Distributed Uplink Beamforming in Cell-Free Networks Using Deep Reinforcement Learning

Firas Fredj, Yasser Al-Eryani, Setareh Maghsudi, Mohamed Akrout, and Ekram Hossain, Fellow, IEEE

Abstract

The emergence of new wireless technologies together with the requirement of massive connectivity results in several technical issues such as excessive interference, high computational demand for signal processing, and lengthy processing delays. In this work, we propose several beamforming techniques for an uplink cell-free network with centralized, semi-distributed, and fully distributed processing, all based on deep reinforcement learning (DRL). First, we propose a fully centralized beamforming method that uses the deep deterministic policy gradient algorithm (DDPG) with continuous space. We then enhance this method by enabling distributed experience at access points (AP). Indeed, we develop a beamforming scheme that uses the distributed distributional deterministic policy gradients algorithm (D4PG) with the APs representing the distributed agents. Finally, to decrease the computational complexity, we propose a fully distributed beamforming scheme that divides the beamforming computations among APs. The results show that the D4PG scheme with distributed experience achieves the best performance irrespective of the network size. Furthermore, the proposed distributed beamforming technique performs better than the DDPG algorithm with centralized learning only for small-scale networks. The performance superiority of the DDPG model becomes more evident as the number of APs and/or users increases. Moreover, during the operation stage, all DRL models demonstrate a significantly shorter processing time than that of the conventional gradient descent (GD) solution.

Index Terms

Cell-free network, distributed beamforming, deep reinforcement learning, deep deterministic policy gradient algorithm (DDPG), distributed distributional deterministic policy gradients algorithm (D4PG).

F. Fredj, Y. Al-Eryani, M. Akrout and E. Hossain are with the Department of Electrical and Computer Engineering at the University of Manitoba, Canada (emails: firas.fredj4@gmail.com, aleryany@myumanitoba.ca, akroutm@myumanitoba.ca, Ekram.Hossain@umanitoba.ca). S. Maghsudi is with Technical University of Berlin, Germany (email: maghsudi@tu-berlin.de). This work was supported in part by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) and in part by Grant 16KIS1165 from the German Federal Ministry of Education and Research (BMBF).
I. INTRODUCTION

To provide ultra-reliable low-latency communications (URLLC) [1] in the beyond 5G wireless systems, the idea of cell-free networks has emerged [2]. A cell-free network will use a fully connected wireless network architecture with centralized processing, control, and storage of data. Such centralized network operations mitigate the adverse effects of non-coordinated collisions and interference among transmitted signals, especially in scenarios such as massive machine-type communications (mMTC) [3]. In a cell-free wireless network (Fig. 1), all access points (APs)/base stations (BSs) cooperate to simultaneously serve all users within the network coverage area [4]–[6]. In a cell-free architecture, fast fronthaul/backhaul links connect all APs to an edge cloud processor which is responsible for simultaneous downlink(uplink) beamforming design for transmit(receive) signals to(from) different users [7], [8]. On the downside, a fully centralized cell-free network architecture requires a huge computational capacity. Moreover, without an efficient design, there would be excessive control signaling [9], [10]. Note that the main benefits of fully centralized cell-free networks include enhanced coverage, improved diversity, and provisioning of efficient interference cancellation mechanisms.

![Fig. 1: A cell-free network model.](image)

The challenges associated with cell-free networking such as multiuser beamforming and channel estimation [11]–[14] can be addressed by using artificial intelligence (AI), specifically, machine learning (ML) techniques. These problems are often characterized by an algorithmic deficit rather than a modeling deficit [15]. Among numerous methods, deep reinforcement learning (DRL) is a notable candidate to design cell-free networks that avoids having a training data set a priori which are extremely hard
to obtain in dynamic wireless environments. Also, DRL enables us to achieve a trade-off between centralized- and distributed processing of computational tasks. Recent results have shown that simultaneous uplink/downlink beamforming within in a centralized unit results in optimal system performance. On the downside, fully centralized processing suffers from high computational complexity and excessive processing delay, especially when several users utilize the same time-frequency resources. However, as the signal processing of the uplink/downlink cell-free network becomes more distributed, the network performance becomes closer to that of a cellular network with non-cooperative APs.

The state-of-the-art literature on cell-free networks focus on uplink/downlink beamforming [4], [8], estimation of channel stat information (CSI) [11], [16], fronthaul imperfections [17], and scalable cell-free network designs [9], [18], [19]. For example, in [20], the authors propose conjugate beamforming and zero-forcing precoding scheme for a fully centralized downlink cell-free network. They show that the zero-forcing technique outperforms the conjugate beamforming. In [11], the authors develop a channel estimation technique for mmWave-enabled massive cell-free networks using the supervised learning-based denoising convolutional neural network. To reduce the complexity of centralized data processing, [18] proposes a partitioned cell-free wireless network architecture. The architecture clusters the cooperating APs based on current network CSI (user-centric clustering). The scheme enables an efficient design of practical mMTC systems by compensating the effect of inter-cluster interference. The compensation follows by network partitioning and enabling multi-level successive interference cancellation (SIC) at each receiver [6]. Another low-complexity design of cell-free network architecture appears in [19]. The core idea is to reduce the dimensionality of beamforming matrices by a dynamic clustering of APs. Each cluster then represents a single multi-antenna AP (transmit/receive diversity). In [21], the authors utilize supervised learning to solve the beamforming problem in cell-free networks. They locate a complete neural network optimizer in each AP. Every AP then obtains the local CSI knowledge by estimating only the large-scale fading while considering the small-scale fading as a constant. Table I summarizes some of the important works in the area of beamforming in cell-free networks.

As is evident, DRL, and more specifically distributed DRL, is yet to be exploited fully to solve the signal processing problems in cell-free networks. In this paper, we investigate several practical beamforming designs for cell-free networks considering both centralized and distributed settings. Most importantly, DRL-based optimization methods do not necessitate a repetitive solving of the beamforming problem at every coherence time interval, as they learn the optimal beamforming matrix given the state of the wireless communication environment. The main contributions of this paper can be summarized as follows:
TABLE I: Summary of beamforming schemes in cell-free networks

| Ref. | System Model | Main Objective | Techniques and Characteristics |
|------|--------------|----------------|-------------------------------|
| [22] | Centralized  | Max-Min Fairness | The paper formulates downlink beamforming problem as a quasi-concave optimization and uses bisection method. |
| [23] | Centralized  | Max-Min Fairness | The paper designs an angle-of-arrival-based beamforming/combining scheme for FDD-based cell-free network. |
| [4], [8] | Centralized | Max-Min Fairness | The paper uses conjugate beamforming and zero forcing techniques. |
| [24] | Centralized | Maximize Sum Rate, Max-Min Fairness | The paper trains a deep neural network to learn the mapping between a set of input data and the optimal solution of the power allocation strategy. |
| [25] | Co-located, Cell-free | Max-Min Fairness | The paper uses Lyapunov optimization techniques to develop a dynamic scheduling algorithm to perform user scheduling based on time slot and transmission rate. |
| [18] | User-centric clustering | Per-Cluster Maximize Sum Rate | The paper formulates downlink beamforming problem with optimal CSI as concave optimization and uses Lagrangian multiplier method. |
| [19] | Dynamic | Maximize Sum Rate | The paper uses hybrid DRL-based model for AP clustering and beamforming that utilizes the DDQN and DDPG algorithms. |

- For a fully centralized uplink cell-free network, we formulate the beamforming optimization problem that maximizes the per-user normalized sum rate. We then solve this problem by developing a solution based on the deep deterministic policy gradient (DDPG) algorithm [26].
- We also propose a novel distributed experience-oriented beamforming system based on the distributed distributional deterministic policy gradient (D4PG) algorithm [27] with the APs representing the distributed agents.
- To reduce the complexity of centralized learning, we propose a novel DRL-based beamforming scheme with distributed learning.
- We evaluate the proposed beamforming designs numerically for different system settings considering non-orthogonal pilots contamination and shadow fading.

