TO SUPERVISE OR NOT TO SUPERVISE: HOW TO EFFECTIVELY LEARN WIRELESS INTERFERENCE MANAGEMENT MODELS?

Bingqing Song[⋆] *Haoran Sun*[⋆] *Wenqiang Pu*[*] *Sijia Liu*[†] *Mingyi Hong* [⋆]

[⋆] ECE Department, University of Minnesota, Minneapolis, MN
[†] MIT-IBM Watson AI Lab, Boston, MA
[*] Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, China

ABSTRACT

Machine learning techniques have become successful in solving wireless interference management problems. Different kinds of deep neural networks (DNNs) have been trained to accomplish key tasks such as power control, beamforming and admission control. There are two state-of-the-art approaches to train such DNNs based interference management models: supervised learning (i.e., fits labels generated by an optimization algorithm) and unsupervised learning (i.e., directly optimizes some system performance measure). However, it is by no means clear which approach is more effective in practice. In this paper, we conduct some theory - and experiment - guided study about these two training approaches. First, we show a somewhat surprising result, that for some special power control problem, the unsupervised learning can perform much worse than its counterpart, because it is more likely to stuck at some low-quality local solutions. We then provide a series of theoretical results to further understand the properties of the two approaches, as well as a semi-supervised training algorithm that bridges them. To our knowledge, these are the first set of theoretical results trying to understand different training approaches in learning-based wireless communication system design.

I. INTRODUCTION

Motivation. Recently, machine learning techniques have become very successful in solving wireless interference management problems. Different kinds of deep neural network (DNN), such as fully connected network (FCN) [1], recurrent neural network (RNN) [2], graph neural network (GNN) [3], [4] have been designed to accomplish key tasks such as power control, beamforming [5], admission control [6], MIMO detection [7], among others. These DNN based models are capable of achieving competitive and sometimes even superior performance compared to the state-of-the-art optimization based algorithms [8].

However, despite its success, there is still a fundamental lack of understanding about *why* DNN based approaches work so well for this class of wireless communication problems – after all, the majority of interference management problems (e.g., beamforming) are arguably more complex than a typical machine learning problem such as image classification. It is widely believed that, exploiting task-specific properties in designing network architectures, as well as training objectives can help significantly reduce the network complexity and input feature dimension [8], boost the training efficiency [8], and improve the expressiveness of the DNN [1].

The overarching goal of this research is to understand how problem-specific properties can be effectively utilized in the DNN design. More concretely, we attempt to provide an in-depth understanding about how to effectively utilize problem structures in designing efficient training procedures. Throughout the paper, we will utilize the classical weighted sum rate (WSR) maximization problem in single-input single output (SISO) interference channel as a working example, but we believe that our approaches and the phenomenon we observed can be extended to many other related problems.

Problem Statement and Contributions. Consider training DNNs for power control, or more generally for beamforming. There are two state-of-the-art approaches to train such DNNs:

1) *supervised learning (SL)*, in which lots of "labels" of optimal power allocation are generated by an optimization algorithm, then the training step minimizes the mean square error (MSE) between the the DNN outputs and the labels [1];

2) *unsupervised learning (UL)*, which optimizes some system performance measure such as WSR [8].

It is clear that the above unsupervised approach is unique to the interference management problem, because the specific task of WSR maximization offers a natural training objective to work with. Further, it does not require any existing algorithms to help generate high-quality labels (which could be fairly expensive). On the other hand, such an objective is difficult to optimize since the WSR is a highly non-linear function with respect to (w.r.t.) the transmit power, which is again a highly non-linear function of the DNN parameters.

Which training method shall we use in practice? Can we rigorously characterize the behavior of these methods? Is it possible to properly integrate these two approaches to yield a

more efficient training procedure? Towards addressing these questions, this work makes the following key contributions:
❶ We focus on the SISO power control problem in interference channel (IC), and identify a simple 2-user setting, in which UL approach has *non-zero probability* of getting stuck at low-quality solutions (i.e., the local minima), while the SL approach always finds the global optimal solution;
❷ We provide rigorously analysis to understand properties of UL and SL for DNN-based SISO-IC problem. Roughly speaking, we show that when high-quality labels are provided, SL should outperform UL in terms of solution quality. Further, the SL approach converges faster when the labels have better solution quality;
❸ In an effort to leverage the advantage of both approaches, we develop a *semi-supervised* training objective, which regularizes the unsupervised objective by using a few labeled data points. Surprisingly, by only using a small fraction ($\approx 1\%$) of samples of the supervised approach, the proposed method is able to avoid bad local solutions and attain similar performance as supervised learning.

To the best of our knowledge, this work provides the first in-depth understanding about the two popular approaches for training DNNs for wireless communication.

II. PRELIMINARIES

Consider a wireless network consisting of K pairs of transmitters and receivers. Suppose each pair of transmitter and receiver equips with a single antenna, denote $h_{kj} \in \mathbb{C}$ as the channel between the kth transmitter and the jth receiver, p_k as the power allocated to the kth transmitter, $P_{\max}$ as the budget of transmitted power, and σ^2 as the variance of zero-mean Gaussian noise in the background. Further, we use w_k to represent the prior importance of the kth receiver, then the classical WSR maximization problem can be formulated as

$$\max_{p_1,\ldots,p_K} \sum_{k=1}^{K} w_k \log\left(1+\frac{|h_{kk}|^2 p_k}{\sum_{j\neq k}|h_{kj}|^2 p_j+\sigma_k^2}\right) := R(\mathbf{p};|\mathbf{h}|)$$
$$\text{s.t.} \quad 0 \le p_k \le P_{\max}, \forall k=1,2,\ldots,K \tag{1}$$

where $\mathbf{h} := \{h_{kj}\}$ collects all the channels; $|\cdot|$ is the componentwise absolute value operation; and $\mathbf{p} := (p_1, p_2, \ldots, p_K)$ denotes the transmitted power of K transmitters. The above problem is well-known in wireless communication, and it is known to be NP-hard [9] in general. For problem (1) and its generalizations such as the beamforming problems in MIMO channels, many iterative optimization based algorithms have been proposed, such as waterfilling algorithm [10], interference pricing [11] , WMMSE [12], SCALE [13].

Recently, there has been a surge of works that apply DNN based approach to identify good solutions for problem (1) and its extensions [1], [7], [8], [14]. Although these works differ from their problem settings and/or DNN architectures, they all use either the SL, UL, or some combination of the two to train the respective networks. Below let us take problem (1) as an example and briefly compare the SL and UL approaches.

• **Data Samples:** Both approaches require a collection of the channel information over N different snapshots, denoted as $\mathbf{h}^{(n)}$, $n=1,2,\ldots,N$. SL requires an additional N labels $\bar{\mathbf{p}}^{(n)}$, $n=1,2,\ldots,N$, which are usually obtained by solving N independent problems (1) using some optimization algorithm, such as the WMMSE [12]. Notice that the quality of such labels may depend on the accuracy of the optimization algorithm being selected.

• **DNN Structure:** We will assume that the power allocation $\mathbf{p}$ is parameterized by some DNN. More precisely, the inputs of the DNN are absolute values of channel samples $\mathbf{h}^{(n)}$, and let $\mathbf{\Theta}$ be the parameters of the DNN (of appropriate size), then the output of DNN can be expressed as $\mathbf{p}(\mathbf{\Theta};|\mathbf{h}^{(n)}|) \in \mathbb{R}^K$. To simplify notation, we write the output of the DNN and its kth component as:

$$\mathbf{p}^{(n)} = \mathbf{p}\left(\mathbf{\Theta};\left|\mathbf{h}^{(n)}\right|\right), \quad p_k^{(n)} := p_k\left(\mathbf{\Theta};\left|\mathbf{h}^{(n)}\right|\right). \tag{2}$$

Unless otherwise noted, we will assume that different training approaches will use the same DNN architecture, so we can better focus on the training approaches itself.

