2021 IEEE 22nd International Workshop on Signal Processing Advances in Wireless Communications (SPAWC) | 978-1-6654-2851-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/SPAWC51858.2021.9593184

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
[†] CSE Department, Michigan State University, East Lancing, MI
[*] Shenzhen Research Institute of Big Data, The Chinese University of Hong Kong, Shenzhen, China

ABSTRACT

Machine learning has 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 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 get stuck at some low-quality local solutions. We then provide a series of theoretical results to further understand the properties of the two approaches. 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] have been designed to accomplish key tasks such as power control [4], beamforming [1], admission control [5], MIMO detection [6], 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 [4].

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 reduce the network complexity and input feature dimension [4], boost the training efficiency [4], and improve the expressiveness [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 for training:
1) *supervised learning (SL)*, in which "labels" of optimal power allocations 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 [4].

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 inter-

B. Song, M. Hong and H. Sun are partially supported by NSF grant CNS-2003033.

978-1-6654-2851-4/21/$31.00 ©2021 IEEE

ference 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 rigorous 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.

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 equipped 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 [7] in general. For problem (1) and its generalizations such as the beamforming problems in MIMO channels, many iterative optimization based algorithms have been proposed [8].

Recently, there has been a surge of works that apply DNN based approach to identify good solutions for problem (1) and its extensions [1], [4]. 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 $\bar{\mathbf{p}} := \{\bar{\mathbf{p}}^{(n)}\}_{n\in[N]}$ (where $[N] := \{1, \cdots, N\}$), which are usually obtained by solving N independent problems (1) using some optimization algorithm, such as the WMMSE [8]. 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 $\boldsymbol{\Theta}$ be the parameters of the DNN (of appropriate size), then the output of DNN can be expressed as $\mathbf{p}(\boldsymbol{\Theta}; |\mathbf{h}^{(n)}|) \in \mathbb{R}^K$. To simplify notation, we write the output of the DNN and its kth component as:

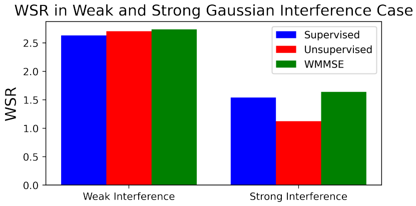


Fig. 1. Comparison between SL, UL and WMMSE in testing time, when SL, UL are trained using data where the interference channel power is equal to direct channel power (weak interference), or 10 times of the direct channel power (strong interference) when there are 10 users. In strong interference case,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

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

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:

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

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:

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

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* $R\left(\mathbf{p}(\boldsymbol{\Theta}; |\mathbf{h}^{(n)}|), |\mathbf{h}^{(n)}|\right)$ *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* $\boldsymbol{\Theta}$ *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 [9].

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

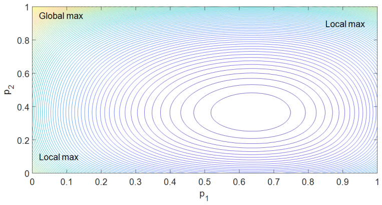


Fig. 2. For two-user IC with 2 snapshots, the true labels $\bar{\mathbf{p}}^{(1)} = (0, 1)$, $\bar{\mathbf{p}}^{(2)} = (1, 0)$. Keep the sum of label of each snapshot to be the 1: $\bar{\mathbf{p}}^{(1)} = (p_1, 1 - p_1)$, $\bar{\mathbf{p}}^{(2)} = (p_2, 1 - p_2)$. Plot the sum-rate of the two snapshots.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

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 $\boldsymbol{\Theta}$, 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 $\mathbf{p}$: $\mathbf{p} = \boldsymbol{\Theta}|\mathbf{h}|$, where $\boldsymbol{\Theta} \in \mathbb{R}^{K\times K^2}$, and $\boldsymbol{\Theta} := [\boldsymbol{\Theta}_1; \cdots ; \boldsymbol{\Theta}_K]$, with $\boldsymbol{\Theta}_k := \{\boldsymbol{\Theta}_{k,(uv)}\}_{(uv)\in W} \in \mathbb{R}^{1\times K^2}$, 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 [10], [11], 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 $\boldsymbol{\Theta}$. 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: $\mathbf{p}(\boldsymbol{\Theta}; |\mathbf{h}|) = \boldsymbol{\Theta}|\mathbf{h}|$. 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 $\boldsymbol{\Theta}_{\text{global}}$ and $\boldsymbol{\Theta}_{\text{local}}$. However, these two solutions generate different predictions:*