The rest of this paper is organized as follows. Section [II] briefly reviews the preliminaries of DRL and DRL-based beamforming. The system model and the beamforming problem are presented in Section [III]. We design a centralized DRL solution as well as a distributed DRL solution for the beamforming problem in Section [IV]. Section [V] presents another beamforming solution based on distributed learning and also discusses the complexity of all of the DRL-based solutions. In Section [VI], we present and discuss the numerical results. Section [VII] concludes the paper.

**Notations:** For a random variable (rv) $X$, $F_X(x)$ and $f_X(x)$, respectively, represent cumulative distribution function (CDF) and probability density function (PDF). Moreover, $\mathbb{E}[]$ denotes the expectation. For a given matrix $A \in \mathbb{C}^{M \times N}$, $A^H$ represents the Hermitian transpose of $A$. The PDF of a random variable $X$ following the Nakagami-$M$ distribution is given by $f_X(x) = \frac{2^{M-1} e^{M} x^{M-1}}{\Gamma(M) \Omega^M} x^{2M-1} e^{-\frac{M}{\Omega} x^2}$. A random variable $X$ that follows the Gamma distribution is denoted by $X \sim \mathcal{G}(\alpha, \beta)$, with the PDF being $f_X(x) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad x > 0$, where $\beta > 0$, $\alpha \geq 1$, and $\Gamma(z)$ is the Euler’s Gamma function. Moreover, the
distributions $O\mathcal{G}(\alpha_i, \beta_i), i = 1, \ldots, N$, refer to the $i$-th ascending-ordered rv from a set of $N$ gamma rvs with parameters $\alpha_i$ and $\beta_i$.

II. BACKGROUND ON DRL AND BEAMFORMING OPTIMIZATION

A. DRL Preliminaries

In reinforcement learning (RL), an agent interacts with an environment in discrete time. At each step, the agent takes an action and receives some reward while observing a new state or some other information. The agent utilizes the feedback to learn a behavior (policy) that maximizes the cumulative reward. DRL is indeed a solution for Markov decision processes (MDPs). An MDP is characterized by a tuple $(S, A, \mathcal{P}, \mathcal{R}, \zeta)$, where $S$ and $A$ represent the state space and the agent’s action space, respectively. Moreover, $\mathcal{P}$ is the transition probability matrix, where $\mathcal{P}(s, a, s') \in [0, 1]$ is the probability that state $s$ changes to state $s'$ by selecting action $a$. $\mathcal{R} : S \times A \rightarrow \mathbb{R}$ defines the expected reward of performing action $a$ at state $s$. Finally, $\zeta$ is the reward’s discount factor. The goal is to select the best action at each step so as to maximize the accumulated discounted reward.

Deep RL (DRL) combines neural networks with RL algorithms. Neural networks approximate functions, which are especially useful when the state space and the action space are too large to be completely known and belong to continuous space. A neural network learns to map the states to values or state-action to Q-values using the historical outcomes. DRL algorithms fall into three categories: (i) value-based methods that aim to learn a value function like deep Q-learning (DQL) algorithm [28], (ii) policy-based methods that learn the optimal policy function, and (iii) actor-critic methods that combine value-based and policy-based methods.

In this paper, we use two actor-critic algorithms, namely, DDPG [26] and D4PG [27]. These methods are suitable to optimize beamforming as the action space can be continuous in both DDPG and D4PG. Moreover, the exploration can be distributed for D4PG. The DDPG algorithm uses state-action Q-value critic based on deep Q-learning [28] and updates the policy using its critic gradients. D4PG builds on the DDPG approach by making several enhancements as the Q-value estimation and the distributed collection of experiences. Note that an experience is the process of exploring a new action by executing it in the environment, thereby receiving some reward and observing the new state. Later in this paper, we will provide detailed descriptions of the DDPG and D4PG algorithms.
B. DRL Agent for Beamforming Optimization

For a cell-free network with $M$ APs and $K$ users (Fig. 1), we intend to develop a DRL model to optimize the beamforming matrix $W$. This matrix includes beamforming vectors of all users within the network coverage area given the complete CSI. Note that, since the beamforming vector of the $k$-th user is given by $w_k = [w_{1k} \ldots w_{Mk}]$, the matrix $W$ has a dimension of $K \times M$.

We cast the beamforming optimization problem as a Markov Process Decision Process (MDP). We then solve the problem by applying a DRL method that involves an agent interacting with a cell-free network as the environment. Such a DRL model can be implemented either centrally or in a distributed manner. The design of a cell-free network environment includes the definition of state $s$, the action $a$, and the immediate reward function $r$, needed for the DRL algorithm to estimate the policy and the Q-values. The state of the wireless cell-free network environment can be any key performance indicator. While the action of this model is the optimization variable of the beamforming matrix $W$, the reward may be any performance metric that jointly quantifies the performance of all active users. In Table II, we summarize the design parameters of a DRL model and the corresponding measures in wireless cell-free network.

| Environment Variables | System Equivalence |
|-----------------------|--------------------|
| State $s = \{s_1, \ldots, s_k\}$ | User SINR: $\{\gamma_1, \ldots, \gamma_k\}$ |
| Reward $r$ | Sume-rate for all users: $\sum_{k=1}^{K} \log(1 + \gamma_k)$ |
| Action $a$ | Beamforming matrix: $W$ |

III. System Model, Assumptions, and Problem Formulation

A. Cell-free Network Model

We consider the uplink of a wireless network with $M$ single-antenna APs and $K$ single-antenna users that have fixed locations within a certain coverage area, as shown in Fig. 1. Each AP has a baseband processor to partially process the signal received from all connected users. We refer to such APs as ‘enhanced-AP’ (eAP) to distinguish them with the conventional APs. The eAPs are connected via the backhaul links, hence forming a cell-free network [2]. Such a network architecture enables the distributed eAPs to collaboratively serve all the users within the network’s coverage area.

The beamforming optimization follows either centrally at a edge cloud processor (ECP) or in a distributed manner by several eAPs. All eAPs are connected to the ECP that distributes and organizes the data processing tasks among different eAPs.
To obtain the CSI, a random set of orthogonal pilot sequences is assigned to each user. Between the \( k \)-th user and the \( m \)-th eAP, the channel gain is a random variable

\[
g_{mk} = \mathcal{F}_{mk}^{1/2} h_{mk},
\]

(1)

In (1), \( h_{mk} \) is the small-scale channel fading that follows a Nakagami-\( M \) distribution with spreading and shape parameters \( M_{mk} \) and \( \Omega_{mk} \), respectively. Therefore, \( |h_{mk}|^2 \sim \mathcal{G}(\alpha_{mk}, \beta_{mk}) \), where \( \alpha_{mk} = M_{mk} \) is the shape parameter and \( \beta_{mk} = \frac{M_{mk}}{\Omega_{mk}} \) is the inverse-scale parameter. Moreover, we have

\[
\mathcal{F}_{mk} = L_{mk}^{-2\kappa} \frac{\alpha_{sh}}{10},
\]

(2)

where \( L_{mk} = ||d_{mk}|| \) is the Euclidean distance between the \( k \)-th user and the \( m \)-th eAP. Also, \( \kappa \) is the path-loss exponent with a value depends on the propagation environment with \( \kappa \geq 2 \). \( \alpha_{sh} \) is the shadow fading variance in dB. Furthermore, \( z = \sqrt{\delta}a_{mk} + \sqrt{1-\delta}b_{mk} \), where \( 0 \leq \delta \leq 1 \) is the transmitter-receiver shadow fading correlation coefficient [29]. The parameters \( a_{mk} \sim \mathcal{N}(0, 1) \) and \( b_{mk} \sim \mathcal{N}(0, 1) \) characterize the shadow fading. We assume that \( L_{mk}^{-2\kappa} \) and \( \mathcal{F}_{mk} \) \( \forall m \) and \( k \) are known. This assumption is justified as large-scale fading parameters can be easily estimated given the received signal strength (RSS). Accordingly, we have \( |g_{mk}|^2 \sim \mathcal{G}(\alpha_{mk}, \beta_{mk}/\mathcal{F}_{mk}) \).