For the SL approach, it is common to minimize the MSE loss, and the resulting training problem is given by:

$$\min_{\mathbf{\Theta}} \quad \sum_{n=1}^{N} \|\mathbf{p}(\mathbf{\Theta};|\mathbf{h}^{(n)}|)-\bar{\mathbf{p}}^{(n)}\|^2 := f_{\text{sup}}(\mathbf{\Theta}) \tag{3}$$
$$\text{s.t.} \quad \mathbf{0} \le \mathbf{p}(\mathbf{\Theta};|\mathbf{h}^{(n)}|) \le \mathbf{P}_{\max}, \forall\, n.$$

On the other hand, UL does not need the labels $\bar{\mathbf{p}}^{(n)}$, and it directly optimizes the sum of the samples' WSR as follows:

$$\min_{\mathbf{\Theta}} \quad \sum_{n=1}^{N} -R\left(\mathbf{p}(\mathbf{\Theta};|\mathbf{h}^{(n)}|),|\mathbf{h}^{(n)}|\right) := f_{\text{unsup}}(\mathbf{\Theta}) \tag{4}$$
$$\text{s.t.} \quad \mathbf{0} \le \mathbf{p}(\mathbf{\Theta};|\mathbf{h}^{(n)}|) \le \mathbf{P}_{\max}, \forall\, n.$$

Remark 1. *Problem* (4) *provides a reasonable formulation as it directly stems from the WSR maximization* (1)*. However, this problem can be much harder to optimize compared with* (1) *because of the following: i) Each* $R\left(\mathbf{p}(\mathbf{\Theta};|\mathbf{h}^{(n)}|),|\mathbf{h}^{(n)}|\right)$ *is a composition of two non-trivial nonlinear functions,* $R(\cdot;|\mathbf{h}|)$ *and* $\mathbf{p}(\cdot;|\mathbf{h}|)$*; ii) It finds a single parameter* $\mathbf{\Theta}$ *that maximizes the sum of the WSR across all snapshots, so it couples* N *difficult problems.* ■

III. A STUDY OF SL AND UL APPROACHES

Are there any fundamental differences between these two popular training approaches? This section provides a number of different ways to address this question. Please note that due to space limitation, all proofs in this section will be relegated to the online version [15].

Comparing SL and UL Approaches. Before we start, we use a simple example to illustrate the potential performance

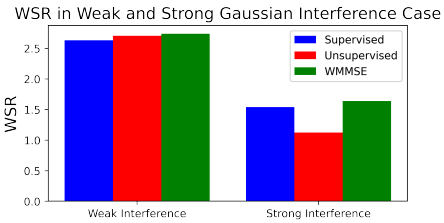


Fig. 1. Comparison between SL, UL and WMMSE in testing time, when SL, UL are trained using data where the interference channel power is equal to direct channel power (weak interference), or 10 times of the direct channel power (strong interference) when there are 10 users. In strong interference case, SL can achieve 92% of the WMMSE sum-rate, while UL achieves relatively lower sum-rate.

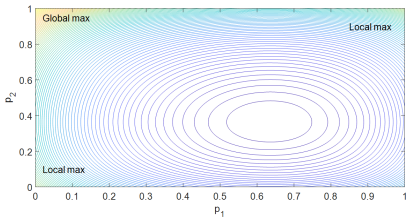


Fig. 2. For two user IC with 2 snapshots, the true label $\bar{\mathbf{p}}^{(1)} = (0,1)$, $\bar{\mathbf{p}}^{(2)} = (1,0)$. Keep the sum of label of the two snapshots to be the 1, i.e., $\bar{\mathbf{p}}^{(1)} = (p_1, 1-p_1)$, $\bar{\mathbf{p}}^{(2)} = (p_2, 1-p_2)$ and plot the sum-rate of the two snapshots. The upper right and lower left corners are local maximums while the upper left is the the global maximum.

difference of the two training approaches. Specifically, Fig. 1 shows that for a 2-user network with different interference situation, the DNN generated by SL and UL can have significantly different test-time performance.

To understand such a phenomenon, let us examine the two optimization problems (3) and (4). From Remark 1, we know that problem (4) can be challenging because the complicated relationship between R and $\mathbf{\Theta}$, and because there are multiple components in the objective. For now, let us focus on cases where one factor is dominating. Suppose $K = 2$ (two user), $w_k = 1, \forall\ k$ (equal weights), and use a linear network to parameterize $\mathbf{p}$: $\mathbf{p} = \mathbf{\Theta}|\mathbf{h}|$, where $\mathbf{\Theta} \in \mathbb{R}^{K \times K^2}$, and $\mathbf{\Theta} := [\mathbf{\Theta}_1; \cdots; \mathbf{\Theta}_K]$, with $\mathbf{\Theta}_k := \{\mathbf{\Theta}_{k,(uv)}\}_{(uv)\in W} \in \mathbb{R}^{1\times K^2}$, where $W := \{(i,j) : i,j \in \{1,\cdots,K\}$ is a set of index tuples. In this case, from the classical results for 2-user IC [16], [17], we know that for each sample n, the sum rate maximization problem (3) is easy to solve, and the solution will be binary. Further, the linear network significantly simplifies the relation between $\mathbf{p}$ and $\mathbf{\Theta}$. Under this setting, we have the following observation.

Claim 1. *Consider the simple SISO-IC case with two users and two samples (i.e., $K = 2$, $N = 2$); let $P_{\max} = 1$, $\sigma = 1$, and suppose a linear network is used: $\mathbf{p}(\mathbf{\Theta}; |\mathbf{h}|) = \mathbf{\Theta}|\mathbf{h}|$. If we use the UL loss (4), then there exist some channel realizations $\mathbf{h}^{(1)} \in \mathcal{C}^{2\times 2}$ and $\mathbf{h}^{(2)} \in \mathcal{C}^{2\times 2}$ whose true labels are $\bar{\mathbf{p}}^{(1)} = (0,1)$, $\bar{\mathbf{p}}^{(2)} = (1,0)$, for which problem (4) has at least two stationary solutions $\mathbf{\Theta}_{\text{global}}$ and $\mathbf{\Theta}_{\text{local}}$. However, these two solutions generate different labels:*

$$\mathbf{p}(\mathbf{\Theta}_{\text{global}}, |\mathbf{h}^{(1)}|) = (0,1),\ \mathbf{p}(\mathbf{\Theta}_{\text{global}}, |\mathbf{h}^{(2)}|) = (1,0), \tag{5}$$

$$\mathbf{p}(\mathbf{\Theta}_{\text{local}}, |\mathbf{h}^{(1)}|) = \mathbf{p}(\mathbf{\Theta}_{\text{local}}, |\mathbf{h}^{(2)}|) = (1,0). \tag{6}$$

On the other hand, if the SL loss (3) is used, then $f_{\text{sup}}(\mathbf{\Theta})$ is a convex function w.r.t. $\mathbf{\Theta}$, and the problem only has a single optimal solution satisfying (5).

This result illustrates that when multiple channel realizations are directly and jointly optimized using UL, it is more likely to possess bad local minima; see Fig 2.

Next, we analyze more general cases. Towards this end, we first investigate the relationship between stationary solutions of the SL problem (3) and the UL problem (4).

