MR-iNet Gym: Framework for Edge Deployment of Deep Reinforcement Learning on Embedded Software Defined Radio

Jithin Jagannath[1], Kian Hamedani[1], Collin Farquhar[1], Keyvan Ramezanpour[1], Anu Jagannath[1]
[1]Marconi-Rosenblatt AI/ML Innovation Lab, ANDRO Computational Solutions LLC
Rome, NY, USA
{jjagannath,khamedani,cfarquhar,kramezanpour,ajagannath}@androcs.com

ABSTRACT

Dynamic resource allocation plays a critical role in the next generation of intelligent wireless communication systems. Machine learning has been leveraged as a powerful tool to make strides in this domain. In most cases, the progress has been limited to simulations due to the challenging nature of hardware deployment of these solutions. In this paper, for the first time, we design and deploy deep reinforcement learning (DRL)-based power control agents on the GPU embedded software defined radios (SDRs). To this end, we propose an end-to-end framework (MR-iNet Gym) where the simulation suite and the embedded SDR development work cohesively to overcome real-world implementation hurdles. To prove feasibility, we consider the problem of distributed power control for code-division multiple access (DS-CDMA)-based LPI/D transceivers. We first build a DS-CDMA ns3 module that interacts with the OpenAI Gym environment. Next, we train the power control DRL agents in this ns3-gym simulation environment in a scenario that replicates our hardware testbed. Next, for edge (embedded on-device) deployment, the trained models are optimized for real-time operation without loss of performance. Hardware-based evaluation verifies the efficiency of DRL agents over traditional distributed constrained power control (DCPC) algorithm. More significantly, as the primary goal, this is the first work that has established the feasibility of deploying DRL to provide optimized distributed resource allocation for next-generation of GPU-embedded radios.

CCS CONCEPTS

• **Networks → Network protocols**; • **Computing methodologies → Machine learning**.

KEYWORDS

Deep reinforcement learning, software defined radio, GPU, power allocation, machine learning

ACM Reference Format:
Jithin Jagannath[1], Kian Hamedani[1], Collin Farquhar[1], Keyvan Ramezanpour[1], Anu Jagannath[1]. 2022. MR-iNet Gym: Framework for Edge Deployment of Deep Reinforcement Learning on Embedded Software Defined Radio. In *Proceedings of WiSeML '22: 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSeML '22).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3522783.3529530

1 INTRODUCTION

Wireless communication has been evolving rapidly to keep up with the requirement of the consumers both in the commercial and tactical domain. As compared to wired networks, the challenges posed by dynamic and rapidly changing channel is well known. In addition, the resources available for wireless devices are limited and do not follow the exponential growth trajectory of the number of wireless devices. These factors dictate the need for intelligent decision engines that can learn from the environment to dynamically use the scarce resources while meeting the requirements of the end customer.

The rapid growth of computational resources and availability of digital data has revived machine learning from its second winter. Machine learning has been revolutionizing several domains when applied judiciously. The wireless domain is no exception to these influences. There have been significant efforts made to effectively leverage various strengths of machine learning to solve crucial hurdles encountered in the wireless domain [7, 9, 12, 14]. For example, supervised learning is being widely deployed to perform blind signal characterization and classification; several approaches are being leveraged to detect intrusions, anomalies, and enhance the security of wireless systems including the Internet-of-Things (IoT) ; and reinforcement learning is being used in several applications for dynamic resource allocation. Even with the recent advances, there are several limitations and challenges that have held back the real-world deployment of machine learning based decision engines in the wireless domain. In this paper, we propose an end-to-end framework and related methodology for deployment of deep reinforcement learning (DRL) for a distributed wireless network. To accomplish this, we choose the problem of distributed power control in low probability of intercept and detection (LPI/D) networks which play a crucial role in tactical communication. Direct sequence code-division multiple access (DS-CDMA) is especially popular for implementing a robust LPI/D physical layer due to its unique advantages, such as easy frequency management (no frequency planning) in multi-user scenarios, low peak-to-average power ratio (PAPR), and less stringent synchronization requirements as necessary for orthogonal frequency division multiplexing (OFDM). We would like to emphasize the fact that this is just one example of an optimization problem that can be solved using the proposed framework which is the focus of this work.