B. Uplink Network Training

To estimate the CSI of the cell-free network, we train the network using a set of orthonormal pilot sequences. Let \( \varphi_k = [\varphi_{k,1} \ldots \varphi_{k,\tau_p}]^H \), \( ||\varphi_k||^2 = 1 \), be the pilot sequence with sample size \( \tau_p \) that is assigned to the \( k \)-th user. Note that \( \tau_p \leq \tau_c \), where \( \tau_c \) is the coherence time of the channel via which the sequence is sent to all the eAPs with constant power. The received pilot vector at the \( m \)-th eAP yields

\[
y_{p,m} = \sum_{k=1}^{K} \sqrt{\tau_p \rho_k} g_{mk} \varphi_k + \eta_m,
\]

(3)

where \( \rho_k \) is the normalized transmission power for each symbol of the \( k \)-th user pilot vector. Moreover, \( \eta_m \in \mathbb{C}^{\tau_p \times 1} \) is the zero-mean complex additive white Gaussian noise (AWGN) vector related to pilot symbols with independent and identically distributed (i.i.d) rvs, i.e., \( \eta_m \sim \mathcal{C}\mathcal{N}(0, 1/2) \). To find the best estimate of \( g_{mk} \) (denoted by \( \hat{g}_{mk} = \mathcal{F}_{mk}^{1/2} \hat{h}_{mk} \)) given the vector of observations \( y_{p,m} \), we first project \( y_{p,m} \) over \( \varphi_k^H \). Therefore,

\[
\hat{y}_{p,m} = \varphi_k^H y_{p,m} = \sqrt{\tau_p \rho_k} g_{mk} + \sum_{l=1,l\neq k}^{K} \sqrt{\tau_p \rho_l} g_{ml} \varphi_k^H \varphi_l + \varphi_k^H \eta_m.
\]

(4)
Here \( g_{mk} \) can be estimated from (4) by using the maximum a posteriori decision rule (MAP), which is identical to the minimum mean square method (MMSE) \([30], [31]\). Furthermore, given that the pilot signals are partially orthogonal and partially non-orthogonal, \( \varphi_k^H y_{p,m} \) in (4) represents a sufficient statistics for the optimal estimation of \( g_{mk} \) (MMSE). Thus the best estimate of \( g_{mk} \) is given by [3]

\[
\hat{g}_{mk} = \frac{\mathbb{E} [\dot{y}^*_{mk} g_{mk}]}{\mathbb{E} [||\dot{y}_{mk}||^2]} \dot{y}_{mk} = \mathcal{E}_{mk} \dot{y}_{mk}.
\]

(5)

Under the assumption that for all \( m \) and \( k \), \( g_{mk}s \) are proper independent but non-identically distributed (i.n.d) complex Gaussian rvs, and that \( \eta_{m}s \) are zero-mean i.i.d rvs, the MMSE estimation constant \( \mathcal{E}_{mk} \) yields

\[
\mathcal{E}_{mk} = \frac{\sqrt{\tau_p \rho_k} \mathcal{F}^{1/2}_{mk} \left( \frac{\alpha_{mk}}{\beta_{mk}} \right)}{\tau_p \sum_{l=1}^{K} \rho_l \mathcal{F}^{1/2}_{ml} \left( \frac{\alpha_{ml}}{\beta_{ml}} \right) |\varphi_{mk}^H \varphi_{ml}|^2 + 1},
\]

(6)

where we use the fact that \( \mathbb{E} [||h_{mk}||^2] = \frac{\alpha_{mk}}{\beta_{mk}} \). If all UEs receive a set of mutually orthogonal pilot sequences (i.e. \( |\varphi_k^H \varphi_l| = 0, \forall k \neq l \)), the estimated small-scale channel fading in (4) reduces to a scaled version of the exact fading gain plus a relatively small AWGN noise portion. However, depending on the applications and due to the limitations concerning the length of the training sequence, non-orthogonal pilot signals have to be used among some active UEs. To decrease the computational complexity at the ECP, one solution is to estimate the CSI at distributed eAPs, where the \( m \)-th eAP estimates channel gains \( g_{mk}, \forall k = 1, \ldots, K \).

C. Uplink Data Transmission

In a cell-free network, each eAP receives the composite of signals from all users. For each user, a weighted sum of composite signals from all eAPs is constructed to maximize the signal component while minimizing the remaining interference plus noise. This process takes place at the baseband level in the ECP, before forwarding the detected signal of each user to its final destination. Formally, the overall
signal received by the ECP to be used in detecting the $k$-th user component is given by

$$y_k = \sum_{m=1}^{M} w_{mk} \left[ \sum_{l=1}^{K} \hat{g}_{ml} \sqrt{p_k} x_l + \tilde{\eta}_m \right]$$

$$= \sqrt{\sum_{m=1}^{M} w_{mk}^2} \left[ \sum_{l=1}^{K} \hat{g}_{ml} \sqrt{p_k} x_l + \sum_{m=1}^{M} w_{mk} \sqrt{\sum_{l=1,l\neq k}^{K} \hat{g}_{ml} \sqrt{p_k} x_l} \right]$$

Desired Signal

$$+ \sum_{m=1}^{M} w_{mk} \left[ \sqrt{p_k} x_k \sqrt{\sum_{l=1,l\neq k}^{K} \hat{g}_{ml} \sqrt{p_k} x_l \varphi_l^H \varphi_0 | g_{mv} \right] + \sum_{q=1,q\neq k}^{K} \sum_{u=1,u\neq q}^{K} \sqrt{p_q} \rho_q \sum_{v=1,v\neq k}^{M} \sum_{z=1,z\neq k}^{K} \sqrt{p_z} \rho_z \sum_{u=1,u\neq q}^{K} \left[ \varphi_z^H \varphi_u | g_{mu} \right] \right]$$

Inter-User Interference

$$+ \sum_{v=1,v\neq k}^{M} w_{mk} \left[ \sqrt{p_k} x_k \sqrt{\sum_{l=1,l\neq k}^{K} \hat{g}_{ml} \sqrt{p_k} x_l \varphi_l^H \varphi_0 | g_{mv} \right] + \sum_{z=1,z\neq k}^{K} \sum_{u=1,u\neq q}^{K} \sqrt{p_z} \rho_z \sum_{z=1,z\neq k}^{K} \left[ \varphi_z^H \varphi_u | g_{mu} \right] \right]$$