Claim 2. *Consider an SISO-IC training problem with K users and N training samples. Suppose the following hold:*
i). For each data sample $n \in \{1,\cdots,N\}$, we can generate a stationary solution $\bar{\mathbf{p}}^{(n)}$ of (1)as the training label .
ii). Let $\mathbf{\Theta}^*(\bar{\mathbf{p}})$ denote the optimal solution for the SL problem (3) with label $\bar{\mathbf{p}}$, and it achieves zero loss: $f_{\text{sup}}(\mathbf{\Theta}^*(\bar{\mathbf{p}})) = 0$.
iii) The solution $\mathbf{\Theta}^*(\bar{\mathbf{p}})$ can be computed for all $\bar{\mathbf{p}}$.
Let $\mathcal{B}$ denote the set of stationary points of (4). Then the following holds:

$$\{\mathbf{\Theta}^*(\bar{\mathbf{p}}) \mid \bar{\mathbf{p}}^n \text{ is a stationary solution of (1)}, \forall\ n\} \subseteq \mathcal{B}. \tag{7}$$

Intuitively, this result shows that if we impose some additional assumptions to the SL approach (i.e., good labels, zero training loss, and good training algorithm), then it is less likely for SL to be trapped by local minima. Additionally, if each label $\bar{\mathbf{p}}^{(n)}$ exactly maximizes (1), then SL can find a neural network that simultaneously optimizes all training instances. On the other hand, it is difficult to impose favorable assumptions for the UL approach to induce better solution quality. This result is a generalization of Claim 1.

It certainly appears that assumptions *ii)* and *iii)* are stringent. However, recent advances in deep learning suggest that they can be both achieved for certain special neural networks. In particular, the assumption that $f_{\text{sup}}(\mathbf{\Theta}^*) = 0$ has been verified when the neural network is "overparameterized"; see. e.g., [18]. Further, it has been shown in [19], [20] that, gradient descent (GD) can indeed find such a global optimal solution. However, these works cannot be applied to analyze our training problem because they require that the inputs are normalized, and that the outputs are scalars instead of vectors.

In the following, we show that it is possible to construct a special neural network and a training algorithm, such that condition *ii)* and *iii)* in Claim 2 can be satisfied, so that (7) holds true. Our result extends the recent work [21].

To proceed, consider an L-layer fully connected network with activation function denoted by $f : \mathbb{R} \to \mathbb{R}$. The weights

of each layer are $(W_l)_{l=1}^L$. Let $\|\cdot\|_F$ denote the Frobenius norm and $\|\cdot\|_2$ denote the L_2 norm. The input and output of the network (across all samples) are $\mathbf{h} \in \mathbb{R}^{N\times K^2}$ and $\mathbf{p} \in \mathbb{R}^{N\times K}$, respectively. Let the output of the l-th layer (across all samples) be $F_l \in \mathbb{R}^{N\times n_l}$, which can be expressed as:

$$F_l = \begin{cases} \mathbf{h} & l = 0 \\ \sigma\left(F_{l-1}W_l\right) & l \in [1:L-1] \\ F_{L-1}W_L & l = L \end{cases} \tag{8}$$

where σ is some activation function. In our problem setting, the output of the neural network is the power allocation vector, therefore $n_L = K$. Let us vectorize the output of each layer by concatenating each of its column, and denote it as $f_l = \mathrm{vec}\left(F_l\right) \in \mathbb{R}^{Nn_l}$. Similarly, denote the vectorized label as $y = \mathrm{vec}(\mathbf{p}) \in \mathbb{R}^{NK}$. At m-th iteration of training, we use $\Theta^m = (W_l^m)_{l=1}^L$ to denote all the parameters.

Let us define the following quantities, which are related to the singular values of weight matrices at initialization:

$$\bar{\lambda}_l = \begin{cases} \frac{2}{3}\left(1 + \left\|W_l^0\right\|_2\right), & \text{for } l \in \{1,2\}, \\ \left\|W_l^0\right\|_2, & \text{for } l \in \{3,\ldots,L\}, \end{cases} \tag{9}$$

and $\lambda_l = \sigma_{\min}\left(W_l^0\right), \lambda_{i\to j} = \prod_{l=i}^j \lambda_l, \bar{\lambda}_{i\to j} = \prod_{l=i}^j \bar{\lambda}_l$ and $\lambda_F = \sigma_{\min}\left(\sigma\left(XW_1^0\right)\right)$, where $\sigma_{\min}(A)$ and $\|A\|_2$ are the smallest and largest singular value of matrix A.

Let us make the following assumptions about the neural network structure as well as the activation function.

Assumption 1. *(Pyramidal Network Structure) Let $n_1 \geq N$ and $n_2 \geq n_3 \geq \ldots \geq n_L$.*

This assumption defines the Pyramidal Network structure [21], which consists of one wide layer(i.e the number of neurons is at least the sample size) but no comparison between n_1 and n_2 is needed.

Assumption 2. *There exist constants $\gamma \in (0,1)$ and $\beta > 0$, such that the activation function $\sigma(\cdot)$ satisfies:*

$$\sigma'(x) \in [\gamma, 1], \quad |\sigma(x)| \leq |x|, \ \forall\, x \in \mathbb{R}, \quad \sigma' \ \text{ is } \beta\text{-Lipschitz.}$$

In [21], a concrete example is shown that satisfies Assumption 2:

$$\sigma(x) = -\frac{(1-\gamma)^2}{2\pi\beta} + \frac{\beta}{1-\gamma}\int_{-\infty}^{\infty} \max(\gamma u, u) e^{-\frac{x\beta^2(x-u)^2}{(1-\gamma)^2}} du$$

Next we discuss how to train such a network using the SL and UL approaches. Towards this end, we need to fix a training algorithm. Different than the conventional neural network training, problems (3) – (4) has n constraints (one for each sample), and it is difficult for conventional gradient-based algorithms to enforce them. To overcome such a difficulty, we adopt the following approaches.

For the SL training, we will directly consider the unconstrained version of (3) (by removing all power constraints). This is acceptable because, *if* zero training loss can be achieved, and if all the labels are feasible, then the output for each sample will be automatically feasible. However, for the UL training, we cannot simply drop the constraints because we do not have labels. Therefore, we choose to add a sigmoid function to the last layer of output to enforce feasibility. Specifically, the modified network has the following (vectorized) output:

$$F_L = \mathrm{sig}(F_{L-1}W_L) = \frac{\mathbf{1}\times P_{\max}}{1 + e^{-F_{L-1}W_L}}. \tag{10}$$

Now that both training problems become unconstrained, we can use the conventional gradient-based algorithms. We have the following convergence results.

Claim 3. *Consider an SISO-IC training problem with K users and N training samples. Let $P_{max} = 1$. Construct a fully connected neural network satisfying Assumption 1 - 2. Initialize $\mathbf{\Theta}^0$ so that it satisfies [21, Assumption 3.1]. Then the following holds:*
(a) Consider optimizing the unconstrained version of (3) *using the following gradient descent algorithm*

$$\mathbf{\Theta}^{m+1} = \mathbf{\Theta}^m - \eta\nabla f_{\mathrm{sup}}(\mathbf{\Theta}^m).$$

There exists constant stepsize η such that the training loss converges to zero at a geometric rate, that is:

$$f_{\mathrm{sup}}\left(\mathbf{\Theta}^m\right) \leq \left(1 - \eta\alpha_0\right)^m f_{\mathrm{sup}}\left(\mathbf{\Theta}^0\right) \tag{11}$$

where $\alpha_0 = \frac{4}{\gamma^4}\left(\frac{\gamma^2}{4}\right)^L \lambda_F^2\lambda_{3\to L}^2$.
(b) Consider minimizing the unconstrained version of (4) *using the last layer as* (10) *and use the following gradient descent algorithm*

$$\mathbf{\Theta}^{m+1} = \mathbf{\Theta}^m - \eta\nabla f_{\mathrm{unsup}}(\mathbf{\Theta}^m).$$

Suppose all the weights are bounded during training, then $\mathbf{\Theta}$ will converge to a stationary point of the training objective.

