent this issue, implement the following steps.

First, we apply Lagrange dual transform [33] on the data rate function, we get (14) on top of next page.

$$\tilde{R}_{m,k}(\rho, z) = B\log_2\left(1 + 2z_{m,k}\left(\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,k}P_{max}\right)^{\frac{1}{2}} - z_{m,k}^2\left(\sum_{j=1}^{k-1}\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,j}P_{max} + \sum_{n\in M, n\neq m}\sum_{k'\in K}\left|\mathbf{h}_{n,k}^H\right|^2 \rho_{n,k'}P_{max} + \sigma_{m,k}^2\right)\right), \quad (10)$$

$$R_{m,k}(\rho, \gamma) = B\log_2(1 + \gamma_{m,k}) - B\gamma_{m,k} + \frac{B\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,k}P_{max}(1 + \gamma_{m,k})}{\sum_{j=1}^{k-1}\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,j}P_{max} + \sum_{n\in M, n\neq m}\sum_{k'\in K}\left|\mathbf{h}_{n,k}^H\right|^2 \rho_{n,k'}P_{max} + \sigma_{m,k}^2}, \quad (14)$$

$$\rho_{m,k} = \min\left\{\frac{\left[\left(z_{m,k}(1 + \gamma_{m,k})\left|\mathbf{h}_{m,k}^H\right|^2 P_{max}\right)^2 - y_{m,k}^2\left(\log_2(1 + \gamma_{m,k}) + \gamma_{m,k} + z_{m,k}^2\left(1 - I_{m,k} - \sigma_{m,k}^2\right)\right)\right]^2}{4y_{m,k}^4 z_{m,k}^2(1 + \gamma_{m,k})\left|\mathbf{h}_{m,k}^H\right|^2 P_{max}}, 1\right\}, \quad (19)$$

Next, applying multidimensional quadratic transform, $R_{m,k}(\rho, \gamma)$ can be written as

$$R_{m,k}(\rho, \gamma, z) = B\log_2(1 + \gamma_{m,k}) - B\gamma_{m,k} - z_{m,k}^2$$
$$+ 2z_{m,k}\sqrt{B\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,k}P_{max}(1 + \gamma_{m,k})}$$
$$- z_{m,k}^2\left(\begin{array}{l}\sum_{j=1}^{k-1}\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,j}P_{max} \\ + \sum_{n\in M, n\neq m}\sum_{k'\in K}\left|\mathbf{h}_{n,k}^H\right|^2 \rho_{n,k'}P_{max} + \sigma_{m,k}^2\end{array}\right), \quad (15)$$

where $z_{m,k}^*$ is given as follows

$$z_{m,k}^* = \frac{\sqrt{B\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,k}P_{max}(1 + \gamma_{m,k})}}{\sum_{j=1}^{k-1}\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,j}P_{max} + \sum_{n\in M, n\neq m}\sum_{k'\in K}\left|\mathbf{h}_{n,k}^H\right|^2 \rho_{n,k'}P_{max} + \sigma_{m,k}^2}, \quad (16)$$

Substituting (15) in (8), the final optimization problem is given as follows

$$\underset{\rho, \gamma, z, y}{\text{maximize}} \quad \sum_{m=1}^M \sum_{k=1}^K f_{m,k}^{CF} \quad (17)$$
$$\text{subject to} \quad (6a),$$
$$y_{m,k}, z_{m,k} \in \mathbb{R}, \quad (17a)$$

where $f_{m,k}^{CF}$ is given as follows

$$f_{m,k}^{CF} = \left(2y_{m,k}R_{m,k}(\rho, \gamma, z) - y_{m,k}^2\left(\rho_{m,k}P_{max} - p_{k,c}\right)\right) \quad (18)$$

The problem (17) is convex in $p_{m,k}$ when other variables are fixed. The solution for $p_{m,k}$ can be expressed in a closed-form by taking the derivative for the objective function in (17) and set it to zero. Hence, the closed-form solution is given as in (19) on to of next page, where $I_{m,k}$ is given as

$$I_{m,k} = \sum_{j=1}^{k-1}\left|\mathbf{h}_{m,k}^H\right|^2 \rho_{m,j}P_{max} + \sum_{n\in M, n\neq m}\sum_{k'\in K}\left|\mathbf{h}_{n,k}^H\right|^2 \rho_{n,k'}P_{max}, \quad (20)$$

Algorithm 2 Proposed Iterative Solution for Problem (17)

Initialization: $\rho_{m,k}$, ϵ, $t = 0$.
1: **while** $|f_{m,k}^{CF}(t) - f_{m,k}^{CF}(t-1)| \geq \epsilon$ **do**
2:     Update $z_{m,k}$ according to (16).
3:     Update $\gamma_{m,k}$ using (3).
4:     Update $y_{m,k}$ according to (9).
5:     Update $\rho_{m,k}$ by solving (19) with fixed $z_{m,k}$, $\gamma_{m,k}$, and $y_{m,k}$.
6:     $t = t + 1$.
7: **end while**

Theorem 1: Algorithm 1 and Algorithm 2 converge to stationary point of (6).

Proof: Please, refer to Appendix A.

Obtaining the solution for (12) requires solving the problem and updating the parameters for all users, this procedure requires $O\left(K(3M+1)\log\left(\frac{1}{\epsilon}\right)\right)$. Algorithm 2 performs the update for one additional step. Hence, Algorithm 2 entails an asymptotic complexity $O\left(K(3M+2)\log\left(\frac{1}{\epsilon}\right)\right)$ in the worst case.

IV. MODEL-BASED DESIGN FOR ENERGY-EFFICIENT POWER CONTROL

In this section, we discuss the structure of the proposed model-based design techniques for the energy-efficient power control. Two models are designed; the first model is a semi-unfolding based model in which we unfold Algorithm 1 while in the second model we fully unroll Algorithm 2 to design the model.

The main idea of deep unfolding is the unrolling of the iterations of principled inference algorithm to form a layered neural network [19]. Hence, in deep unfolding-based power control, we employ the main domain knowledge in wireless resource allocation and deep leaning to build more efficient and low complexity models. Fig. 1 shows the steps of the typical deep unfolding process.