AWGN-related estimation error

$$\sum_{m=1}^{M} w_{mk} \left[ \sqrt{p_k} x_k \sqrt{\sum_{l=1,l\neq k}^{K} \hat{g}_{ml} \sqrt{p_k} x_l \varphi_l^H \varphi_0 | g_{mv} \right] + \sum_{z=1,z\neq k}^{K} \sum_{u=1,u\neq q}^{K} \sqrt{p_z} \rho_z \sum_{z=1,z\neq k}^{K} \left[ \varphi_z^H \varphi_u | g_{mu} \right] \right]$$

non-orthogonal pilot-related estimation error

where $w_{mk}$ is the $m$-th element of the beamforming vector related to the $k$-th user such that $0 \leq w_{mk} \leq 1$. Moreover, $p_k$ is the uplink transmission power of the $k$-th user such that $0 \leq p_k \leq P_k$, where $P_k$ is the power budget of the $k$-th user. Also, $x_k$ is the transmitted symbol of the $k$-th user such that $\mathbb{E}[|x_k|^2] = 1$, and $\tilde{\eta}_m$ is the AWGN at the $m$-th eAP with $\tilde{\eta}_m \sim \mathcal{CN}(0, 1/2)$. The instantaneous signal-to-interference-plus-noise ratio (SINR) experienced by the $k$-th user is given by [32]

$$\gamma_k = \frac{\sum_{m=1}^{M} w_{mk}^2 \sum_{l=1,l\neq k}^{K} |\hat{g}_{ml}|^2 + \sum_{v=1,v\neq k}^{M} |\hat{g}_{mv}|^2 + \sum_{q=1,q\neq k}^{K} \sum_{u=1,u\neq q}^{K} |\hat{g}_{mu}|^2 + 1}{\sum_{m=1}^{M} w_{mk}^2 \sum_{l=1,l\neq k}^{K} |\hat{g}_{ml}|^2 + \sum_{v=1,v\neq k}^{M} |\hat{g}_{mv}|^2 + \sum_{q=1,q\neq k}^{K} \sum_{u=1,u\neq q}^{K} |\hat{g}_{mu}|^2}$$

Equation (8) is concluded from the following: When both the transmitter and the receiver know the estimated CSI, one can replace the second moments of channel fading parameters with their instantaneous values. For example, the numerator of (8) can be written as $\mathbb{E}[|G_k^H W_k x_k|^2] = W_k^H R_k W_k$, where $G_k = [C_{1k} g_{1k} \ldots C_{Mk} g_{Mk}]$, $W_k = [w_1 \ldots w_M]$, and $C_{mk} = \sqrt{p_k} \rho_k \sqrt{p_k} \rho_k \hat{g}_{ml} \sqrt{p_k} x_l$. Moreover, $R_k$ represents the auto-correlation matrix of $k$-th user signal and is defined as $R_k = \mathbb{E}[|G_k^H G_k|^2] = \bar{G}_k \bar{G}_k^H + C_{\bar{G}_k}$, where $\bar{G}_k$ and $C_{\bar{G}_k}$ are the mean and covariance matrices of $G_k$, respectively. If both the transmitter and the receiver know the instantaneous CSI, $R_k$ yields $R_k = \bar{G}_k \bar{G}_k^H$ [32]. By a similar procedure one can
characterize the interference power component of (8). Additionally, one can compute the power of AWGN component by utilizing the fact that all noise samples are i.i.d circularly symmetric Gaussian rvs with zero-mean and constant variance $\sigma_m^2 = \sigma^2 = 1/2$, $\forall \ m$.

D. Problem Formulation

Since all users transmit in the same time-frequency channel, the receiver deploys successive interference cancellation (SIC) to increase the users’ SINR values, thereby enhancing the per-user performance. To this end, the beamforming vector of each specific user is designed optimally to separate the signal component of that user from other components at least by some value referred to as receiver sensitivity. More precisely, the receiver of every user $k$ first decodes the signal components of other users with a higher power. It subtracts these components from the overall signal. It then decodes the desired signal treating the remaining users’ components, i.e. those with lower power, as interference. Formally, when detecting the signal from the $k$-th user, we first arrange the received signal components from the UEs in an ascending order such that $\sum_{m=1}^{M} |g_{m1}|^2 \leq \ldots \leq \sum_{m=1}^{M} |g_{mk}|^2 \leq \ldots \leq \sum_{m=1}^{M} |g_{mK}|^2$ [18]. The beamforming vector of the $k$-th user (denoted by $w_k = [w_{1k} \ldots w_{Mk}]$) shall maximize an objective which is a function of $\gamma_k$, $\forall k = 1, \ldots, K$. For this, $\gamma_k$ in (8) is modified as

$$\gamma_k = \frac{\sum_{m=1}^{M} w_{mk}|\tilde{g}_{mk}|^2}{\sum_{m=1}^{M} w_{mk}|\tilde{g}_{mk}|^2 + \sum_{v=1}^{M} |g_{m}\bar{\phi}|^2 + \sum_{q=1}^{K} |g_{q}\bar{\phi}|^2 + \sum_{u=1,u\neq k}^{M} |\tilde{g}_{mu}|^2} + 1, \quad (9)$$

in which $|\tilde{g}_{mi}|^2 \sim O\left(\tilde{\alpha}_{mi}, \tilde{\beta}_{mi}\right)$ and $\tilde{\alpha}_{mi} = M_{mi}$ for $i = k, l, v, u$. Moreover, $\tilde{\beta}_{mi} = \frac{M_{mi}\tilde{\sigma}_{mk}}{\Omega_{mi}F_{mi}q_{pv}p_{vk}\bar{\phi}^2 |\Phi_v^t|}$ for $i = k, l$. In addition, $\tilde{\phi}_{tv} = \frac{M_{mk}\tilde{\sigma}_{mk}}{\Omega_{v}\sum_{i}^{M} F_{mv}q_{pv}p_{vk}\bar{\phi}^2 |\Phi_v^t| |\bar{\phi}_v^t|} + 1$. Also, we have

$$\tilde{\sigma}_{mk} = \sum_{m=1}^{M} w_{mk} \left[ \sum_{t=1}^{T_v} \left( p_{k}\bar{\phi}^2_{mk} |\bar{\phi}_k^t| + \sum_{l=1,l\neq k}^{K} p_{l}\bar{\phi}^2_{ml} |\bar{\phi}_l^t| \right) + 1 \right]. \quad (10)$$

The beamforming problem can be then formulated as

$$P_1 : \text{maximize} \quad \sum_{k=1}^{K} \log (1 + \gamma_k)$$

$$\text{S.t.} \quad C_1 : \sum_{m=1}^{M} \left( w_{m\delta_{l}}^2 - \sum_{i=\delta_l+1}^{l} w_{mi}^2 \right) \tilde{\gamma}_{ml} \geq P_s, \quad (11)$$

$$C_2 : ||w_k||^2 \leq 1,$$

$$\forall k, \forall \delta_{l} = 1, \ldots, l - 1 \text{, and } l = 2, \ldots, K,$$

where $\tilde{\gamma}_{ml} = p_l|\tilde{g}_{ml}|^2$, and $W \in [0 1]^{MK \times K}$ is the overall beamforming matrix in which $w_k = [w_{1k} \ldots w_{Mk}]$. Note that in (11), the objective function is a function of $\gamma_k$, which is a function of $\omega_k$. The constraint $C_1$
corresponds to the \( \sum_{l=2}^{K} (l-1) = \frac{K(K-1)}{2} \) conditions of successful SIC operation with a receiver sensitivity of \( P_s \). For maximizing the minimum transmission rate (max-min fairness), the objective function of the optimization problem shall be minimum \( \log_2 (1 + \gamma_k) \). The receiver deals only with measured (estimated) channel values that include the estimation error and the AWGN component. However, the SINR value after the SIC procedure decreases due to the pilot contamination components. With optimal CSI estimation, (11) is a concave optimization problem with affine constraints [18]. This holds as the objective function in (11) can be written as the difference of two monotonic concave and convex functions with the concave one being always greater than the other. However, when introducing the CSI estimation error, this argument does not hold due to the additional power of the CSI estimation error. This is indeed a varying component that depends on pilot assignment among users and the number of orthonormal pilot sequences. Consequently, (11) is not solvable by conventional optimization methods. As a result, we use a set of iterative optimization methods that provably converges to a local minimum after a finite number of iterations. In the next three sections, we develop three solutions for the formulated problem using the theory of DRL-based optimization.