Claim 3-(a) indicates that when the neural network satisfied Assumptions 1 – 2, and with some special initialization, then conditions (ii) – (iii) in Claim 2 can be satisfied, so the conclusion in Claim 2 holds. On the other hand, for UL, the best one can say is that a stationary solution for the training problem is obtained. No global optimality can be claimed, nor any convergence rate analysis can be done. Intuitively, this result again says one can identify sufficient conditions that SL can perform well, while the UL approach is much more challenging to analyze. We note that the analysis of Claim 3-(a) follows similar approaches as [21, Theorem 3.2]. However, Claim 3-(b) is different since we need to analyze the special network with the sigmoid activation function.

Impact of Label Quality. The above results show different objective functions can have different performance in maximizing the sum rate. Next, we show an additional property about the SL approach – that the *quality of labels* can affect training efficiency. Intuitively, it is reasonable to believe that neural networks trained using high-quality labeled data can achieve higher sum rate compared with those trained with

Quality \ # samples	30,000	40,000	50,000
Low	1.38 (83.6%)	1.38 (83.6%)	1.39 (84.2%)
High	1.72 (92.0%)	1.76 (94.1%)	1.78 (95.2%)

Quality \ # samples	50,000	100,000	200,000
Low	1.11 (59.0%)	1.32 (70.2%)	1.39 (73.9%)
High	1.31 (65.6%)	1.55 (77.5%)	1.74 (87.0%)

Table I. Comparison between using high-quality labels and low-quality labels in SL. The top (resp. bottom) table shows the $K = 10$ (resp. $K = 20$) case. The number in each entry shows the testing performance (in bits/sec), where the model is trained using a fixed number of training sample (shown at the first row), with either low or high quality labels. The percentages in the table means the relative sum rate achieved at testing time v.s. what is achieved by the training labels.

with low-quality labels. To see this, we conduct two simple experiments. We generate two training sets, one with low-quality labels and the other with high-quality labels. The low-quality labels are the power allocations that achieve an average of 1.65 bits/sec (resp. 1.88 bits/sec) for 10 users (resp. for 20 users) case. The high-quality labels are the power allocations that achieve an average of 1.87 bits/sec (resp. 2.00 bits/sec) for 10 users (resp. for 20 users) case. We use different number of samples to train the network, derive the sum rate using test samples and compare the result to the corresponding sum rate achieved by the given labels; the results are shown in Table I. We see that for a particular setting, using high-quality label not only achieves higher absolute sum rate, but also higher *relative* sum rate comparing with what can be achieved by the labels.

Below, we argue the benefit of high-quality label from a slightly different perspective – the label quality can influence the convergence speed of training algorithm.

Claim 4. *Suppose* $(\mathbf{h}, \mathbf{p})$ *and* $(\mathbf{h}^{'}, \mathbf{p}^{'})$ *are two sets of data, each consists of* N *samples, and* $\mathbf{h}^{'} = \mathbf{h}$*. Suppose for each* n*,* $\mathbf{p}^{(n)}$ *is the unique globally optimal power allocation for problem* (1)*, given channel realization* $\mathbf{h}^{(n)}$*. Suppose two samples in* $\mathbf{h}$ *are identical, say,* $\mathbf{h}^{(1)} = \mathbf{h}^{(2)}$*. Construct the labels for* $\mathbf{h}^{'}$ *in the following way*

$$\begin{cases} \mathbf{p}^{',(2)} \neq \mathbf{p}^{(2)}, \\ \mathbf{p}^{',(n)} = \mathbf{p}^{(n)}, \quad \forall\ n,\ \text{s.t. } n \neq 2. \end{cases} \tag{12}$$

Under Assumption 1 and [22, Assumption 3.1], use the same training algorithm as Claim 3(a) to optimize the unconstrained version of (3) *using* $(\mathbf{h}, \mathbf{p})$ *and* $(\mathbf{h}^{'}, \mathbf{p}^{'})$ *respectively, at each iteration* m*, there exist* β_m *and* $\beta_m^{'}$ *that satisfy:*

$$f_{\text{sup}}(\mathbf{\Theta}^{m+1}) \leq \beta_m f_{\text{sup}}(\mathbf{\Theta}^{m}), \tag{13}$$

$$f_{\text{sup}}(\mathbf{\Theta}^{',m+1}) \leq \beta_m^{'} f_{\text{sup}}(\mathbf{\Theta}^{',m}). \tag{14}$$

Further, we have $\beta_m < \beta_m^{'}$*, that is, the problem with the correct label can be trained faster.*

In our analysis, we combined the pyramid network analysis with the decomposition technique from [23]. This result uses a simple construction to reveal the importance of *consistency* of labels among "similar" samples. Intuitively, it somewhat explains why in Table I, the models trained by high-quality labels can achieve higher percentage of the rates. The reason may be that when the quality of the label is better, the training speed is also faster.

IV. A SEMI-SUPERVISED LEARNING REMEDY FOR POWER ALLOCATION

From the previous section, we know that under a few assumptions, especially when high-quality labels are available, SL could perform better than the UL. However, one drawback of the SL approach is that finding high-quality labels can be costly. Is there a way to design a proper learning strategy that only requires a few labels, while still achieving the state-of-the-art training and testing performance? In this section, we address this by proposing a *semi-supervised* learning strategy which combines both the SL and UL approaches in (3) – (4).

As indicated by Claim 1, UL may get stuck at some local solutions once parameters enter some "bad" regions. To alleviate such a "bad" local minimum issue, we propose to add some regularization in the training objective, which in fact changes the landscape of loss function. More specifically, we propose to use some label-dependent regularizations that utilizes the available high-quality labels. Suppose there are N unlabeled samples denoted as $\{|\mathbf{h}^{(n)}|\}$ and M labeled samples denoted as $\{|\mathbf{h}^{(m)}|, \bar{\mathbf{p}}^{(m)}\}$, and $M \ll N$ (few labeled samples). Then, the final objective function to be maximized is:

$$\begin{aligned} \max_{\mathbf{\Theta}} \ & \sum_{n=1}^{N} R\left(\mathbf{p}(\mathbf{\Theta}; |\mathbf{h}^{(n)}|), |\mathbf{h}^{(n)}|\right) - \\ & \lambda \sum_{m=1}^{M} \left\| \mathbf{p}(\mathbf{\Theta}; |\mathbf{h}^{(m)}|) - \bar{\mathbf{p}}^{(m)} \right\|^2, \end{aligned} \tag{15}$$

where $\lambda > 0$ specifies the trade-off between the sumrate loss (unsupervised) and squared loss (supervised). In classical semi-supervised learning, *cluster assumption* [24] is often included, which means samples with same label belong to the same class. The regularization term above, will serve as a clustering classifier which clusters the same label $\mathbf{p}$ to one class.

V. SIMULATION RESULTS

V-A. Data Generation

The Rayleigh fading channel model [25] is considered in the simulation and the number of users is 5, 10 or 20. Direct channels h_{kk} and interfering channels $h_{kj}, k \neq j$ are generated from zero-mean complex Gaussian distribution $\mathcal{CN}(0, \sigma^2)$, where σ denotes the standard deviation. To evaluate the stability of different learning approaches, two

representative cases are considered. In the first case (referred as weak interference case), both direct and interfering channels are generated from the same complex Gaussian distribution with $\sigma = 1$. For the second case (referred as strong interference case), direct channels have the same setting as in the first case while the standard deviation of interfering channels is 10 times of the the direct channels.