The existence of the closed-form solution for the power allocation helps in designing a full unfolding-based model. However, this is difficult to apply in case of Algorithm 1 where the final form of the problem is solved numerically.
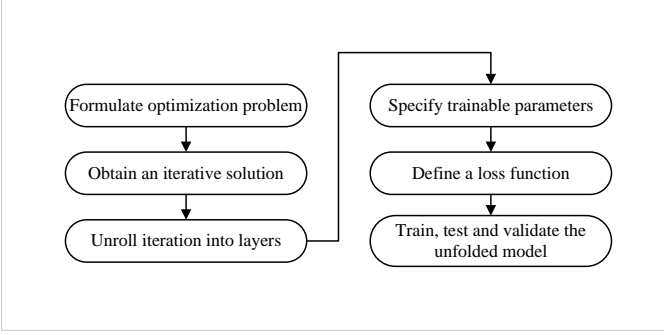
Fig. 1. A typical deep unfolding procedure.

A. Semi-Unfolding based Energy-Efficient Power Control

Fig. 2 shows the overall structure of the proposed fractional programming based multi-attention-based semi-unfolding model (MASUM). When projecting the above definition and concepts on our case, we can conceptualize the model design as follows

- Domain knowledge embedded design: Algorithm 1 requires initialization of power and channel gain as input. The proposed algorithm has the advantage of separating the signal and the interference and converting the original problem into subproblems using the auxiliaries y and z. The channel coefficients vector is considered as the main input to MASUM in addition to the initial power. Since y and z are functions of p, sub-networks are designed to represent their corresponding equations (9) and (11) as in the illustration of MASUM stages in Fig. 3.
- Leveraging deep learning advances: Due to the difficulties of fully representing the solution of (12) using the generic deep unfolding mechanism, we resort to the data-driven advances to overcome this problem. As in Fig. 3, convolutional layer is designed to create feature map for prediction. To increase the accuracy of the prediction, we design a permutation subnet [34] to augment the data and enrich the input with more possible variations. The convolutional feature map is cloned and fed parallelly into two pipelines. The first pipeline includes flattening layer to reshape the features and a fully connected layer to convey power prediction.

  In the second pipeline, multiple attention blocks take the convolutional features and create feature maps where the impact of each single parameter is coded and the patterns are learnt and spatially connected [18].

  Given the convolutional feature map f_c, we apply parallel three 1×1 Conv layers on the f_c. Thus, we obtain the outputs f_c^1, f_c^2, and f_c^3. f_c^1 and f_c^2 are multiplied and inserted into a softmax as follows

$$f_s = softmax\left(f_c^1 \cdot f_c^2\right) \tag{21}$$

  f_s is the interaction between the different positions in the data where each element in f_s is a weighted sum of features of other elements. Then, f_s is used to calibrate the impact of each element in the feature map f_c as follows

$$f_c^{cal} = f_s \cdot f_c^3 \tag{22}$$

f_c^{cal} is reshaped and 1×1 Conv is applied where $\hat{f}_c^{cal}$ is obtained. The final output feature map of the attention block is as

$$f_A = \hat{f}_c^{cal} + f_c \tag{23}$$

Multiple attention feature maps are fused using fusion network. To perform feature fusion, multiple attention feature maps are concatenated and fed into two 1×1 Conv layers to deliver the fusion map f^{fusion}.

The fused features are fed into pooling and flattening layer for reshaping. Then, a fully connected layer is used to predict the power $\mathbf{p}_{Att}$. The second pipeline can be seen as compensation of the prediction loss in the first pipeline.

The power predictions from the two pipelines are refined to convey the final power prediction $\hat{\mathbf{p}}$. Where $\hat{\mathbf{p}}$ is given as

$$\hat{\mathbf{p}} = \sigma\left(\mathbf{p}_{Att} + \mathbf{p}_{main}\right) \cdot \left(\mathbf{p}_{Att} + \mathbf{p}_{main}\right) \tag{24}$$

Fig. 4 shows the detailed structure of the attention block, fusion block, and refining block.

B. Full-Unfolding based Energy-Efficient Power Control

The closed-form solution in (19) is helpful in fully unfolding Algorithm 2. Fig. 5 illustrates the block diagram of the full-unfolding based model(FUM). In Fig. 3, the second block represents the structural illustration of the single stage of FUM.

The structure in Fig. 5 can be viewed as DNN with number of layers equals to the iteration of Algorithm 2. The dynamics of each layer are described as in the second block in Fig. 3. The structure can be viewed as a graph with shared variables among the computation nodes. To optimize the parameters, we employ the well-known technique of backpropagation. To efficiently optimize the model in Fig. 5, employ the available data and the underlaying distribution of the channel realization of in $\mathbf{G}$.

C. Technical Description and Inference Process

In this subsection, we give the technical description of the proposed two models
  1) MASUM Case:
- Forward Propagation: Provided with the channel coefficients the initial power, each block in the main pipeline in Fig. 2 outputs a power vector. However, due to the presence of the data-driven layers (namely, the convolution, flatten, and fully connected layers), the model exhibits some performance loss. Hence, an effective attention-based loss compensation mechanism in the second pipeline is adopted. The forward propagation $\mathcal{F}(\cdot)$ of MASUM can be expressed as