IV. DRL-BASED BEAMFORMING ALGORITHMS: DDPG AND D4PG

A. The DDPG Algorithm

The work in [19] establishes that, in cell-free networks, the total network transmission rate (social welfare) can be maximized by centralized processing at the ECP. In this section, we propose a DRL-based centralized solution for the beamforming problem in (11) using the DDPG learning algorithm. This solution serves as a benchmark for the proposed distributed beamforming techniques in subsequent sections.

The actor-critic algorithm in DDPG can handle continuous state space and continuous action space. Since the elements \( w_{ij} \) of \( W \) are continuous in the range \([0, 1]\), to find the optimal beamforming matrix \( W \in [0 1]^{M \times K} \), we use DDPG. It uses two neural networks as function estimators: (i) The critic, \( Q(s, a) \), whose parameters are \( \theta^Q \) and calculates the expected return given state \( s \) and action \( a \); (ii) The actor, \( \mu(s) \), whose parameter is \( \theta^\mu \) and determines the policy. In DDPG, the actor directly maps states to actions instead of outputting a probability distribution across a discrete action space. The starting point for learning an estimator to \( Q^*(s, a) \) is the Bellman equation given by

\[
Q^*(s, a) = \mathbb{E}_{(s, a, r, s') \in \mathbb{R}} \left[ r(s, a) + \zeta \max_{a'} Q^*(s', a') \right],
\]

(12)
where $R$ is the set of the experiences and $\zeta$ is the discount factor. Computing the maximum over actions in the target is quite challenging in continuous action spaces, since $r + \zeta \max_a Q^*(s', a')$ depends on optimization variables. DDPG handles this coupling by using two target networks, critic target network $Q'(s, a)$ and a policy target network $\mu'(s)$ to make the algorithm stable. These two networks use a set of parameters $\theta^{Q'}$ and $\theta^{\mu'}$ updated by Poylak averaging with factor $\tau$ as follows:

$$\theta^{Q'} \leftarrow \rho \theta^{Q} + (1 - \rho) \theta^{Q'} \quad (13)$$

$$\theta^{\mu'} \leftarrow \rho \theta^{\mu} + (1 - \rho) \theta^{\mu'} \quad (14)$$

Therefore, the critic network minimizes the following loss function:

$$Loss = \frac{1}{N} \sum_i \left( Q(s_i, a_i) - r_i + \zeta Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \right)^2. \quad (15)$$

In policy learning, DDPG learns $\theta^{\mu}$ that maximizes $Q$. At every round, it maximizes the expected return as

$$J(\theta^{\mu}) = \mathbb{E}\left[ Q(s, a) \Big| s = s_t, a = a_t \right], \quad (16)$$

and updates the weights $\theta^{\mu}$ by following the gradient of (16)

$$\nabla J_{\theta^{\mu}}(\theta) \approx \nabla_a Q(s, a) \cdot \nabla \mu(s|\theta^{\mu}). \quad (17)$$

This update rule represents the Deterministic Policy Gradient theorem $^{33}$. The term $\nabla_a Q(s, a)$ is obtained from the backpropagation of the Q-network $Q(s, a|\theta^{Q})$ w.r.t the action input $\mu(s|\theta^{\mu})$. Algorithm $\square$ summarizes the DDPG learning process.

**B. D4PG-Based Beamforming with Distributed Agents’ Experience**

In the previous section, we discussed the fully centralized DDPG-based beamforming scheme as a replacement of conventional centralized optimization at the ECP in the absence of error-free CSI. In this section, we take a step toward distributed beamforming in cell-free networks. We propose a DRL-based beamforming scheme which exploits the experiences of the distributed agents. This scheme is based on the distributed distributional deep deterministic policy gradient (D4PG) algorithm $^{34}$. The core idea is to distribute the exploration process among the eAPs and collect the experiences at the ECP. Also, only the learning process is retained in the ECP. This approach improves the quality of training data in the actor and critic networks since the eAPs simultaneously generate multiple experiences with different exploration processes.
Algorithm 1 DDPG learning process [26]

1: Randomly initialize the actor network $\mu(s|\theta^\mu)$ and the critic network $Q(s, a|\theta^Q)$ with weights $\theta^\mu$, $\theta^Q$.
2: Initialize the target networks $Q'$ and $\mu'$ with weights $\theta'^Q \leftarrow \theta^Q$ and $\theta'^\mu \leftarrow \theta^\mu$.
3: Initialize the replay buffer $\mathcal{R}$.
4: for episode = 1 to max-number-episodes do
5: Initialize a random process $\mathcal{N}$ for action exploration.
6: Observe the initial state $s_1$.
7: for $t = 1$ to max-episode-steps do
8: Perform action $a_t = \mu(s_t) + \mathcal{N}_t$. Observe reward $r_t$ and the next state $s_{t+1}$.
9: Store the transition $(s_t, a_t, r_t, s_{t+1})$ in $\mathcal{R}$.
10: Let $N$ be the batch size. Sample random mini batch of $N$ transitions from $\mathcal{R}$.
11: Update the critic by minimizing the loss

$$L = \frac{1}{N} \sum_j (y_j - Q(s_j, a_j|\theta^Q))^2,$$

$$y_j = r_j + \zeta Q'(s_{j+1}, \mu'(s_{j+1})|\theta'^\mu|\theta'^Q).$$

12: Update the actor policy using sampled policy gradient ascent as

$$\nabla_{\theta^\mu} \approx \frac{1}{N} \sum_j \nabla_a Q(s, a|\theta^Q)|_{s=s_j, a=\mu(s_j)} \nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s=s_j}.$$ 

13: Update the target networks as

$$\theta'^Q \leftarrow \tau \theta^Q + (1 - \tau) \theta'^Q,$$

$$\theta'^\mu \leftarrow \tau \theta^\mu + (1 - \tau) \theta'^\mu.$$ 

14: end for
15: end for

Fig. 2 illustrates the implementation of D4PG in a cell-free network with $M$ eAPs and one ECP. D4PG is an actor-critic method that enhances the DDPG algorithm to perform in a distributed manner, thereby improving the estimations of the Q-values. Below we discuss these enhancements to the D4PG algorithm, which we explicitly describe in Appendix A.

1) Distributional Critic: In D4PG, the Q-value is a random variable following some distribution $Z_w$ with parameters $w$, thus $Q_w(s, a) = \mathbb{E}[Z_w(s, a)]$. The objective function for learning the distribution
minimizes some measure of the distance between the distribution estimated of the target critic network and that of the critic network, e.g., the binary cross-entropy loss. Formally,

$$L(w) = \mathbb{E}[d(T_\mu, Z_w(s, a), Z_{w'}(s, a))],$$

where $T_\mu$ is the Bellman operator. The deterministic policy gradient update yields

$$\nabla_\theta J(\theta) \approx \mathbb{E}_\rho[\nabla_\theta \mu_\theta(s) \nabla_a Q_w(s, a)|a=\mu_\theta(s)] = \mathbb{E}_\rho[\nabla_\theta \mu_\theta(s) \mathbb{E}[\nabla_a Z_w(s, a)|a=\mu_\theta(s)]].$$  

(24)

2) $N$-steps returns: An agent in D4PG computes the $N$-step Temporal Difference (TD) target instead.

Formally,

$$(T^N_{\mu_\theta} Q)(s_0, a_0) = r(s_0, a_0) + \mathbb{E}\left[\sum_{n=1}^{N-1} r(s_n, a_n) + \zeta^N Q(s_N, \mu_\theta(s_N))|s_0, a_0\right].$$  

(25)

3) Multiple Distributed Parallel Actors: This process takes place in parallel in $K$ actors, each one generating samples independently. The samples are collected in a replay table from which a learner samples batches to update the networks.