V-B. Neural Network Structure

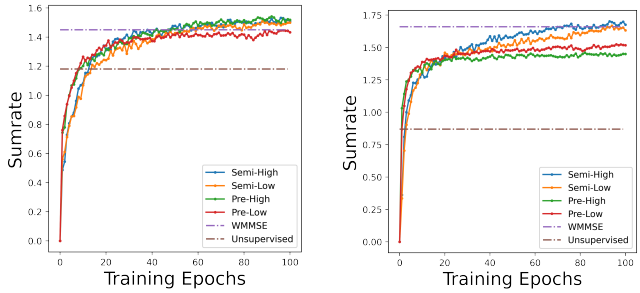
A fully connected neural network with 3 hidden layers is used in the simulation. The number of neurons in each hidden layers are $200, 80, 80$ for 5 and 10 user case and $600, 200, 200$ for 20 user case, respectively. The activation function of the hidden layers is *ReLu* function and *Sigmoid* function is used at output layer. To stabilize the training process, *Batch Normalization* [26] is used after each hidden layer.

V-C. Benchmarks and Training Strategy

We compare UL (see (4)), semi-supervised learning with pre-training and semi-supervised learning with squared loss regularization and $\lambda = 1$ (see section IV), and select the WMMSE [12] as the baseline. In order to get higher quality label, we train the network with GNN [4] and fine tune the label using WMMSE. Both labels are used in pre-training and semi-supervised learning. All the three DNN-based learning approaches use the same neural network structure as specified in Section V-B. In the strong interference case, the total number of unlabeled and labeled samples are $50,000$ and 400 for 10 user, while $10,000$ and 100 for 5 user, respectively. Unlabeled samples are used in UL approach, which together with the labeled samples are used in the two semi-supervised learning approaches. Also, we change the number of labeled data and train the model along with the unlabeled data. RMSprop [27] is used as the optimizer, within each batch, the number of unlabeled data is 200 addition with the fixed labeled samples. For weak interference case, since UL can already work well with fewer samples, the total number of unlabeled and labeled samples used in training are $20,000$ and 100, respectively. The batch size is 200 with 200 unlabeled samples and additional 100 labeled samples as regularization. Other setups are the same as used in the strong interference case and we compare UL and proposed semi-supervised learning. To evaluate the performance, 1000 additional unlabeled samples are generated and their averaged sumrate is used as performance metric.

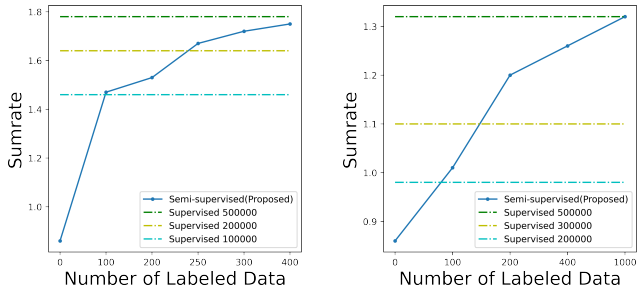
V-D. Simulation Results and Analysis

The sumrate of the UL and the two semi-supervised learning approaches in the strong interference case are compared in Fig. 3. Compare with the UL, the proposed semi-supervised learning with squared loss regularization approach significantly improves the sum-rate in 10 user case. However, the pre-training approach does not bring significant improvement in sumrate. One possible reason is that only a few labeled samples are not enough to pre-train a 'good' initialization. The gradually increased labeled data can improve the performance of semi-supervised learning in high interference scenario, and higher-quality label can produce better performance. In weak interference scenario, the performance of semi-supervised learning is similar to unsupervised learning. So in this case, regularization is actually not needed. This is also a potential direction in future work. In the scenario where UL can already work, is there a way that labeled data can improve the performance?



(a) Strong interference with K=5 (b) Strong Interference with K=10

Fig. 3. Comparison between proposed semi-supervised learning, pre-training, unsupervised learning and WMMSE uner strong interference case in sum-rate maximization.



(a) Strong interference case $K = 10$. (b) Strong interference case $K = 20$.

Fig. 4. Comparison between using different number of (high-quality) labeled data in proposed semi-supervised learning.

Method \ User Number	K=5	K=10
Semi-supervised	2.09 (bits/sec)	2.60 (bits/sec)
Unsupervised	2.09 (bits/sec)	2.64 (bits/sec)
WMMSE	2.06 (bits/sec)	2.74 (bits/sec)

Table II. For weak interference scenario, compare the performance of unsupervised learning and proposed semi-supervised learning both using $20,000$ samples, and semi-supervised learning with 100 additional labeled data.

VI. REFERENCES

[1] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[2] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 119–131, 2018.

[3] M. Eisen and A. R. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Transactions on Signal Processing*, 2020.

[4] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," in *proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps)*.

[5] T. Maksymyuk, J. Gazda, O. Yaremko, and D. Nevinskiy, "Deep learning based massive MIMO beamforming for 5G mobile network," in *2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, 2018, pp. 241–244.

[6] A. Bashar, G. Parr, S. McClean, B. Scotney, and D. Nauck, "Machine learning based call admission control approaches: A comparative study," in *2010 International Conference on Network and Service Management*, 2010, pp. 431–434.

[7] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2017, pp. 1–5.

[8] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1760–1776, 2019.

[9] Z.-Q. Luo and S. Zhang, "Dynamic spectrum management: Complexity and duality," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 1, pp. 57–73, 2008.

[10] G. Scutari, D. P. Palomar, and S. Barbarossa, "The MIMO iterative waterfilling algorithm," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 1917–1935, 2009.

[11] D. A. Schmidt, C. Shi, R. A. Berry, M. L. Honig, and W. Utschick, "Distributed resource allocation schemes," *IEEE Signal Processing Magazine*, vol. 26, no. 5, pp. 53–63, 2009.

[12] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[13] J. Papandriopoulos and J. S. Evans, "Scale: A low-complexity distributed protocol for spectrum balancing in multiuser dsl networks," *IEEE Transactions on Information Theory*, vol. 55, no. 8, pp. 3711–3724, 2009.

[14] W. Lee, M. Kim, and D.-H. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1276–1279, 2018.

[15] B. Song, H. Sun, W. Pu, S. Liu, and M. Hong. To supervise or not to supervise: How to effectively learn wireless interference management models? [Online]. Available: http://people.ece.umn.edu/~mhong/mingyi.html

[16] G. O. A. Gjendemsjo, D. Gesbert and S. Kiani, "Optimal power allocation and scheduling for two-cell capacity maximization," in *IEEE WiOpt (Workshop on Resource Allocation in Wireless Networks)*, 2006, pp. 1–5.

[17] M. Charafeddine and A. Paulraj, "Maximum sum rates via analysis of 2-user interference channel achievable rates region," in *43rd Annual Conference on Information Sciences and Systems*, march 2009, pp. 170 –174.

[18] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[19] Z. Allen-Zhu, Y. Li, and Z. Song, "A convergence theory for deep learning via over-parameterization," in *International Conference on Machine Learning*. PMLR, 2019, pp. 242–252.

[20] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 1675–1685.

[21] Q. Nguyen and M. Mondelli, "Global convergence of deep networks with one wide layer followed by pyramidal topology," *arXiv preprint arXiv:2002.07867*, 2020.

[22] N. T. H. Phuong and H. Tuy, "A unified monotonic approach to generalized linear fractional programming," *Journal of Global Optimization*, vol. 26, pp. 229–259, 2003.