$$\hat{\mathbf{p}} = \mathcal{F}\left(\{G\}; \mathbf{p}\right) \tag{25}$$

- Backpropagation: In this procedure, the gradients of the learnable parameters are computed following the derivative of the loss function $\mathcal{L}(\cdot)$ [35].
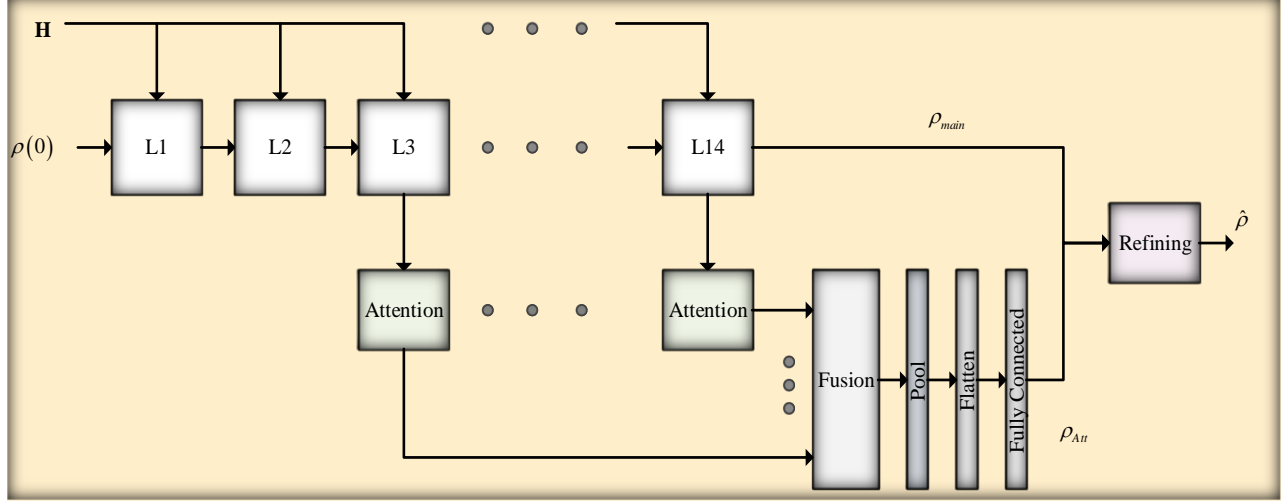- Updating Parameters: Since each block in Fig. 2 receives power input from the previous block, the update of the
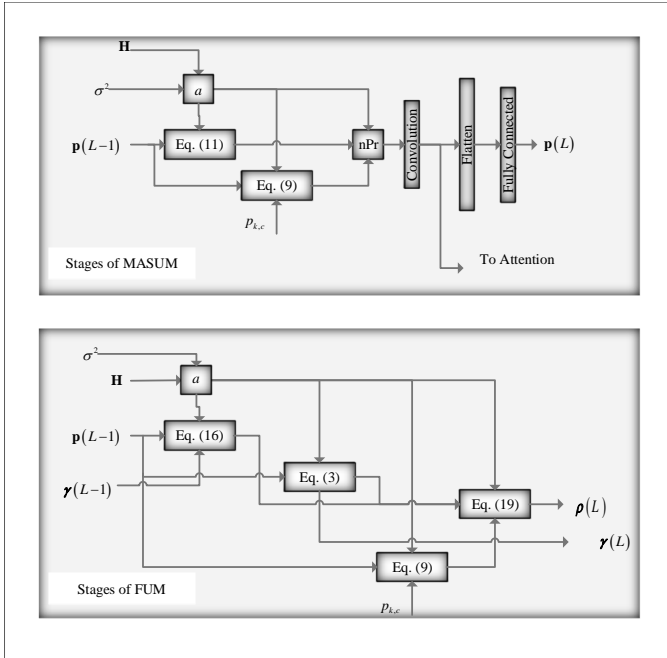
Fig. 2. Structure of the proposed model.

Fig. 3. Detailed structure of stages of the proposed two models.

Fig. 4. Detailed building blocks for attention, fusion, and refining blocks.

Fig. 5. Structure of the proposed FUM.

parameters is achieved based on (9) and (11) in addition to the gradient over $\mathcal{L}(\cdot)$ with respect to the power.

- *Loss Function Characterization:* The loss function can be expressed as

$$\mathcal{L}(\{G, \mathbf{p}\}; \hat{\mathbf{p}}) \stackrel{\Delta}{=} \tilde{f}(G, \mathcal{F}(\{G\}; \mathbf{p}); \hat{\mathbf{p}}) \qquad (26)$$

where $\tilde{f}(\cdot)$ given in (13).

During the inference process, in each layer, the loss function is optimized in such a way that the power prediction is improved. However, performance loss is inevitable due to the data-driven. This loss is compensated by employing the changes throughout the main pipeline via the attention pipeline.
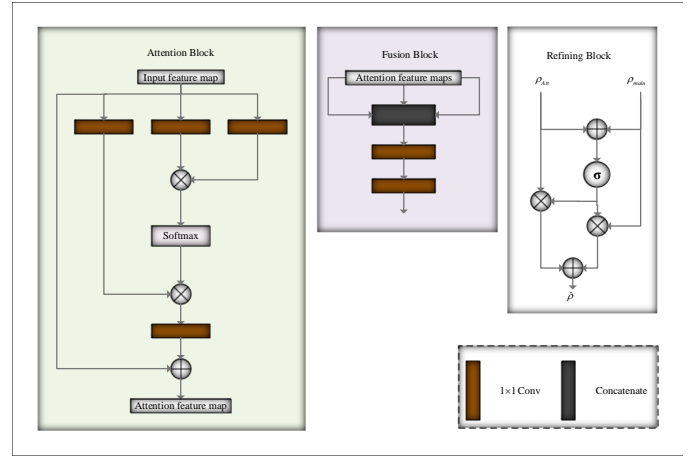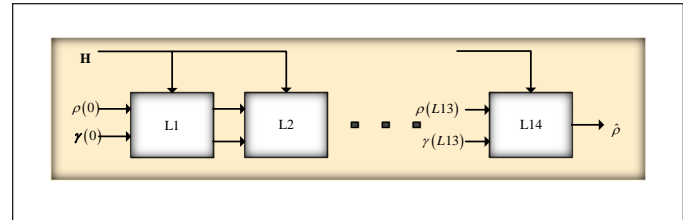
2) FUM Case: The learnable parameters in this case are $\mathbf{p}$ and γ. Each block outputs the $\mathbf{p}$ and γ while the final block conveys only $\mathbf{p}$. The forward propagation $\mathcal{F}(\cdot)$ in this case can be expressed as

$$\{\hat{\mathbf{p}}, \gamma\} = \mathcal{F}(G; \{\mathbf{p}, \gamma\}) \qquad (27)$$