Deploying machine learning at the edge (on embedded platform) has always been a formidable challenge in the wireless domain. To the best of our knowledge, this is the first work that puts forth an

WiSeML '22, May 16–19, 2022, San Antonio, Texas, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/3522783.3529530

end-to-end framework to demonstrate the feasibility of designing, training, and deploying DRL-based decision engines for *next generation of embedded software defined radio (SDR) hardware.* The key contributions and impact of this work are as follows,

- We develop the first known DS-CDMA network architecture in ns3-gym to ensure rapid training of the DRL module.
- Next, we perform optimization of the trained model to ensure lightweight deployment on embedded SDR to minimize the inference time and ensure real-time decision making.
- The results of our over-the-air experiments indicate that DRL consumes significantly less power while maintaining high reliability by taking into account higher dimensional input yet making a real-time decision.
- This is the first work that establishes the feasibility of leveraging DRL for distributed decision making in the wireless network *using GPU-embedded transceiver.* We believe edge deployment of machine learning solutions will be a key enabler for future networks including but not limited to 6G.
- Overall, the progress and impact of this work can arguably be abstracted from the specific optimization (resource allocation) problem discussed here.

2 RELATED WORK

Machine learning has been emerging as a tool suited to solve several challenging problems in the domain of wireless communication [7, 9, 10, 12]. Yet, there are several hurdles that need to be overcome to transition some of these solutions from concept to simulations to actual hardware implementation. The majority of the advances have been limited to simulations due to the challenges associated with demonstrating the effectiveness of machine learning-based approaches on actual radio hardware. It is also important to point out that in many cases (especially in the case of supervised learning) hardware is used to collect relevant data or execute the decision of a machine learning engine but the actual computation still happens on large host PCs with ample computing resources [5, 8]. Here we argue that in several cases for machine learning to truly impact the wireless domain, techniques and solutions will have to be deployed on the devices themselves using the embedded computing resources that could include GPP, GPU, or FPGA. This argument can also be seen in the light of the importance and every increasing attention edge computing has been witnessing to meet the QoS constraints (including and not limited to latency).

In[4, 11], distributed constrained power control (DCPC) algorithm was introduced where the power is adjusted iteratively by DCPC given the signal-to-interference-plus-noise ratio (SINR). In this approach, i.e., DCPC, a minimum required SINR threshold is considered and the optimization process continues as long as the minimum required quality of service (QoS) is achieved. Later, several other algorithms [2, 6, 17, 19] were introduced which were the modified version of [4]. In[6], a power and admission control algorithm with beamforming is introduced where the coexistence of the primary users (PUs) and secondary users (SUs) in a shared spectrum is guaranteed while the total transmission power is minimized. The hard SINR constraint issue in distributed power control is addressed in[19] where a utility based power control (UBPC) is presented. In this approach[19], the problem is reformulated using a softened SINR constraint (utility) and a penalty for power consumption (cost) where the goal is to maximize the net utility which is equal to utility without the cost. In[2], a twofold multiuser interference management approach for wireless ad hoc networks is introduced where the objectives are to increase single-hop throughput and reduce the power consumption. In this approach[2], distributed power control is performed to achieve the admissible power vector which can be used by the scheduled transmitters to satisfy their single-hop requirements. A distributed power allocation algorithm is presented in[17] where the PUs communicate under a minimum QoS requirement, whereas the SUs opportunistically utilize the primary band. Fractional programming (FP)[15] and weighted minimum mean square error (WMMSE)[16] are the two iterative centralized power control algorithms which are model driven [13]. In the two aforementioned centralized power control algorithms, i.e., FP and WMMSE, the delay caused by the feedback exchange mechanism between a central controller and the users [13] could deteriorate performance in a distributed LPI/D network. On the other hand, the existing power control algorithms require perfect CSI knowledge and are model driven. The data driven methods, e.g., DRL do not require *perfect CSI knowledge*, hence, they are more promising for realistic wireless environments. To tackle the challenges of conventional power control algorithms, in this paper for the first time, we train and deploy the DRL agents on GPU-embedded SDRs. In particular, we train and deploy actor-critic (AC) networks with deep deterministic policy gradient (DDPG) for power control in a distributed network of GPU-embedded AIR-T SDRs. *Additionally, all the recent works discussed above have been limited to only simulations.*