[23] S. Arora, S. Du, W. Hu, Z. Li, and R. Wang, "Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 322–332.

[24] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[25] B. Sklar, "Rayleigh fading channels in mobile digi-

tal communication systems. i. characterization," *IEEE Communications magazine*, vol. 35, no. 7, pp. 90–100, 1997.

[26] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[27] T. Tieleman and G. Hinton, "Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *COURSERA Neural Networks Mach. Learn*, 2012.

VII. APPENDIX

VII-A. Proof for Theorems

Claim 5. *Suppose $K = 2$, $N = 2$ (two user and two samples), and $P_{\max} = 1$. Consider the unsupervised training loss* (4)*, where the predicted power is expressed as*

$$\mathbf{p}(\mathbf{\Theta}; \mathbf{h}) = \sigma(\mathbf{\Theta}\mathbf{h}), \tag{16}$$

where $\sigma(z) = \max\{0, z\} \in \mathbb{R}^K$ is the ReLu activation function. Then there exists some channel realizations $\mathbf{h}^{(1)} \in \mathcal{C}^{2\times 2}$ and $\mathbf{h}^{(2)} \in \mathcal{C}^{2\times 2}$ whose true labels are $\bar{\mathbf{p}}^{(1)} = (0, 1)$, $\bar{\mathbf{p}}^{(1)} = (1, 0)$, for which problem (4) *has at least two stationary solutions $\mathbf{\Theta}_{\text{global}}$ and $\mathbf{\Theta}_{\text{local}}$, and they satisfy the following*

$$\mathbf{p}(\mathbf{\Theta}_{\text{global}}, \mathbf{h}^{(1)}) = (0, 1),\ \mathbf{p}(\mathbf{\Theta}_{\text{global}}, \mathbf{h}^{(2)}) = (1, 0) \tag{17}$$

$$\mathbf{p}(\mathbf{\Theta}_{\text{local}}, \mathbf{h}^{(1)}) = \mathbf{p}(\mathbf{\Theta}_{\text{local}}, \mathbf{h}^{(2)}) = (1, 0). \tag{18}$$

On the other hand, considering using the supervised loss (3)*, with $\mathbf{p}(\theta; \mathbf{h})$ given as in* (16)*, and $\bar{\mathbf{p}}^{(n)}$ computed as the optimal solution of* (1) *(by using techniques in [16], [17]). Then the problem only has a single optimal solution satisfying* (17)

Proof. The objective value is

$$f = \log\left(1 + \frac{|h_{11}^{(1)}|^2 p_1^{(1)}}{|h_{12}^{(1)}|^2 p_2^{(1)}+1}\right) + \log\left(1 + \frac{|h_{22}^{(1)}|^2 p_2^{(1)}}{|h_{21}^{(1)}|^2 p_1^{(1)}+1}\right) + \log\left(1 + \frac{|h_{11}^{(2)}|^2 p_1^{(2)}}{|h_{12}^{(2)}|^2 p_2^{(2)}+1}\right) + \log\left(1 + \frac{|h_{22}^{(2)}|^2 p_2^{(2)}}{|h_{21}^{(2)}|^2 p_1^{(2)}+1}\right)$$

If take derivative over the any weight θ_{kj} related to the power of the first user, the derivative is If take the derivative over $\mathbf{\Theta}_{kj}$, $j = 1, 2, 3, 4$, $k = 1, 2$
The gradient is derived as

$$\frac{\partial f_{\text{unsup}}}{\partial \mathbf{\Theta}_{kj}} = \frac{\partial f_{\text{unsup}}}{\partial p_k^{(1)}} \cdot \frac{\partial p_k^{(1)}}{\partial \mathbf{\Theta}_{kj}} + \frac{\partial f_{\text{unsup}}}{\partial p_k^{(2)}} \cdot \frac{\partial p_k^{(2)}}{\partial \mathbf{\Theta}_{kj}}$$

where

$$\begin{cases} \frac{\partial f_{\text{unsup}}}{\partial p_1^{(n)}} = \frac{|h_{11}^{(n)}|^2}{|h_{11}^{(n)}|^2 p_1^{(n)}+|h_{12}^{(n)}|^2 p_2^{(n)}+1} - \frac{|h_{21}^{(n)}|^2 |h_{22}^{(n)}|^2 p_2^{(n)}}{\left(|h_{21}^{(n)}|^2 p_1^{(n)}+|h_{22}^{(n)}|^2 p_2^{(n)}+1\right)\left(|h_{21}^{(n)}|^2 p_1^{(n)}+1\right)} \\ \frac{\partial f_{\text{unsup}}}{\partial p_2^{(n)}} = \frac{h_{22}^{(n)2}}{h_{21}^{(n)2} p_1^{(n)}+h_{22}^{(n)2} p_2^{(n)}+1} - \frac{h_{11}^{(n)2} h_{12}^{(n)2} p_1^{(n)}}{\left(h_{12}^{(n)2} p_2^{(n)}+h_{11}^{(n)2} p_1^{(n)}+1\right)\left(h_{12}^{(n)2} p_2^{(n)}+1\right)} \end{cases} \tag{19}$$

and

$$\frac{\partial p_k^{(1)}}{\partial \mathbf{\Theta}_{k,(uv)}} = |h_{uv}^{(1)}|, \quad \frac{\partial p_k^{(2)}}{\partial \mathbf{\Theta}_{k,(uv)}} = |h_{uv}^{(2)}|,\ \forall\ (u, v) \in W, \forall\ k.$$

We claim that if we find $\delta_1, \delta_2, \epsilon_1, \epsilon_2$ which satisfies the following two conditions, then the gradient descent method optimizing the neural network weights will lead the predicted label to the above local maximum.
Condition 1:

$$\frac{\partial f_{\text{unsup}}}{\partial p_1^{(n)}} < 0 \quad n = 1, 2 \tag{20}$$

Condition2:

$$\frac{\partial f}{\partial \theta_{2j}} < 0 \quad n = 1, 2 \tag{21}$$

This is true because when Condition1 is satisfied, the gradient over every θ_{kj} is positive, so the power of the first user will always converge to 1. Similarly, when Condition2 is satisfied, the weight θ_{2j} relates to the power of the second user will always decrease. Even when the power of the second user in data $\mathbf{h}^{(2)}$ reaches 0 so the second term reaches minimum and the gradient vanishes, it can still happen that for some channel realization $\mathbf{h}^{(1)}$, the true label of which is $(0, 1)$, will finally converge to $(1, 0)$ as long as the first term is always negative in this region. This is easier to happen when the interference channel is strong.