In the backpropagation, the gradients of the loss function with respect to the learnable parameters are derived using the chain rule. The loss function $\mathcal{L}(\cdot)$ can be expressed as

$$\mathcal{L}(G, \{\mathbf{p}, \gamma\}; \hat{\mathbf{p}}) \stackrel{\Delta}{=} f(G, \mathcal{F}(G; \{\mathbf{p}, \gamma\}); \hat{\mathbf{p}}), \qquad (28)$$

TABLE I Simulation Parameters

Parameter	Value
Number of users per BS	2,4,8
Number of BSs	4,7
Number of antennas N	2
Fast-Fading	$Z \sim \mathcal{CN}(0,1)$
Path loss exponent	3.5
Number of BS antennas	8
B	250 kHz
Carrier frequency	2.4 GHz
Circuit power	20 dBm

where $f(\cdot)$ is defined in (18). The updates of the learnable parameters at the sequence t can be expressed as

$$\gamma^{(t+1)}(L) = \gamma^{(t)}(L) - \delta \frac{\partial \mathcal{L}\left(G, \{\mathbf{p}, \gamma\}; \hat{\mathbf{p}}^{(t)}\right)}{\partial \gamma} \quad (29)$$

$$\mathbf{p}^{(t+1)}(L) = \mathbf{p}^{(t)}(L) - \delta \frac{\partial \mathcal{L}\left(G, \{\mathbf{p}, \gamma\}; \hat{\mathbf{p}}^{(t)}\right)}{\partial \mathbf{p}} \quad (30)$$

where δ is the learning rate. From Fig. 3, the above update is also equivalent to (3) and the first term in (19).

Different from MASUM, during the inference, FUM optimizing the loss function straightforwardly while leveraging the full domain knowledge of the problem and the statistical information.

D. Training and Testing Process

Both of the models can be trained using the incremental learning procedure. In the increment round $\tau \in \{0, ..., T\}$, we optimize the cost function for the layer $l \in \{0, ..., L14\}$ by learning the learnable parameters of each model with mini-batches. At the end of increment round τ, the layer $l = l + 1$ is added to the network and a new round of training starts. The training results of learnable parameters of the layer l are considered as initial values for the next round. At the final round, the entire model goes through a full training round. This training mechanism is proven to be the most effective training procedure for deep unfolding-based model.

For the simulation, we generate 22000 samples where 8000 samples are used for training where we run each of Algorithm 1 and Algorithm 2 multiple times to retain the best results. We generate the channel coefficients using the complex Gaussian distribution $\mathcal{CN}(0,1)$ and the number of antennas $N = 8$.

The batch size is set to 128 and 64 for MASUM and SUM respectively. The learning rate is set to 0.001 for both models. For MASUM, the kernel size is 3 in the main stage while the stride is set to 1. The simulations are performed with Python 3.6, PyTorch 2.0, CUDA 10.1, cuDNN V8 on a desktop Intel Core i7-8700 CPU @ 3.20GHz, 8.00 GB RAM, and NVIDIA GeForce RTX 3050 8GB.

V. SIMULATION RESULTS

In this section, we introduce the simulation results and performance evaluation for the proposed models. Each cell radius is 500 m and the adopted channel model follow the in force ITU-R P.1411-11. The users are randomly and uniformly distributed in the cells. The subchannel frequency is 250 kHz and the carrier frequency is 2.4 GHz. The noise power spectral

density is -174 dBm/Hz and the noise figure is 7dB. The simulation parameters are deemed default as shown in TABLE I unless stated otherwise.

For comparison, we considered the deep learning model in [18] as a black-box model (pure data-driven model). The black-box model is a multi-modal deep convolutional self-attention neural network. The black-box model is trained with multivariate data generated from Algorithm 1.

A. Accuracy and Convergence

From Theorem 1, both Algorithm 1 and Algorithm 2 converge to a fixed point of the original problem (6). Fig. 6 illustrates the convergence of both algorithms. The required number of iterations until convergence is 14 and 13 respectively for Algorithm 1 and Algorithm 2. Despite the slow convergence, both algorithms have the advantage of exploring the solution space.

From the complexity discussion of both algorithms, the number of users has the most impact on the running time of both algorithms. For instance, running both algorithms on CPU for 4 users per BS, we obtain the final output in 25.44 ms and 25.56 ms respectively for Algorithm 1 and Algorithm 2. In case of running both algorithms on GPU, we have the running time 10.04 ms and 10.01 ms respectively for Algorithm 1 and Algorithm 2.
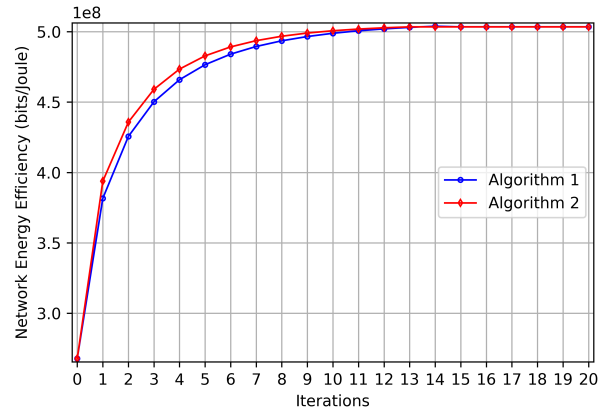

Fig. 6. Convergence of Algorithm 1 and Algorithm 2.

Fig. 7 shows the training and validation accuracy of MASUM. MASUM has the advantage of partially exploiting the domain knowledge of the problem and the available data. However, due to the presence of the data-driven parts in the model, the presence of the attention is advantageous as we can see later in the ablation study (see Section V-E). MASUM achieves training accuracy 98.80% and validation accuracy of 98.78 for the case of full exploitation of model number of layers (e.g., 9 layers).