3 PROBLEM FORMULATION

3.1 An overview of Reinforcement Learning

In a general Markov decision process (MDP), a single or multiple agents interact with their surrounding environment. For each action taken by the agents, a feedback reward and the new state is returned by the environment. At each given time step, t, the environment's essential features or state space (observation space), $s^{(t)} \in \mathcal{S}$, is observed by the agent where $\mathcal{S}$ is the set of all possible states. The agent picks an action $a^{(t)}$ from the set of all possible actions, $\mathcal{A}$, based on a policy which could be either deterministic and stochastic. The deterministic and stochastic policies are denoted by μ, and π, respectively, where $a^{(t)} = \mu(s^{(t)})$ or $a^{(t)} = \pi(s^{(t)})$. After the $a^{(t)}$ is taken by the agent, the environment's state moves to a new state, namely, $s^{(t+1)}$. The transition probability from state-action pair, i.e., $< s^{(t)}, a^{(t)} >$ to the new state $s^{(t+1)}$ is denoted by $P^{a^{(t)}}_{s^{(t)} \longrightarrow s^{(t+1)}} = Pr(s^{(t+1)}|s^{(t)}, a^{(t)})$. The reward signal, $r^{(t+1)}$, which is returned from the environment, determines how good or bad the action is. The set of described interactions is called as an experience at $t+1$ and is denoted as $e^{(t+1)} = (s^{(t)}, a^{(t)}, s^{(t+1)}, r^{(t+1)})$.

In model free reinforcement learning, these interactions are learned without any prior knowledge about the transition probabilities and the optimal policy is achieved through maximizing the long-term accumulative discounted return of agent at time t,

$$G^{(t)} = \sum_{i=0}^{\infty} \gamma^i r^{(t+i+1)}, \tag{1}$$

where $\gamma < 1$ denotes the discount factor. DDPG is a well known actor-critic based technique which supports continuous action spaces. In a DDPG network, a critic network is trained iteratively which learns the action value function. The optimal deterministic policy is determined by another network, namely, actor.

3.2 Power Control Algorithm

We assume K nodes (wireless transceivers) with spreading sequence matrix $\mathrm{S} \in \mathbb{C}^{L \times K}$ forming $K/2$ transmit-receive pairs operating simultaneously in the same frequency and L corresponds to the length of the spreading codes.

To maintain distributed operation and minimize overhead, we use only local information and minimal information gathered from immediate neighbors. The state of a node i at time slot τ can be expressed as

$$\mathcal{S}_i^\tau = \{\mathbb{D}_i^\tau, B_i^\tau, I_i^\tau, S_i^\tau\} \tag{2}$$

where $\mathbb{D}_i^\tau = \{d_{ij}^\tau | j = 1, 2, \cdots, K\}$ is the set of distances to neighboring receivers, B_i^τ is the number of packets in the buffer, I_i^τ is the interference caused by a transmission from node i (can be estimated by node i), and S_i^τ is the interference sensed by the receiver of node i. The value for interference sensed at the receiver is communicated to node i through an ACK. If the transmission from node i is not received, then it will not receive an ACK message and the value for S_i^τ is set to -1. The action of node i at time slot τ is given by

$$\mathcal{A}_i^\tau = \{a_i | a_i \in \mathbb{A}\}$$

where the action space is the set of available power levels $\mathbb{A} = [0.1\text{mW}, 5\text{mW}]$. This is a hardware specific choice and can be customized when the target hardware capabilities change.