Now we illustrate that the above two conditions is easy to satisfy.
For Condition1, there is

$$\frac{{h_{21}^{(2)}}^2 {h_{22}^{(2)}}^2 \delta_2}{({h_{21}^{(2)}}^2(1-\delta_1)+{h_{22}^{(1)}}^2\delta_2+1)({h_{21}^{(1)}}^2(1-\delta_1)+1)} < {h_{21}^{(2)}}^2 {h_{22}^{(2)}}^2 \delta_2$$

and

$$\frac{{h_{11}^{(1)}}^2}{{h_{11}^{(1)}}^2(1-\delta_1)+{h_{12}^{(1)}}^2\delta_2+1} > \frac{{h_{11}^{(1)}}^2}{{h_{11}^{(1)}}^2+{h_{12}^{(1)}}^2+1}$$

So it's sufficient to satisfy

$$\frac{{h_{11}^{(1)}}^2}{{h_{11}^{(1)}}^2+{h_{12}^{(1)}}^2+1} > {h_{21}^{(1)}}^2 {h_{22}^{(1)}}^2 \delta_2$$

Thus we can derive

$$\delta_2 < \frac{{h_{11}^{(1)}}^2}{\left({h_{11}^{(1)}}^2+{h_{12}^{(2)}}^2+1\right){h_{21}^{(1)}}^2 {h_{22}^{(1)}}^2}$$

Similarly, we can derive

$$\epsilon_2 < \frac{{h_{11}^{(2)}}^2}{\left({h_{11}^{(2)}}^2+{h_{12}^{(1)}}^2+1\right){h_{21}^{(2)}}^2 {h_{22}^{(2)}}^2}$$

Now we derive ϵ_1 and δ_1 to satisfy Condition2. Denote

$$d = \max_{u,v} \frac{h_{uv}^{(2)}}{h_{uv}^{(1)}}$$

Let

$$\epsilon_1 = 0.1$$

$$\epsilon_2 = \min\left(\frac{1}{{h_{12}^{(2)}}^2}, \frac{{h_{11}^{(2)}}^2}{\left({h_{11}^{(2)}}^2+1\right){h_{21}^{(2)}}^2 {h_{22}^{(2)}}^2}\right)$$

There is

$$\frac{{h_{22}^{(1)}}^2}{{h_{21}^{(1)}}^2(1-\delta_1)+{h_{22}^{(1)}}^2\delta_2+1} - \frac{{h_{11}^{(1)}}^2 {h_{12}^{(1)}}^2(1-\delta_1)}{\left({h_{12}^{(1)}}^2\delta_2+{h_{11}^{(1)}}^2(1-\delta_1)+1\right)\left({h_{12}^{(1)}}^2\delta_2+1\right)} < {h_{22}^{(1)}}^2$$

In this case

$$\begin{aligned}
&\frac{{h_{22}^{(2)}}^2}{{h_{21}^{(2)}}^2(1-\epsilon_1)+{h_{22}^{(2)}}^2\epsilon_2+1} - \frac{{h_{11}^{(2)}}^2 {h_{12}^{(2)}}^2(1-\epsilon_1)}{\left({h_{12}^{(2)}}^2\epsilon_2+{h_{11}^{(2)}}^2(1-\epsilon_1)+1\right)\left({h_{12}^{(2)}}^2\epsilon_2+1\right)} \\
&< \frac{{h_{22}^{(2)}}^2}{0.9{h_{21}^{(2)}}^2+{h_{22}^{(2)}}^2\epsilon_2+1} - \frac{{h_{11}^{(2)}}^2 {h_{12}^{(2)}}^2 \times 0.9}{\left(0.9{h_{11}^{(2)}}^2+2\right)\times 2} \\
&< \frac{{h_{22}^{(2)}}^2}{0.9{h_{21}^{(2)}}^2+1} - 0.45 \times \frac{{h_{11}^{(2)}}^2 {h_{12}^{(2)}}^2}{0.9{h_{11}^{(2)}}^2+2}
\end{aligned}$$

Since the formulation is monotone about ${h_{12}^{(2)}}^2$ and ${h_{22}^{(2)}}^2$, so we can make ${h_{12}^{(2)}}^2$ large and ${h_{22}^{(2)}}^2$ small to satisfy the condition

$$\begin{aligned}
&\frac{{h_{22}^{(2)}}^2}{{h_{21}^{(2)}}^2(1-\epsilon_1)+{h_{22}^{(2)}}^2\epsilon_2+1} - \frac{{h_{11}^{(2)}}^2 {h_{12}^{(2)}}^2(1-\epsilon_1)}{\left({h_{12}^{(2)}}^2\epsilon_2+{h_{11}^{(2)}}^2(1-\epsilon_1)+1\right)\left({h_{12}^{(2)}}^2\epsilon_2+1\right)} \\
&< -\frac{d}{h_{22}^2}
\end{aligned}$$

Thus Condition2 can be satisfied.
For fixed δ_1, δ_2, there always exists $h_{12}^{(1)^2} = h_{21}^{(1)^2} >> h_{22}^{(1)^2} > h_{11}^{(1)^2}$, $h_{12}^{(2)^2} = h_{21}^{(2)^2} >> h_{11}^{(2)^2} > h_{22}^{(2)^2}$, for which it's easy to verify that $p_1^{(1)} = 0, p_2^{(1)} = 1$ is the true label, but

$$\frac{h_{22}^{(2)^2}}{h_{21}^{(2)^2}(1-\delta_1) + h_{22}^{(2)^2}\delta_2 + 1} - \frac{h_{11}^{(2)^2} h_{12}^{(2)^2}(1-\delta_1)}{\left(h_{12}^{(2)^2}\delta_2 + h_{11}^{(2)^2}(1-\delta_1) + 1\right)\left(h_{12}^{(2)^2}\delta_2 + 1\right)} < 0$$

In this case, even if the power of the second user in the second data point reaches minimum, the gradient of power of the second user in the data point $\mathbf{H}$ will always smaller than 0 and converge to 0. □

Claim 6. *Consider an SISO-IC training problem with K users and N training samples. Suppose $P_{max} = 1$. Given a fully connected neural network satisfies Assumption 1 - 2. Initialize Θ^0 so that it satisfies [21, Assumption 3.1]. Then the following holds:*
(a) If we optimize (3) *using Gradient Descent*

$$f(\mathbf{\Theta}^{m+1}) = f(\mathbf{\Theta}^m) - \eta \nabla f_{\text{sup}}(\Theta^m)$$

Then there exists constant setpsize η such that it can be ensured the training loss converges to zero at a geometric rate as:

$$f_{\text{sup}}\left(\mathbf{\Theta}^m\right) \leq \left(1 - \eta\alpha_0\right)^m f_{\text{sup}}\left(\mathbf{\Theta}^0\right) \tag{22}$$

where $\alpha_0 = \frac{4}{\gamma^4}\left(\frac{\gamma^2}{4}\right)^L \lambda_F^2 \lambda_{3\to L}^2$.
(b) minimize the unsupervised loss (4) *with an extra sigmoid activation at the last output layer to constrain the power in $[0, P_{max}]$, and suppose all the weights are bounded during training, performing Gradient Descent can only ensure that $\mathbf{\Theta}$ will converge to a stationary point.*

Proof. Part (a) is a directly from [21], for Part (b) derive the proof as following: By Lemma 4.1 in [21],we have

$$\text{vec}\left(\nabla_{W_l}\Phi\right) = \left(\mathbb{I}_{n_l} \otimes F_{l-1}^T\right) \prod_{p=l+1}^{L} \Sigma_{p-1}\left(W_p \otimes \mathbb{I}_N\right) \Sigma_L\left(f_L - y\right) g$$

where $g = \text{vec}(\frac{\partial f_{\text{unsup}}}{\partial \mathbf{p}})$. We want to apply Lemma 4.3 in [21], so we need to show

$$||\nabla f_{\text{unsup}}(\mathbf{\Theta}_k^t) - \nabla f_{\text{unsup}}(\mathbf{\Theta}_m)||_2 \leq C||\mathbf{\Theta}_m^t - \mathbf{\Theta}_m||_2, \forall t \in [0,1]$$

We have First, bound $||Jf_L(\mathbf{\Theta}_m^t)||$.

$$\left\|Jf_L\left(\mathbf{\Theta}_m^t\right)\right\|_2 \leq \sum_{l=1}^{L}\left\|\frac{\partial f_L\left(\mathbf{\Theta}_m^t\right)}{\partial \,\text{vec}\left(W_l\right)}\right\|_2 \leq \sum_{l=1}^{L}\prod_{p=0}^{L-l-1} ||\Sigma_L\left(W_{L-p}^T \otimes \mathbb{I}_N\right)\Sigma_{L-p-1}\left(\mathbb{I}_{n_l} \otimes F_{l-1}\right)||_2$$