Different from MASUM, FUM is fully exploiting both domain knowledge of the problem and the given available data. As in Fig. 8, FUM achieves accuracy 99.32% and 99.30% respectively for training and validation. The proposed MASUM and FUM has a very quick inference since it does not require solving the problem as in the case Algorithm 1
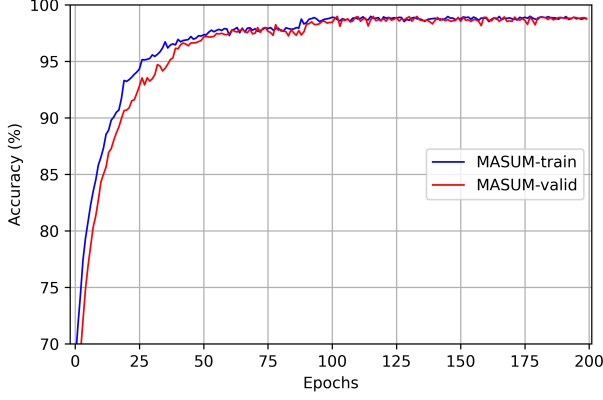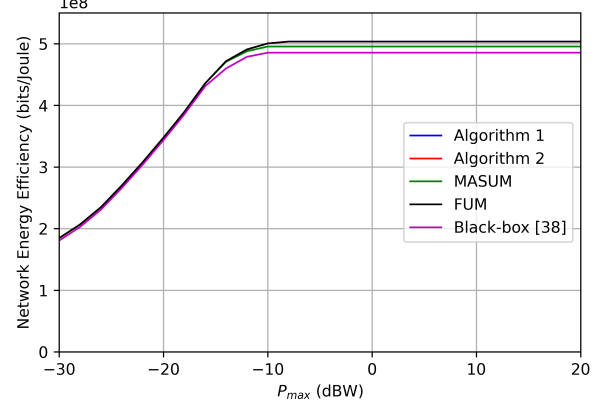
Fig. 7. Accuracy of MASUM.

Fig. 9. Network energy efficiency for different values of P_{max}.

and Algorithm 2. Hence, both MASUM and FUM exhibit great potential for real time applications. More details on the inference speed are revealed in Section V-E.

Fig. 8. Accuracy of FUM.

B. Power Budget

Fig. 9 illustrates the energy efficiency performance for different values of P_{max}. We set the number of BSs to 7 and the number of users per BS is set to 4. The network energy efficiency is increasing with the increasing of P_{max} and converges at 5.0342×10^8 bits/Joule when $P_{max} = -9$ dBW. This means any increase in P_{max} will lead only to increasing interference and the consumption and will not help in improving the rate. The performance of Algorithm 1, Algorithm 2, and FUM is identical while MASUM has slightly lower performance. Black-box model achieves the worst performance among all.

C. Problem Size

For smaller problem when we have 7 BSs and each BS is serving 2 users, the performance improves dramatically. We reduced the size of the model to 6 layers in the main pipeline while including the attention in the last 5 layers and trained it for shorter training sequence. In Fig. 10, we can notice that

the accuracy of MASUM is high to 99.01 % and 98.94% respectively for training and validation.
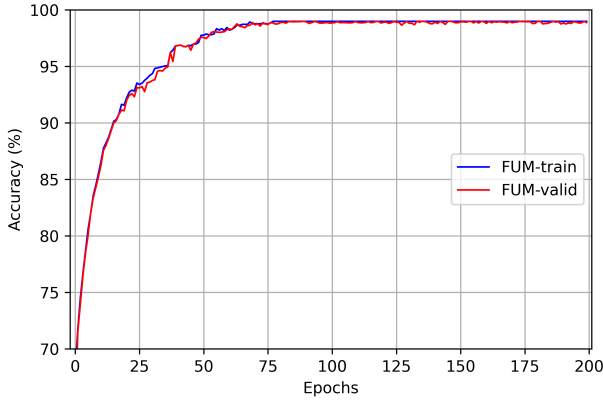
Fig. 10. Accuracy of MASUM for small problem (2 users per BS).
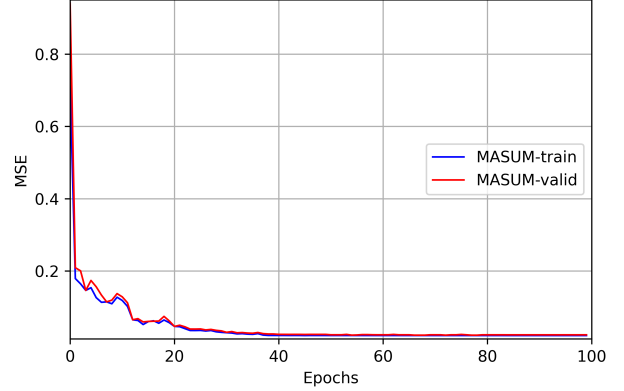
In case of FUM, we have 6 layers and we apply it on small problem similar to that in last case where we have 7 BSs; each BS is serving 2 users. From Fig. 11 FUM achieves accuracy 99.37% and 99.34% respectively for training and validation.
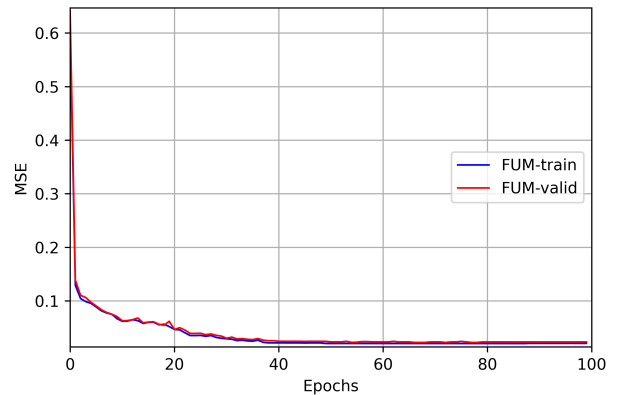

Fig. 11. Accuracy of FUM for 2 users per BS.