We implement a reward function which aims to maximize SINR on successful transmission and penalize based on interference caused for unsuccessful transmission (*i.e.,* transmission does not complete with an acknowledgement). The SINR of a transmission from node i to node j at a given slot τ is represented as Γ_{ij}^τ. Let us define the interference caused by a transmission from agent i to agent j be $I_{ij} = \sum_{k \neq j \neq i} \frac{a_k}{d^\alpha}$ where the summation index k goes over all neighboring nodes where d is the distance and α is the path loss coefficient.

The reward for taking an action $\mathcal{A}_i$ is given below.

$$R_i = \begin{cases} \tilde{\Gamma_{ij}} \text{if transmission successful} \\ -\tilde{I_{ij}}, \text{if transmission unsuccessful} \end{cases} \tag{3}$$

Where $\tilde{\Gamma_{ij}}$ and $\tilde{I_{ij}}$ are, respectively, the normalized SINR and normalized interference caused from a transmission from agent i to agent j. We normalize these quantities by experimentally finding the largest value likely to be observed and then using this value as a divisor. We emphasize that this is only one formulation of a resource allocation problem that can leverage the proposed framework. It can be easily seen how this could be applied to a plethora of wireless network optimization problems.

4 MR-INET GYM FRAMEWORK

In this work, we present the Marconi-Rosenblatt framework for intelligent networks (MR-iNet Gym) shown in Fig. 1. The MR-iNet Gym facilitates the designing, training, and deployment of the DRL decision engine on embedded SDR. To accomplish the objective of deploying DRL on embedded transceiver hardware, we identify three main components, (i) A high fidelity simulator that facilitates design and setup of network scenarios while allowing the transceivers (agents) to run DRL decision engines with ease, (ii) methodology to optimize (from computation standpoint) the trained model for embedded deployment on wireless transceivers, and (iii) a flexible embedded software defined radio hardware that can house the AI-enabled communication protocol stack.

4.1 Simulation Suite for AI-enabled Networks

For our simulation environment, to ensure rapid development, we leverage several open source software packages for C++ and Python. We begin by building the first known custom DS-CDMA module for ns-3 to simulate a distributed LPI/D wireless network. Next, we use OpenAI Gym [1] to handle the reinforcement learning training loop. This enables us to rapidly implement several DRL algorithms to compare and fine-tune the best-performing model. In this work, we focus on DDPG with continuous action space as it maps naturally onto the power control problem. Finally, we use ns3-gym [3] to serve as a gateway between the simulation environment provided by ns-3 with reinforcement learning structure provided by Gym. Inside of Gym training loop our reinforcement learning agent then receives state information, uses this information to select an action, and then passes the selected action back to the environment. The information flow between ns-3 and Gym that ns3-gym facilitates is shown in Fig. 2.

4.2 Target Embedded Software Defined Radio

In the conventional embedded SDRs, General Purpose Processing (GPPs) or Field Programmable Gate Arrays (FPGAs) are commonly used to provide the processing capabilities. Neither of these is ideal for deploying DRL decision engines and often leads to undesired latency. Therefore, deploying the deep learning (DL)-based solutions on the SDRs will even further reduce their speed which makes the adoption of the conventional SDRs in the real-world applications impractical in the fifth generation (5G) and beyond communication systems. Therefore, it is required to utilize the parallel processing computational power of the graphical processing units (GPUs) in the embedded SDRs for: 1) offloading the processing of GPP/CPU onto GPU to accelerate the signal processing; 2) making it practical to deploy the DL-based solutions while not affecting the speed of the signal processing algorithms through the parallel processing capabilities of the embedded GPU. Due to the aforementioned reasons, in this work, we implement and deploy our signal processing operations and AC power control engines on the *Deepwave Digital's Artificial Intelligence Radio Transceiver (AIR-T) which is the world's first SDR equipped with embedded GPUs.* The AIR-Ts (shown in Fig. 7) are designed and developed to operate in the DL-based radio frequency (RF) applications and they are equipped with embedded NVIDIA GPU, an FPGA, and dual embedded CPUs.