4) Prioritized Experience Replay (PER): Finally, D4PG collects $R$ samples from the replay buffer with non-uniform probability $p_i$. The $i$-th sample is selected with priority $(R p_i)^{-1}$ that also indicates the importance of the sample.
V. DRL-BASED BEAMFORMING WITH DISTRIBUTED LEARNING

A. Distributed Learning Algorithm

In the previous section, we utilized the D4PG learning method to solve the beamforming optimization problem (11). We enhanced the learning performance by allocating an agent per eAP and then utilizing the distributed experience collected from several agents. The enhanced method converges faster and shows better performance compared to the fully centralized DDPG solution. However, the proposed D4PG method still conducts the learning process at the ECP that involves a large body of the computational tasks.

In this section, we propose a cell-free beamforming scheme, which, in addition to distributing the agents’ experiences, splits the learning process of the DRL among all eAPs. In such a model, the eAPs divide the computational tasks equally and the ECP only performs limited control and coordination task. In this scheme, every eAP is responsible to find the optimal beamforming vector for all users. To this end, all vectors of the overall beamforming matrix are considered to be constants, to be simultaneously found by other eAPs. As an example, for a cell-free network with $M$ eAPs and $K$ users, eAP $m$ is responsible to optimize $\omega_m = W (m, [1 \ldots k])$ as shown in Fig. 3. Thus, eAP $m$ solves the following subproblem:

$$P_m : \max_{\omega_m \in [0,1]^{M \times K}} \sum_{k=1}^{K} \log_2 (1 + \gamma_k)$$

$$\text{s.t. } C_1 : \sum_{m=1}^{M} \left( w_{m_1}^2 - \sum_{i=\delta_l+1}^{l} w_{mi}^2 \right) \bar{\gamma}_{ml} \geq P_s,$$

$$C_2 : ||\omega_{mk}||^2 \leq 1,$$

$$\forall k, \forall \delta_l = 1, \ldots, l - 1, \text{ and } l = 2, \ldots, K.$$  

(26)

In the following, we present a new system design for the cell-free network environment where the eAPs interact with each other to find the optimal beamforming vector.
We intend to solve the optimization problem in (11) through multiple eAPs solving the problem in (26). We thus distribute the learning process among eAPs by implementing a DDPG agent in each eAP with its local experience buffer, actor network, and critic network. Beamforming is then optimized as follows: The first step is the DDPG process that includes (i) generating experiences, and (ii) updating the network parameters using Bellman equation (12) and the policy gradient in (17). Each agent learns to optimize one row of the beamforming matrix. Here the presence of a coordinator is essential to guarantee and organize the sharing of the optimized rows between the other agents. At each episode, the eAPs discover multiple actions. They select the action with the highest reward in terms of the maximum normalized sum rate during the episode and send it to the coordinator. The coordinator receives all the rows from the eAPs, then concatenates them to create a new beamforming matrix. Finally, it broadcasts the new matrix to all eAPs. The number of updates of the beamforming matrix, i.e. the horizon, can be selected based on the required accuracy. Algorithm 2 summarizes the learning process.

B. Complexity Analysis

Previously, we discussed that the beamforming optimization problem (11) is non-convex. Conventional approaches to solving such a problem include iterative algorithms such as exhaustive search, steepest descent, gradient descent, and interior-point methods.\(^1\) Considering the exhaustive search technique, we quantize the beamforming vector of user \(k\), i.e., \(\omega = [w_{1k} \ldots w_{MK}]\), by a certain step size \(\Delta\). For the beamforming optimization problem, this results in a complexity of \(O \left( \frac{1}{\Delta}^{M+K} \right)\). The exponential complexity renders the conventional beamforming techniques impractical, in particular, for dense cell-free networks. This fact calls for novel methods such as DRL-based solutions that retain low complexity while guaranteeing efficiency. Table III compares the inference floating-point operations per second (FLOPS) complexity order, as well as the convergence rate, of the three proposed DRL-based methods. The number of FLOPS during the inference is mainly determined by the matrix multiplications of the policy network, which has four layers with size \(|S|, 256, 128, \text{ and } |A|\). We compute the number of FLOPS of every fully

\(^1\)Note that the objective function in (11) is twice differentiable.
**Algorithm 2**

1. Initiate the cell-free network environment with random beamforming matrix $W_{local}$.  
2. Initialize actor network $\mu(\cdot)$ and critic network $Q(\cdot)$ with parameters $\theta^\mu$ and $\theta^Q$ respectively.  
3. Initiate the random process $\mathcal{N}$.  
4. Initiate $r_{\text{max}} = 0$.  
5. **for** each episode **do**  
6. Observe initial state $s$ from the cell-free network environment.  
7. **for** each step **do**  
8. Apply $a = \mu(s) + \mathcal{N}_t$ on the cell-free network environment and observe new state $s'$ and reward $r$.  
9. Store $(s, a, r, s')$ in the replay table $\mathcal{R}$.  
10. Update $\theta^\mu$ and $\theta^Q$ weights with a batch from $\mathcal{R}$ using Bellman equation (12) and policy gradient (17).  
11. **if** $r \geq r_{\text{max}}$ **then**  
12. $a_{\text{optimal}} = a$.  
13. $r_{\text{max}} = r$.  
14. **end if**  
15. **end for**  
16. **if** update time of beamforming matrix **then**  
17. Send $a_{\text{optimal}}$ to the coordinator.  
18. Receive $W_{local}$ from the coordinator.  
19. $W_{local} = W_{\text{new}}$.  
20. **end if**  
21. **end for**

Connected layer as the product of the size of its weight matrix. For D4PG with distributed training, the number of FLOPS is equal to that of one DDPG since the feed-forward operation on the policy network remains identical during the inference. In contrast, for distributed D4PG, all $N$ eAPs run the inference to predict one row of the beamforming matrix. Therefore, the number of FLOPS is $N$ times higher than that of D4PG with distributed experience and DDPG. In the next section, we investigate the performance of DRL models in more detail and also evaluate the convergence rate.
Algorithm 3 Coordinator

1: repeat
2: \( W_{\text{new}} = 0 \)
3: repeat
4: if \( a_{\text{optimal}} \) is received from \( i \)-th eAP then
5: \( W_{i,\text{new}} = a_{\text{optimal}} \).
6: end if
7: until \( a_{\text{optimal}} \) is received from every eAP.
8: Broadcast \( W_{\text{new}} \) to all agents.
9: until all agents stops training.

VI. Numerical Analysis

A. Simulation Parameters

We numerically evaluate the proposed beamforming methods in terms of transmission performance (sum rate) and convergence rate. Table IV summarizes the most important parameters of the simulation setting.

| Parameter                           | Value          |
|-------------------------------------|----------------|
| AWGN PSD per UE                    | \(-169 \text{ dBm/Hz}\) |
| Path-loss exponent, \( \kappa \)   | 2              |
| Nakagami fading parameters, \( (M, \Omega) \) | \((1, 1)\) |
| Training sequence length, \( \tau_p \) | \( K \) Samples |
| Pilot transmission power, \( \rho_k \) | 100 mW, \( \forall k \) |
| SIC sensitivity, \( P_s \)         | 1 dBm         |

For simplicity, we assume the following: (i) For every AP \( m \) and every user \( k \), \( h_{mk} \) is a rv with \( M_{mk} = M = 1 \) and \( \Omega_{mk} = \Omega = 1 \); (ii) For every AP \( m \), the AWGN has PSD \( \sigma_m/2 = -169 \text{ dBm/Hz} \); (iii) Concerning large-scale fading, all eAPs and UEs are uniformly distributed over a disc of radius \( r = 18 \) meters, implying a coverage area of \( 1 \text{km}^2 \). The ECP knows the large-scale fading of each user.