Compared with the procedure in Fig. 9, we test the models over large problem as in Fig. 12. We set the number of BS to

TABLE II ABLATION STUDY ON THE IMPACT OF MODEL'S NUMBER OF STAGES/LAYERS

Model	No. of stages	Prediction accuracy (%)	Inference speed (ms)	
			CPU	GPU
MASUM	5	98.55	0.0692	0.0571
	9	98.80	0.0823	0.0615
	14	98.80	0.0936	0.0652
FUM	5	99.11	0.0513	0.0420
	9	99.32	0.0534	0.0464
	14	99.32	0.0545	0.0474

7 while the number of users per BS is set to 8. Considering the case of MASUM and FUM, we can observe that there is a degradation in the performance of both models. However, unlike the case of MASUM, this degradation is smaller in case of FUM. In case of MASUM, this degradation might be alleviated by increasing the number of neurons, the layers, and the attention blocks. Similarly, the increase of the number of neurons can decrease the impact of this degradation. It is worth mentioning that the adopted training procedure in this study has great impact on solidifying the performance of both models. MASUM achieves 2.4% performance less than Algorithm 1 while FUM achieves about 1% performance less than Algorithm 1. Although Black-box has the capability of learn the statistical information, its performance deteriorates with larger problems as in Fig. 12. Black-box achieves about 6.8% performance less Algorithm 1.
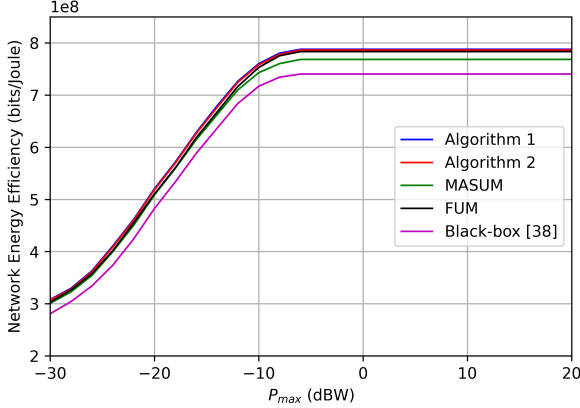
better than MASUM. FUM achieves 99.20% performance of Algorithm 1 while MASUM achieves 96.57% performance of Algorithm 1. Finally, Black-box performance significantly degrades and only 89.55% of Algorithm 1 can be achieved.
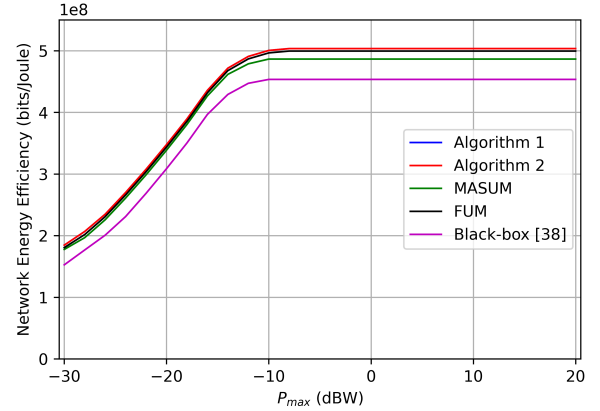

Fig. 13. Performance of different models over off-training data.

E. Ablation Study

According to the deep unfolding concept, the number of the layers of the resulting model is mostly expected to be equal to the number of the iterations of the traditional algorithm. However, the number of the required layers to achieve the maximum possible accuracy usually does not depend on the number of iterations. Hence, we implement an ablation study to investigate the optimal structure to build each of the proposed MASUM and FUM models. TABLE II investigates the optimal number of layers to obtain the maximum possible accuracy while maintaining the inference speed at reasonable level. In case of FUM, we can observe that five to six layers are enough to obtain excellent performance. For instance, for five layers, the accuracy is 99.11% and the model delivers the output in 0.0513 ms and 0.0420 ms respectively in CPU and GPU. However, the larger number of layers, take 9 layers, we obtain accuracy of 99.32% with inference speed 0.0534 ms and 0.0464 ms for CPU and GPU, respectively. Obviously, increasing the number of layers beyond 6 layers will only lead to increase the inference speed with any improvement in the inference speed. It is also worth mentioning that the adopted training mechanism has great impact in tuning the model and optimizing the number of layers. Similar observations can be drawn in the case MASUM. However, since MASUM

Fig. 12. Network energy efficiency for different values of P_{max} 7 BSs and 8 users per BS.

D. Inference over Off-training Input

Taking into consideration the changes in the channel conditions in the real-world, we test the schemes with off-training data and the result is shown as in Fig. 13. We set the number of BSs to 7 and the number of users per BS is set to 4. Although there is a deterioration in the performance of FUM and MASUM compared with Algorithm 1 and Algorithm 2, the proposed two deep learning models show robustness against the changes in the channel conditions. FUM employs the full domain knowledge of the problem and learns the statistical information from the data. Therefore, FUM performs

TABLE III IMPACT OF THE NUMBER OF ATTENTION BLOCKS

No. of attention blocks	Accuracy (%)	Inference speed (ms)	Location of attention blocks
0	97.92	0.0052	-
2	98.68	0.0588	4, 5
3	98.55	0.0602	3,4,5
5	98.80	0.0615	5,6,7,8,9
7	98.80	0.0711	8,9,10,11,12,13,14

is of a hybrid nature in which model-driven and data-driven mechanisms are fused, other factors such as structure of the data-driven blocks and the position of these blocks have crucial impact on the performance.

TABLE III illustrates the impact of the attention block on the performance of MASUM. First, it is obvious that the number of the usable attention block is dependent on the number of the overall layers. Second, the locations of the attention blocks have impact on the accuracy because it employ the changes in the features and create spatial connection and better context for prediction. Therefore, in the case of have five-layer MASUM, we deploy the attention blocks in the last two and three layers (depending on the number of the deployed attention blocks). From TABLE III, we can observe that the reasonable number of attention blocks is 5 if we considered the 7-layer structure. Also it is obvious that the inference speed decreases with the increase of number of attention blocks.