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

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

On the other hand, if the SL loss (3) *is used, then $f_{\text{sup}}(\boldsymbol{\Theta})$ is a convex function w.r.t. $\boldsymbol{\Theta}$, 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 $\boldsymbol{\Theta}^*(\bar{\mathbf{p}})$ denote the optimal solution for the SL problem (3) *with label $\bar{\mathbf{p}}$, and it achieves zero loss: $f_{\text{sup}}(\boldsymbol{\Theta}^*(\bar{\mathbf{p}})) = 0$.*
iii) The solution $\boldsymbol{\Theta}^*(\bar{\mathbf{p}})$ can be computed for all $\bar{\mathbf{p}}$.
Let $\mathcal{B}$ denote the set of stationary points of (4)*. Then the following holds:*

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

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 $f_{\text{sup}}(\boldsymbol{\Theta}^*) = 0$ has been verified when the neural network is "overparameterized"; see. e.g., [12]. Further, it has been shown that gradient descent (GD) can indeed find such a global optimal solution [13]. However, the work [13] cannot be applied to analyze our training problem because they require the normalized inputs, 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 [14].

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:

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

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 $f_l = \text{vec}\left(F_l\right) \in \mathbb{R}^{Nn_l}$. Similarly, denote the vectorized label as $y = \text{vec}(\mathbf{p}) \in \mathbb{R}^{NK}$. At m-th iteration of training, we use $\boldsymbol{\Theta}^m = (W_l^m)_{l=1}^{L}$ to denote all the parameters.

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 $n_2 \geq n_3 \geq \ldots \geq n_L$.*

Assumption 2. *There exist constants $\gamma \in (0, 1)$ and $\beta > 0$, such that the activation function $\sigma(\cdot)$ satisfies:*
$\sigma'(x) \in [\gamma, 1]$, $|\sigma(x)| \leq |x|$, $\forall\ x \in \mathbb{R}$, σ' is β-Lipschitz.

The first assumption defines the so-called Pyramidal Network structure [14], which consists of at least one wide layer (i.e., the number of neurons is at least the sample size). The second assumption is shown to hold true for certain activation functions [14] .

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 also be feasible. However, for the UL training, we cannot simply drop the constraints, so a sigmoid function should be added to the last layer of the output to enforce feasibility. Specifically, the modified network has the following (vectorized) output:

$$F_L = \mathrm{sig}(F_{L-1}W_L) = \frac{\mathbf{1} \times P_{\max}}{1 + e^{-F_{L-1}W_L}}. \tag{9}$$

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 [14, Assumption 3.1]. Then the following holds:*
(a) Consider optimizing the unconstrained version of (3) *using gradient descent $\mathbf{\Theta}^{m+1} = \mathbf{\Theta}^m - \eta\nabla f_{\mathrm{sup}}(\mathbf{\Theta}^m)$. There exists constant stepsize η such that the training loss converges to zero at a geometric rate, that is:*

$$f_{\mathrm{sup}}\left(\mathbf{\Theta}^m\right) \leq \left(1 - \eta\alpha_0\right)^m f_{\mathrm{sup}}\left(\mathbf{\Theta}^0\right) \tag{10}$$

where α_0 is a constant.
(b) Consider minimizing the unconstrained version of (4) *using the last layer as* (9) *and use the gradient descent algorithm (with step size η). 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 [14, Theorem 3.2]. However, Claim 3-(b) is different since we need to analyze the special network with the sigmoid activation function.

# samples Quality	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%)
# samples Quality	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 mean the relative sum rate achieved at testing time v.s. what is achieved by the given labels.

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 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 labels not only achieves higher absolute sum rate, but also higher *relative* sum rate comparing with what can be achieved by the labels.

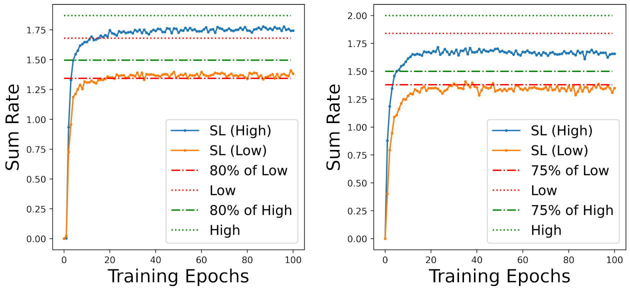
Below, we argue the benefit of high-quality label from a slightly different perspective – the label quality can influence the convergence speed of training algorithm.