4.3 Bridging the Reality Gap

Another key roadblock to overcome is the reality gap between simulations and hardware. Even with reliable simulations, it is extremely difficult to capture the unpredictable behaviors of the wireless channels and its instantaneous effects on the physical layer of modern

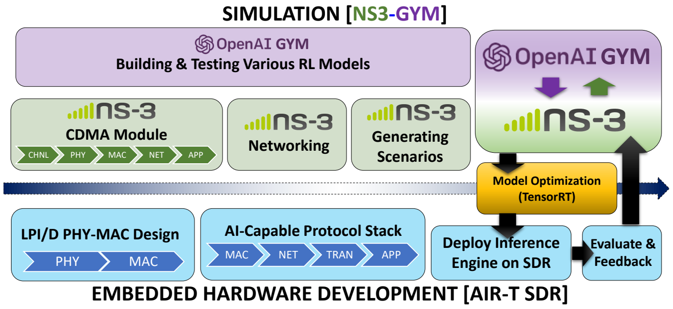

Figure 1: MR-iNet Gym Framework for Intelligent Networks

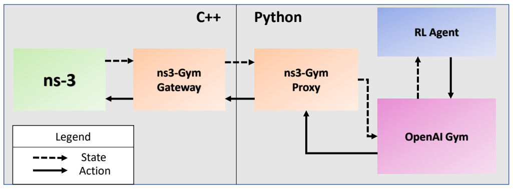

Figure 2: ns3-gym information flow: Dotted lines indicate the transfer of state information. Solid lines indicate the communication of the action selected by the RL agent.

communication systems. While certain abstractions at the physical layer are appropriate and highly beneficial in reducing the complexity of the simulator, it is important those abstractions happen in an informed manner trying to represent hardware-specific nuances and model the interaction with the expected channel conditions as closely as possible. In other words, it is important to feedback the finding regarding generalized hardware-specific operation characteristics from hardware evaluation back into the simulation suite to minimize the degradation that can be caused by these reality gaps. This interaction is represented by the interaction between simulation and hardware deployment in Fig. 1.

5 SIMULATION AND TESTBED EXPERIMENTS

Our testbed shown in Fig. 3 consists of six AIR-T SDRs which in total establish three CDMA-based communication pairs. Due to the space restriction, we do not elaborate on the physical layer implementation but the solution can be extended to any CDMA based tactical or commercial network. We implement *AI-enabling protocol stack* on SDR to ensure communication and interfaces between layers so that information can flow from different layers to the DRL engine. The DRL agents are deployed on each transmitter and the power control decisions are made by the agents after they receive feedback from their corresponding receiver.

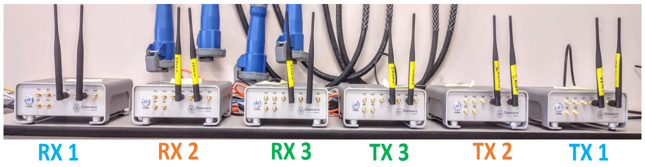

Figure 3: 6-node GPU-embedded SDR testbed.

5.1 Training in NS3-GYM

We train our AC power control agents with DDPG algorithm in ns3-gym environment. The AC networks constitute an actor network and a critic network. After trying different configurations, the final actor network is a fully connected DNN with one hidden layer where there are 10 neurons in the hidden layer. We use rectified linear unit (ReLU) used as the activation function in all the layers. The critic network is a fully connected DNN with four hidden layers where there are 16, 32, 32, and 256 neurons in each hidden layer, respectively. Similarly, ReLU function is adopted as the activation function in all layers. The training is performed using the reward function given in Eq 3 and the reward plot of the AC agents trained by DDPG algorithm in ns3-gym environment is plotted in Fig. 4. As it can be seen in Fig.4, the reward plot of the AC agents respectively converging within 2000 steps. However, as it can be seen, there is a high variance between different runs for the agents while the environment has not changed. We will address this issue in section 5.2 by introducing an approach to improve the stability of DDPG actors.