VI. CONCLUSIONS AND FUTURE WORKS

In this work, we designed and compared two deep unfolding-based models for power control in multiple interference links. The problem was formulated as SoRP which is nonconvex. We considered two solution scenarios for the solution. First, we employed multidimensional fractional programming to obtain a numerical solution. Second, we combined multidimensional fractional programming and Lagrange dual transform to obtain a closed-form solution for the problem. On the highlights of the two solutions, we designed two deep unfolding-based models. The first model (namely MASUM) is based on the numerical solution where two pipelines were designed to predict the power. The first pipeline emulates the iterations of the iterative algorithm with the help of some data-driven layers. Due to the loss in the main pipeline, a second pipeline consists of attention subnets is incorporated to compensate the loss. Due to the presence the closed-form solution, the second model-driven neural network (namely FUM) can fully emulate the iterative solution with neither convolution nor attention subnet. Both MASUM and FUM achieved high accuracy in terms of power allocation with spectacular inference speed compared to the iterative solutions. We conclude that the presence of closed-form solution can facilitate unfolding the iterative solution into model-driven neural network. Furthermore, in case of difficulty of fully unfolding the iterative solution, we resort to semi-unfolding where we can partially unfold the iterative solution then employ the recent advances in data-driven models to enhance the performance. In future, we consider designing improved model-driven neural networks for large size problems using

architectures such as inception-residual. To make the future model more suitable for dynamic environment and sparsity of wireless communication networks, we aim to incorporate the idea of liquid neural networks when building the model.

APPENDIX A

PROOF OF THEOREM 1

Let us consider the case of Algorithm 1. Let $f(p) = \sum_{m=1}^{M} \sum_{k=1}^{K} \omega_{m,k} \eta_{m,k}$. We know from the definition of $\eta_{m,k}$ it is nondecreasing SoR. Hence, the problem (6) can be conveniently considered equivalent to the following:

$$\underset{\rho,y}{\text{maximize}} \quad f(p,y)$$
$$\text{subject to} \quad (6a), (8a),$$

where $f(p,y) = \sum_{m=1}^{M} \sum_{k=1}^{K} \omega_{m,k} \left(2y_{m,k} \sqrt{R_{m,k}} - y_{m,k}^2 (p_{m,k} + p_{k,c}) \right)$. Furthermore, we can rewrite the (6) as a function of η as below:

$$\underset{\rho,y}{\text{maximize}} \quad f(\eta)$$
$$\text{subject to} \quad (6a),$$
$$\eta_{m,k} = \frac{R_{m,k}}{\rho_{m,k} P_{max} + p_{k,c}},$$

Hence, according to the quadratic transform theory in [30], we can replace the variable η with $\omega_{m,k} \left(2y_{m,k} \sqrt{R_{m,k}} - y_{m,k}^2 (\rho_{m,k} P_{max} + p_{k,c}) \right)$ and since $f(\eta)$ is nondecreasing, $\max_{p} \sum_{m=1}^{M} \sum_{k=1}^{K} \max_{y} \left[\omega_{m,k} \left(2y_{m,k} \sqrt{R_{m,k}} - y_{m,k}^2 (\rho_{m,k} P_{max} + p_{k,c}) \right) \right]$ can be rewritten as (31).

Next, we recast the data rate term as concave as in (10). Consequently, we obtain problem (12). At each iteration, and when $z_{m,k}$ and $y_{m,k}$ are fixed, the solution represents a stationary point in which Algorithm 1 converges. Similar reasoning can be applied in case of Algorithm 2.

REFERENCES

[1] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The Road Towards 6G: A Comprehensive Survey," *IEEE open j. Commun. Soc.,* vol. 2, pp. 334-366, 2021.

[2] R. Li, "Network 2030 A Blueprint of Technology, Applications and Market Drivers Towards the Year 2030 and Beyond," July 2020.

[3] P. Yang, Y. Xiao, M. Xiao, and S. Li, "6G Wireless Communications: Vision and Potential Techniques," *IEEE Network,* vol. 33, no. 4, pp. 70-75, 2019.

[4] M. A. Ouamri, G. Barb, D. Singh, A. B. M. Adam, M. S. A. Muthanna and X. Li, "Nonlinear Energy-Harvesting for D2D Networks Underlaying UAV With SWIPT Using MADQN," *IEEE Commun. Lett.,* vol. 27, no. 7, pp. 1804-1808, July 2023.

[5] A. B. M. Adam, X. Wan and Z. Wang, "Energy Efficiency Maximization in Downlink Multi-Cell Multi-Carrier NOMA Networks With Hardware Impairments," *IEEE Access,* vol. 8, pp. 210054-210065, 2020.

[6] A. B. M. Adam, X. Wan, Z. Wang, "Energy Efficiency Maximization for Multi-Cell Multi-Carrier NOMA Networks," *Sensors,* 2020, 20(22):6642.

[7] A. B. M. Adam, X. Wan, Z. Wang, "Clustering and Auction-Based Power Allocation Algorithm for Energy Efficiency Maximization in Multi-Cell Multi-Carrier NOMA Networks," *App. Sci.,* 2019; 9(23):5034.

[8] A. B. M. Adam, X. Wan, Z. Wang, "User scheduling and power allocation for downlink multi-cell multi-carrier NOMA systems," Digital Communications and Networks, vol. 9, no. 1, pp. 252-263, 2023.

[9] B. Marinberg, A. Cohen, E. Ben-Dror, and H. H. Permuter, "A Study on MIMO Channel Estimation by 2D and 3D Convolutional Neural Networks," *in Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Sys. (ANTS),* 2020, pp. 1-6.

[10] H. Ye, G. Y. Li, and B. H. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," *IEEE Wireless Commun. Lett.,* vol. 7, no. 1, pp. 114-117, 2018.

[11] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep Learning for Super-Resolution Channel Estimation and DOA Estimation Based Massive MIMO System," *IEEE Trans. Veh. Technol.,* vol. 67, no. 9, pp. 8549-8560, 2018.

[12] Z. Mao and S. Yan, "Deep learning based channel estimation in fog radio access networks," *China Communications,* vol. 16, no. 11, pp. 16-28, 2019.

[13] Q. Bai, J. Wang, Y. Zhang, and J. Song, "Deep Learning-Based Channel Estimation Algorithm Over Time Selective Fading Channels," *IEEE Trans. Cogn. Commun. Netw.,* vol. 6, no. 1, pp. 125-134, 2020.