To benchmark the proposed DRL schemes, we simulate the proposed centralized system model using the ‘gradient descent’ algorithm, which can find the local minimum of any first-order differentiable function [35]. We train the proposed models by using Python and TensorFlow 2.1.0 for 10 episodes with 1000 steps per one episode. Actor- and critic networks have fully-connected layers with two hidden
layers of 256 and 128 neurons followed by the ReLU activation function in each. The dimension of the final layer for the actor network and its corresponding target network is defined in the cell-free network environment, depending on the approach followed: In the first two approaches the dimension of the output layer is equal to the number of elements in the beamforming matrix. In the third approach, it is equal to the number of elements of one row. To include the constraints imposed in the problem formulation, we use the sigmoid activation function. The hyperparameters of the DRL-model are as follows: discount factor $\zeta = 0.99$, learning rate $\nu = 0.001$ for both actor and critic networks, a Poylak averaging parameter $\tau = 0.005$, and size of experience replay buffer $R = 10^6$. We use Adam for the critic and actor optimizer. In the distributed experiencing approach, we use $N$-step returns $N = 5$ and 51 atoms in the distributional representation with $V_{\text{min}} = -20$ and $V_{\text{max}} = 100$. We use the binary cross-entropy as the metric of the distance between distributions.

To verify the effectiveness of the proposed scheme, we evaluate the performance in terms of instantaneous reward defined by (11). We first train the proposed methods under three possible network models, namely: (i) small-scale cell-free network with $M = 15$ and $K = 5$; (ii) medium-scale cell-free network with $M = 50$, $K = 15$; iii) large-scale cell-free network with $M = 70$ and $K = 20$.

**B. Results**

Fig. 4 shows the normalized transmission sum rate versus the overall number of training steps in the small-scale setting. From this figure, it is obvious that the centralized DRL-based beamforming with distributed learning (D4PG) exhibits the best performance. Moreover, the performance of the fully distributed DRL-based beamforming is better than that of the centralized DRL-beamforming (DDPG). Indeed, in a small network, the performance of the DRL method at each eAP is almost identical to that of the conventional optimization methods (such as the steepest descent iterative algorithms) as the beamforming vector is low-dimensional. Furthermore, the D4PG achieves the closest performance to the gradient descent-based beamforming solution (without learning) with about 90% performance after 6000 learning steps.

Fig. 5 shows the numerical results in a medium-scale cell-free network. The centralized DRL-based beamforming with distributed experience retains its superiority over other methods; however, the performance gap concerning the fully centralized DDPG is smaller. Moreover, the fully distributed DRL-based beamforming is no longer superior to the centralized DDPG. The reason is that in the distributed setting, every eAP uses the beamforming vectors found by the other eAPs in previous iterations. This introduces some noise in treating the ICI component from other eAPs.
Fig. 4: Performance of different models under small-scale scenario.

Fig. 5: Performance of different models under medium-scale scenario.
In Fig. 6 we consider the large-scale setting. The gap between the distributed DRL-based beamforming and other methods with centralized learning (DDPG and D4PG) increases significantly. Additionally, the centralized DRL-based beamforming with distributed experience maintains better performance compared to DDPG. Moreover, compared to the fully centralized DDPG method, it converges in fewer steps. Nevertheless, after a relatively large number of training steps, the DDPG algorithm performs slightly better than the D4PG algorithm.

In Fig. 7 we evaluate the performance of the proposed DRL models with larger actor-critic neural networks (NN) sizes. Specifically, we increase the size of both actor- and critic-NN from $256 \times 128$ to $400 \times 300$. A larger size of the NNs results in a larger performance gap between the fully centralized DRL-based beamforming (DDPG) and the DRL-based beamforming with distributed experience (D4PG). Moreover, the DRL-based beamforming with distributed learning is not affected by the increase in the size of NNs. The reason is that in distributed learning, the size of the optimization variables per eAP is small so that the best possible performance is achievable even for small actor-critic NNs ($256 \times 128$).

Finally, we compare the running time of the gradient descent method and that of the proposed DRL models as a function of the problem dimension. We simulate the inference time of the DRL agent's
policy. We recall that the inference in deep learning is a feedforward propagation for a trained neural network. Thus, for simulation, we use the inference time of the policy network proposed in the DRL-based beamforming (DDPG) approach since its policy network architecture is identical to that of DRL-based beamforming with distributed experience (D4PG). Moreover, it is bigger than the policy network architecture of DRL-based beamforming with distributed learning (output layer of the centralized method is much bigger than that of the distributed approach). The learning rate of the gradient descent algorithm to solve the optimization problem is $\alpha = 0.1$. The problem dimension is defined by the number of eAPs $M$ and the number of users $K$ in the network. Here we set $K = \frac{M}{3}$ and we vary $M$ in the range of $[15, \ldots, 150]$. Fig. 8 shows that the DRL-models solve the problem faster than the gradient descent algorithm. The solution time of the gradient descent algorithm grows exponentially with the dimension of the network, whereas the DRL model requires less than a second for finding the optimal beamforming matrix.
VII. CONCLUSION

We studied the beamforming optimization problem in cell-free networks. First, we considered a fully centralized network and designed a DRL-based beamforming method based on the DDPG algorithm with continuous optimization space. We have also enhanced this method by collecting distributed experiences from geographically-distributed eAPs (D4PG). Afterward, we developed a DRL-based beamforming design with distributed learning, which divides the beamforming optimization tasks among the APs. Even though the D4PG beamforming technique demonstrates a promising performance, it still conducts the learning process, i.e., most of the computational tasks, at an ECP unit. This feature makes D4PG less appealing compared to the fully distributed solution. A future research direction is to generalize the proposed DRL beamforming models to comply with different wireless network parameters such as those for the propagation environment. As an example, the learning rate (discount factor) for the DRL algorithm can be adapted depending on the propagation and interference conditions in the network.
Algorithm 1 D4PG-Based Beamforming With Distributed Agents Experience

1: Input: batch size $M$, trajectory length $N$, number of actors $K$, replay size $R$, initial learning rates $\alpha_0$ and $\beta_0$, time for updating the target networks $t_{target}$, time for updating the actors policy $t_{actor}$.

2: Initialize network $(\theta, w)$ at random.
3: Initialize target weights $(\theta', w') \leftarrow (\theta, w)$.
4: Launch $K$ actors and replicate their weights $(\theta, w)$.

5: for $t = 1, \ldots, T$ do
6: Sample $M$ transitions $(s_{i:i+N}, a_{i:i+N}, r_{i:i+N})$ of length $N$ from replay with priority $p_i$.
7: Construct the target distributions $Y_i = \sum_{n=0}^{N-1} \zeta^n r_{i+n} + \zeta^N Z_w'(s_{i+N}, \mu_\theta'(s_{i+N}))$.
8: Compute the actor and critic updates
   \[
   \delta_w = \frac{1}{M} \sum_i \nabla_w (R_{p_i})^{-1} d(Y_i, Z_w(s_i, a_i)),
   \]
   \[
   \delta_\theta = \frac{1}{M} \sum_i \nabla_\theta \mu_\theta(s_i) \mathbb{E}[\nabla a Z_w(s_i, a)|a = \mu_\theta(s_i)].
   \]
9: Update network parameters $\theta \leftarrow \theta + \alpha_t \delta_\theta$, $w \leftarrow w + \beta_t \delta_w$.
10: If $t \mod t_{target} = 0$, update the target networks $(\theta', w') \leftarrow (\theta, w)$.
11: If $t \mod t_{actors} = 0$, replicate network weights to the actors.
12: end for
13: Return policy parameters $\theta$.

Algorithm 2 Actor

1: repeat
2: Select action $a = \mu_\theta(s) + \mathcal{N}(0, 1)$. Receive reward $r$ and observe state $s'$.
3: Store $(s, a, r, s')$ in replay.
4: until learner finishes, i.e., Algorithm 1 stops.