Claim 4. *Suppose $(\mathbf{h}, \mathbf{p})$ and $(\mathbf{h}^{'}, \mathbf{p}^{'})$ are two datasets constructed below: Each dataset has N samples; $\mathbf{h}^{'} = \mathbf{h}$; Two samples in $\mathbf{h}$ are identical, say, $\mathbf{h}^{(1)} = \mathbf{h}^{(2)}$, and all the other samples are linearly independent; For the second dataset, the labels are constructed as follows:*

$$\mathbf{p}^{',(2)} \neq \mathbf{p}^{(2)}, \quad \mathbf{p}^{',(n)} = \mathbf{p}^{(n)}, \forall\ n, \neq 2. \tag{11}$$

Further, since $\mathbf{h}^{(1)} = \mathbf{h}^{(2)}$ and $\mathbf{h} = \mathbf{h}'$, we also have $\mathbf{h}^{',(1)} = \mathbf{h}^{',(2)}$.

Suppose that Assumption 1 and Assumption 2 hold true, and 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. Let $\mathbf{\Theta}^m$ and $\mathbf{\Theta}^{'m}$ denote the sequences of weights generated by the algorithm for the two data sets respectively. Suppose that the initial solutions of the two algorithms are the same: $\mathbf{\Theta}^{',0} = \mathbf{\Theta}^0$. Define*



(a) Strong interference with K=10 (b) Strong Interference with K=20

Fig. 3. Comparison between SL using different labels. 'Low' and 'High' in the legend means the quality of labels are low or high. We also draw the sum rate of the generated data and labels as baseline, as well as the 80% of the sum rate in 10-user case and 75% of the sum rate in 20-user case.

$$A(\mathbf{\Theta}) := \left(\mathbb{I}_{n_2} \otimes F_1^T\right) \prod_{q=3}^{L} \Sigma_{q-1} \left(W_q \otimes \mathbb{I}_N\right); \;\; A_0 := A(\mathbf{\Theta}^0).$$

Suppose all the eigenvalues of $A_0^T A_0$ are within the interval $[0,1]$. Then if we choose the stepsize η small enough, there exist $\beta > 0$ and $\beta^{'} > 0$ such that the following holds true

$$f_{\text{sup}}\left(\mathbf{\Theta}^1\right) \leq \beta f_{\text{sup}}\left(\mathbf{\Theta}^0\right), \;\; f_{\text{sup}}\left(\mathbf{\Theta}^{',1}\right) \leq \beta^{'} f_{\text{sup}}\left(\mathbf{\Theta}^{',0}\right).$$

Further, we have $\beta < \beta^{'}$, that is, the objective function with the correct label decreases faster.

In our analysis, we combined the pyramid network analysis with the decomposition technique from [15]. 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.

To empirically understand how the quality of labels affect convergence speed, we conduct the following experiments. Consider 10- and 20-user case under the strong interference setting as illustrated in Fig. 1. We generate two sets of labels for each case, the low-quality one directly obtained by WMMSE while the high-quality one first passes a given sample through a pretrained GNN model in [3] and then is fine-tuned by WMMSE. We use a fully connected network with 3 hidden layers, with the number of neurons being 200, 80, 80 for 10-user case and 600, 200, 200 for 20-user case. From Fig. 3, we see that SL with higher-quality labels achieves 80% of the baseline sum rate faster than with lower-quality labels for 10-user case. Similar result can be derived in matching 75% of the baseline for 20-user case.

IV. CONCLUSION

This work analyzes the SL and UL approaches for learning communication systems. It is shown that under certain conditions (such as having access to high-quality labels), SL can exhibit better convergence properties than UL. To our knowledge, this is the first work that rigorously analyzes the relation between these two approaches. Of course, finding high-quality labels is challenging. Is there a way to design a proper learning strategy that only requires a few high-quality labels, while still achieving the state-of-the-art performance? In our full paper [9], we developed some semi-supervised learning approach to address this question. Due to space limitation, we do not include them here.

V. 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] 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)*.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted mse 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.

[9] 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

[10] G. O. A. Gjendemsjo, D. Gesbert and S. Kiani, "Optimal power allocation and scheduling for two-cell capacity maximization," in *IEEE WiOpt*, 2006, pp. 1–5.

[11] 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.

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

[13] 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*, 2019, pp. 1675–1685.

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

[15] 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*, 2019, pp. 322–332.