[14] L. Wan, K. Liu, and W. Zhang, "Deep Learning-Aided Off-Grid Channel Estimation for Millimeter Wave Cellular Systems," *IEEE Trans. Wireless Commun.,* pp. 1-1, 2021.

[15] A. B. M. Adam and X. Wan, "Deep Learning based Efficient User Association and Subchannel Assignment in Multi-Cell Multi-Carrier NOMA Networks," *in Proc. 7th Int. Conf. Inf. Sci. Control Eng. (ICISCE),* Changsha, China, Dec. 2020, pp. 448-452.

[16] M. Labana and W. Hamouda, "Unsupervised Deep Learning Approach for Near Optimal Power Allocation in CRAN," *IEEE Trans. Veh. Technol.,* vol. 70, no. 7, pp. 7059-7070, 2021.

[17] A. B. M. Adam, Z. Wang, X. Wan, Y. Xu and B. Duo, "Energy-Efficient Power Allocation in Downlink Multi-Cell Multi-Carrier NOMA: Special Deep Neural Network Framework," *IEEE Tran. Cogn. Commun. Netw.,* vol. 8, no. 4, pp. 1770-1783, Dec. 2022.

[18] A. B. M. Adam, L. Lei, S. Chatzinotas, and N. U. R. Junejo, "Deep Convolutional Self-Attention Network for Energy-Efficient Power Control in NOMA Networks," *IEEE Trans. Veh. Technol.,* vol. 71, no. 5, pp. 5540-5545, May 2022.

[19] A. Jagannath, J. Jagannath, and T. Melodia, "Redefining Wireless Communication for 6G: Signal Processing Meets Deep Learning With Deep Unfolding," *IEEE Trans. Artificial Intell.,* vol. 2, no. 6, pp. 528-536, 2021.

[20] D. Ito, S. Takabe, and T. Wadayama, "Trainable ISTA for Sparse Signal Recovery,"*IEEE Trans. Signal Process.,* vol. 67, no. 12, pp. 3113-3125, 2019.

[21] S. Khobahi and M. Soltanalian, "Model-Based Deep Learning for One-Bit Compressive Sensing," *IEEE Trans. Signal Process.,* vol. 68, pp. 5292-5307, 2020.

[22] Q. Hu, Y. Cai, Q. Shi, K. Xu, G. Yu, and Z. Ding, "Iterative Algorithm Induced Deep-Unfolding Neural Networks: Precoding Design for Multiuser MIMO Systems," *IEEE Trans. Wireless Commun.,* vol. 20, no. 2, pp. 1394-1410, 2021.

[23] G. Zhang, X. Fu, Q. Hu, Y. Cai, and G. Yu, "Hybrid Precoding Design Based on Dual-Layer Deep-Unfolding Neural Network," *in Proc. IEEE 32nd Annual Int. Symp. Personal, Indoor Mobile Radio Commun. (PIMRC),* 2021, pp. 678-683.

[24] C. H. Lin, Y. T. Lee, W. H. Chung, S. C. Lin, and T. S. Lee, "Unsupervised ResNet-Inspired Beamforming Design Using Deep Unfolding Technique," *in Proc. IEEE Global Commun. Conf. (GLOBECOM),* 2020, pp. 1-7.

[25] S. Takabe and T. Wadayama, "Deep Unfolded Multicast Beamforming," *in Proc. IEEE Global Commun. Conf. (GLOBECOM),* 2020, pp. 1-6.

[26] Y. Liu, Q. Hu, Y. Cai, G. Yu, and G. Y. Li, "Deep-Unfolding Beamforming for Intelligent Reflecting Surface assisted Full-Duplex Systems," *IEEE Trans. Wireless Commun.,* pp. 1-1, 2021.

[27] A. B. M. Adam et al., "Intelligent and Robust UAV-Aided Multiuser RIS Communication Technique With Jittering UAV and Imperfect Hardware Constraints," *IEEE Trans. Veh. Technol.,* vol. 72, no. 8, pp. 10737-10753, Aug. 2023.

[28] A. Chowdhury, G. Verma, C. Rao, A. Swami, and S. Segarra, "Unfolding WMMSE Using Graph Neural Networks for Efficient Power Allocation," *IEEE Trans. Wireless Commun.,* vol. 20, no. 9, pp. 6004-6017, 2021.

[29] A. Zappone and E. Jorswieck, *Energy Efficiency in Wireless Networks via Fractional Programming Theory, ser. Found. Trends Commun. Inf. Theory,* Boston, MA, USA: Now, 2015, vol. 11, no. 3-4, pp. 113-381.

[30] K. Shen and W. Yu, "Fractional Programming for Communication Systems-Part I: Power Control and Beamforming," *IEEE Trans. Signal Process.,* vol. 66, no. 10, pp. 2616-2630, 2018.

[31] A. Zappone, L. Sanguinetti, G. Bacci, E. Jorswieck, and M. Debbah, "Energy-Efficient Power Control: A Look at 5G Wireless Technologies," *IEEE Trans. Signal Process.,* vol. 64, no. 7, pp. 1668-1683, 2016.

[32] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, Version 3.0 Beta," Available online: http://cvxr.com/cvx/beta/, 2017.

[33] K. Shen and W. Yu, "Fractional Programming for Communication Systems-Part II: Uplink Scheduling via Matching," *IEEE Trans. Signal Process.,* vol. 66, no. 10, pp. 2631-2644, 2018.

[34] K. -L. Besser, B. Matthiesen, A. Zappone and E. A. Jorswieck, "Deep Learning Based Resource Allocation: How Much Training Data is Needed?," *2020 In Proc. IEEE 21st Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC),* Atlanta, GA, USA, 2020, pp. 1-5.

[35] Y. Liu, Q. Hu, Y. Cai, G. Yu and G. Y. Li, "Deep-Unfolding Beamforming for Intelligent Reflecting Surface assisted Full-Duplex Systems," *IEEE Trans. Wireless Commun.,* vol. 21, no. 7, pp. 4784-4800, July 2022.