When all the weights are bounded, there is $||Jf_L(\mathbf{\Theta}_m^t)|| \leq C_1$.

$$\left\|Jf_L\left(\mathbf{\Theta}_m^t\right) - Jf_L\left(\mathbf{\Theta}_m\right)\right\|_2 \leq \sum_{l=1}^{L}\left\|\frac{\partial f_L(\mathbf{\Theta}_m^t)}{\partial\,\text{vec}(W_l)} - \frac{\partial f_L(\mathbf{\Theta}_m)}{\partial\,\text{vec}(W_l)}\right\|_2 \leq \sqrt{L}\|X\|_F R\left(1 + L\beta\|X\|_F R\right)\left\|\mathbf{\Theta}_m^t - \mathbf{\Theta}_m\right\|_2 \leq C_2\left\|\mathbf{\Theta}_m^t - \mathbf{\Theta}_m\right\|$$

where $R = \prod_{p=1}^{L}\max\left(1, \bar{\lambda}_p\right)$. We can derive that

$$g(\mathbf{\Theta}) = \left(-\frac{R^{(1)}}{\partial \mathbf{p}_1^{(1)}}, \cdots, -\frac{R^{(N)}}{\partial \mathbf{p}_1^{(N)}}, \cdots, -\frac{R^{(1)}}{\partial \mathbf{p}_{nN_L}^{(1)}}, \cdots, -\frac{R^{(N)}}{\partial \mathbf{p}_{nN_L}^{(N)}}\right)$$

Note that $\frac{\partial R^{(n)}}{\partial \mathbf{p}_i^{(n)}}$ is bounded, so there is $\|g(\mathbf{\Theta}_m)\|_2 \leq C_3$.
Finally, consider $\left\|g(\mathbf{\Theta}_m^t) - g(\mathbf{\Theta}_m)\right\|_2$. Notice that $\|g'(\mathbf{\Theta})\|_2$ is bounded, so there is

$$\left\|g(\mathbf{\Theta}_m^t - g(\mathbf{\Theta}_m))\right\|_2 \leq C_4\left\|\mathbf{\Theta}_m^t - \mathbf{\Theta}_m\right\|_2$$

Then it can be derived

$$\left\|\nabla f_{\text{unsup}}\left(\Theta_k^t\right) - \nabla f_{\text{unsup}}\left(\Theta_m\right)\right\|_2 \leq C_1C_4\left\|\mathbf{\Theta}_m^t - \mathbf{\Theta}_m\right\|_2 + C_2C_3\left\|\mathbf{\Theta}_m^t - \mathbf{\Theta}_m\right\| = \left(C_1C_4 + C_2C_3\right)\left\|\mathbf{\Theta}_m^t - \mathbf{\Theta}_m\right\|$$

By Lemma 4.3 in [21], let $\eta < \frac{1}{C_1C_4+C_2C_3}$ There is

$$f_{\text{unsup}}(\mathbf{\Theta}_{m+1}) \leq f_{\text{unsup}}(\mathbf{\Theta}_m)+\langle \nabla f_{\text{unsup}}(\mathbf{\Theta}_m), \mathbf{\Theta}_{m+1}-\mathbf{\Theta}_m\rangle+\frac{1}{2}(C_1C_2+C_3C_4)\left\|\mathbf{\Theta}_{m+1}-\mathbf{\Theta}_m\right\|_2^2 \leq f_{\text{unsup}}(\mathbf{\Theta}_m)-\frac{1}{2}\eta\left\|\mathbf{\Theta}_{m+1}-\mathbf{\Theta}_m\right\|_2^2$$

□

Claim 7. *Suppose two sets of channel realizations $\mathbf{h} = \mathbf{h}^{'}$ consist of N samples, the optimal labels are both $\mathbf{p}$. Suppose two samples in each datasets are the same, denote as $\mathbf{h}^{(1)} = \mathbf{h}^{(2)} = \mathbf{h}^{',(1)} = \mathbf{h}^{',(2)}, \mathbf{p}^{(1)} = \mathbf{p}^{(2)}$. If the given labels for $\mathbf{h}^{'}$ is $\mathbf{p}_{false}$, where $\mathbf{p}_{false}^{(1)} \neq \mathbf{p}_{false}^{(2)}$, the labels for the other samples are the same as $\mathbf{p}$, then at each iteration of training , the upper bound of f_{sup} using $(\mathbf{h}, \mathbf{p})$ is smaller than using $(\mathbf{h}^{'}, \mathbf{p}_{false})$.*

Proof. By pyramidal structure, we have

$$\begin{aligned}
&f_{\text{sup}}\left(\mathbf{\Theta}_{m+1}\right) \leq f_{\text{sup}}\left(\mathbf{\Theta}_m\right)+\left\langle \nabla f_{\text{sup}}\left(\mathbf{\Theta}_m\right), \mathbf{\Theta}_{m+1}-\mathbf{\Theta}_m\right\rangle+\tfrac{Q_0}{2}\left\|\mathbf{\Theta}_{m+1}-\mathbf{\Theta}_m\right\|_2^2 \\
&=f_{\text{sup}}\left(\mathbf{\Theta}_k\right)-\eta\left\|\nabla f_{\text{sup}}\left(\mathbf{\Theta}_m\right)\right\|_2^2+\tfrac{Q_0}{2}\eta^2\left\|\nabla f_{\text{sup}}\left(\mathbf{\Theta}_m\right)\right\|_2^2 \\
&\leq f_{\text{sup}}\left(\mathbf{\Theta}_m\right)-\tfrac{1}{2}\eta\left\|\nabla f_{\text{sup}}\left(\mathbf{\Theta}_m\right)\right\|_2^2 \quad \text{as } \eta < 1/Q_0 \\
&\leq f_{\text{sup}}\left(\mathbf{\Theta}_m\right)-\tfrac{1}{2}\eta\left\|\text{vec}\left(\nabla_{W_2} f_{\text{sup}}\left(\mathbf{\Theta}_m\right)\right)\right\|_2^2
\end{aligned}$$

Denote $A_m = \text{vec}(\nabla f_{\text{sup}}(\mathbf{\Theta}_m))$, there is

$$\begin{aligned}
f_{\text{sup}}\left(\mathbf{\Theta}_{m+1}\right) &\leqslant f_{\text{sup}}\left(\mathbf{\Theta}_m\right)-\frac{1}{2}\eta\left(f_L^m-y\right)^{\top} A_m^{\top} A_m\left(f_L^m-y\right) \\
&=\frac{1}{2}\left(f_L^m-y\right)^{\top}\left(f_L^m-y\right)-\frac{1}{2}\eta\left(f_L^m-y\right)^{\top} A_m^{\top} A_m\left(f^m-y\right) \\
&=\frac{1}{2}\left(f_L^m-y\right)^{\top}\left(I-\eta A_m^{\top} A_m\right)\left(f_L^m-y\right)
\end{aligned}$$

Denote $S_m = \sum_{i=1}^{Nn_L}\left(1-\eta\lambda_i^m\right) v_i^m v_i^{m^{\top}}$, $f_L^m - y$ can be decomposed as $f_L^k - y = \sum_{i=1}^{Nn_L}\left(v_i^{k^{\top}}\left(f_L^k-y\right)\right) v_i^k = \frac{1}{2}\sum_{i=1}^{NnL}\left(U_i^{kT}\left(f_L^k-y\right)\right)^2\left(1-\eta\lambda_i^k\right)$ □