REFERENCES

[1] P. Popovski, K. F. Trillingsgaard, O. Simeone, and G. Durisi, “5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view,” IEEE Access, vol. 6, pp. 55 765–55 779, 2018.

[2] S. Zhou, M. Zhao, X. Xu, J. Wang, and Y. Yao, “Distributed wireless communication system: A new architecture for future public wireless access,” IEEE Mag. Commun., vol. 41, no. 3, pp. 108–113, March 2003.
[3] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, “Cell-free massive MIMO versus small cells,” *IEEE Trans. on Wireless Commun.*, vol. 16, no. 3, pp. 1834–1850, March 2017.

[4] E. Björnson and L. Sanguinetti, “Making cell-free massive MIMO competitive with MMSE processing and centralized implementation,” *CoRR*, vol. abs/1903.10611, 2019. [Online]. Available: http://arxiv.org/abs/1903.10611

[5] M. Bashar, K. Cumanan, A. G. Burr, M. Debbah, and H. Q. Ngo, “On the uplink max-min sinr of cell-free massive MIMO systems,” *IEEE Trans. on Wireless Commun.*, vol. 18, no. 4, pp. 2021–2036, April 2019.

[6] Y. Al-Eryani and E. Hossain, “The D-OMA method for massive multiple access in 6G: Performance, security, and challenges,” *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 92–99, Sep. 2019.

[7] G. Interdonato, E. Björnson, H. Quoc Ngo, P. Frenger, and E. G. Larsson, “Ubiquitous cell-free massive MIMO communications,” *EURASIP J. on Wireless Commun. and Net.*, vol. 2019, no. 1, p. 197, Aug. 2019.

[8] M. Attarifar, A. Abbasfar, and A. Lozano, “Modified conjugate beamforming for cell-free massive MIMO,” *IEEE Trans. on Wireless Commun.*, vol. 8, no. 2, pp. 616–619, April 2019.

[9] E. Björnson and L. Sanguinetti, “Scalable cell-free massive MIMO systems,” *IEEE Trans. on Commun.*, pp. 1–1, 2020.

[10] A. Liu and V. K. N. Lau, “Joint BS-user association, power allocation, and user-side interference cancellation in cell-free heterogeneous networks,” *IEEE Trans. Signal Processing*, vol. 65, no. 2, pp. 335–345, 2017.

[11] Y. Jin, J. Zhang, S. Jin, and B. Ai, “Channel estimation for cell-free mmWave massive MIMO through deep learning,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10325–10329, 2019.

[12] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary, and M. Asaduzzaman, “Quantum machine learning for 6G communication networks: State-of-the-art and vision for the future,” *IEEE Access*, vol. 7, pp. 46317–46350, 2019.

[13] S. Chakraborty, E. Bjørnson, and L. Sanguinetti, “Centralized and distributed power allocation for max-min fairness in cell-free massive MIMO,” in *2019 53’rd Asilomar Conf. on Signals, Systems, and Computers*, 2019, pp. 576–580.

[14] R. Nikbakht, A. Jonsson, and A. Lozano, “Unsupervised-learning power control for cell-free wireless systems,” in *IEEE 30’th Annual Int’l Symposium on Pers., Indoor and Mobile Radio Commun. (PIMRC’19)*, 2019, pp. 1–5.

[15] O. Simeone, “A very brief introduction to machine learning with applications to communication systems,” *IEEE Trans. on Cognitive Commun. and Networking*, vol. 4, no. 4, pp. 648–664, 2018.

[16] A. Abdallah and M. M. Mansour, “Angle-based multipath estimation and beamforming for FDD cell-free massive MIMO,” in *IEEE 20’th Inter’l Workshop on Signal Proc. Advances in Wireless Commun. (SPAWC’19)*, 2019, pp. 1–5.

[17] G. Femenias and F. Riera-Palou, “Cell-free millimeter-wave massive MIMO systems with limited fronthaul capacity,” *IEEE Access*, vol. 7, pp. 44596–44612, 2019.

[18] Y. Al-Eryani, E. Hossain, and D. I. Kim, “Generalized coordinated multipoint (GCoMP)-enabled NOMA: Outage, capacity, and power allocation,” *IEEE Trans. on Commun.*, vol. 67, no. 11, pp. 7923–7936, Nov. 2019.

[19] Y. Al-Eryani, M. Akrout, and E. Hossain, “Multiple access in dynamic cell-free networks: Outage performance and deep reinforcement learning-based design,” *preprint*, arXiv:2002.02801, 2020.

[20] E. Nayebi, A. Ashikhmin, T. L. Marzetta, H. Yang, and B. D. Rao, “Precoding and power optimization in cell-free massive MIMO systems,” *IEEE Trans. on Wireless Commun.*, vol. 16, no. 7, pp. 4445–4459, 2017.

[21] S. Chakraborty, E. Bjørnson, and L. Sanguinetti, “Centralized and distributed power allocation for max-min fairness in cell-free massive MIMO,” in *2019 53’rd Asilomar Conf. on Signals, Systems, and Computers*, 2019, pp. 576–580.

[22] A. Zhou, J. Wu, E. G. Larsson, and P. Fan, “Max-min optimal beamforming for cell-free massive MIMO,” *IEEE Commun. Lett.*, pp. 1–1, 2020.

[23] A. Abdallah and M. M. Mansour, “Efficient angle-domain processing for FDD-based cell-free massive MIMO systems,” *IEEE Trans. on Commun.*, vol. 68, no. 4, pp. 2188–2203, 2020.
[24] C. D’Andrea, A. Zappone, S. Buzzi, and M. Debbah, “Uplink power control in cell-free massive MIMO via deep learning,” in 2019 IEEE 8'th Int’l Workshop on Comput. Advances in Multi-Sensor Adaptive Proc. (CAMSAP’19), 2019, pp. 554–558.

[25] Z. Chen, E. Bjørnson, and E. G. Larsson, “Dynamic resource allocation in co-located and cell-free massive MIMO,” IEEE Trans. on Green Commun. and Net., vol. 4, no. 1, pp. 209–220, 2020.

[26] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” arXiv preprint arXiv:1509.02971, 2015.

[27] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. Tb, A. Muldal, N. Heess, and T. Lillicrap, “Distributed distributional deterministic policy gradients,” arXiv preprint arXiv:1804.08617, 2018.

[28] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-learning,” in Thirtieth AAAI conf. on artificial intelligence, 2016.

[29] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, “Cell-free massive MIMO: Uniformly great service for everyone,” in 2015 IEEE 16th Int’l Workshop on Signal Proc. Advances in Wireless Commun. (SPAWC), 2015, pp. 201–205.

[30] R. M. Gray and L. D. Davisson, An Introduction to Statistical Signal Processing, 1st ed. New York, NY, USA: Cambridge University Press, 2010.

[31] H. Yin, D. Gesbert, M. Filippou, and Y. Liu, “A coordinated approach to channel estimation in large-scale multiple-antenna systems,” IEEE J. on Sel. Areas in Commun., vol. 31, no. 2, pp. 264–273, Fe. 2013.

[32] W. Liao, T. Chang, W. Ma, and C. Chi, “QoS-based transmit beamforming in the presence of eavesdroppers: An optimized artificial-noise-aided approach,” IEEE Trans. Signal Processing, vol. 59, no. 3, pp. 1202–1216, 2011.

[33] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, “Deterministic policy gradient algorithms,” in Proceedings of the 31st Int’l Conf. on Machine Learning, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. Beijing, China: PMLR, 22–24 Jun 2014, pp. 387–395.

[34] R. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, D. TB, A. Muldal, N. Heess, and T. Lillicrap, “Distributed distributional deterministic policy gradients,” Conference at ICLR, 2018. [Online]. Available: https://arxiv.org/pdf/1804.08617.pdf

[35] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press, 2004.