5.2 Stabilizing DDPG With Model Aggregation

In a DDPG network, the policy is determined by a non-linear neural network which is very sensitive to its parameters, namely, θ. This implies that a small variation in θ could result in a very large difference in the policy that is represented by the actor network. In Figure 4, the reward plots of three agents on three different runs on the same environment have been demonstrated. As it can be seen, the variance of the rewards among different runs for the same agent is very high. In other words, although our environment has not changed, the DDPG agents are converging to different points amongst different runs. This behavior implies that the DDPG agents are not demonstrating a good stability. This is mainly caused due to the non-convex optimization problem, where a small change in the gradient steps can lead to a different policy which could be arbitrarily a bad solution[18]. In general, policy-based DRL methods such as DDPG perform poorly when compared against tabular methods in terms of stability.

Therefore, in this paper, we introduce an approach which improves the stability of policy-based DRL methods, in particular DDPG network. We apply our approach on DDPG network and verify its effectiveness through extensive simulations. However, without loss of generality, this approach can be applied on any policy-based DRL method. The approach that we propose to stabilize the DDPG network is based on model aggregation. Combining different models together is a well know approach in machine learning to improve the robustness and decrease the variance of the models' outputs. In this way, the sensitivity of the models to initial parameters and noise is decreased. Algorithm 1 is the pseudo code representing our approach for model aggregation and stabilizing of DDPG agents.

We perform the model aggregation of the DDPG actors every 100 steps. As it can be seen in Fig 5, after performing the model aggregation, the actors show much less variance during the training among several runs compared to the case where no model aggregation has been performed, i.e., Fig 4.

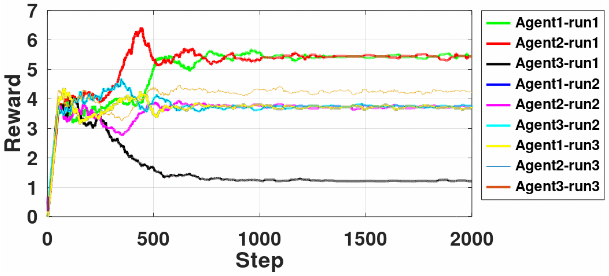

Figure 4: Agents' rewards w/o model aggregation

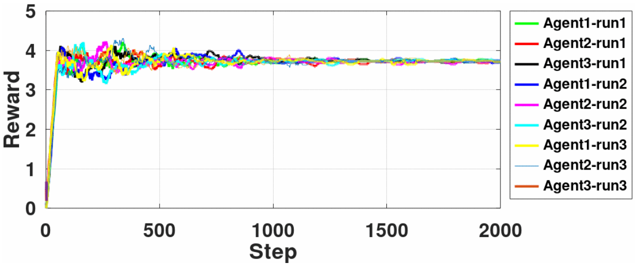

Figure 5: Agents' rewards with model aggregation.

Algorithm 1 Model Aggregated DDPG (MA-DDPG)

```
$N$ = Number of actors
$L$ = Number of layers in each actor
$\theta_i$ = actor $i$
$\omega_{li}$ = weight of layer $l$ of actor $i$
    $\omega_{l_{agg}}$ = weights of layer $l$ after model aggregation $l = 1, 2, \ldots, N$
    $\theta_{agg}$ = aggregated actor model
$S$ = Number of training steps
initialize $\theta_i$ $i = 1, 2, ..., N$
for $steps = 1, 2, \ldots, S$ do
    for $i = 1, 2, \ldots, N$ do
        Run policy $\pi_{\theta_i}$ in environment for 100 time steps and update $\theta_i$
$steps$ % 100 = 0
        for $l = 1, 2, \ldots, L$ do
            $\omega_{l_{agg}}$ = average([$\omega_{li}$]) $i = 1, 2, \ldots, N$
        end for
      $\theta_{agg}$ = clone([$\omega_{l_{agg}}$]) $l = 1, 2, \ldots, L$
      $\theta_i$(i=1,2,...,N)=$\theta_{agg}$
    end for
end for
```

5.3 Optimizing the DRL Model for Hardware

To ensure real-time operation, it is essential to accelerate the decision making process of the DRL-based inference engines on the SDRs. Our preliminary results indicate that adopting the DNN framework directly on the SDRs, can cause high latency in the delivery of the packets, hence it is not an optimal approach for a delay-sensitive application. In our SDR hardware setup, the Nvidia Jetson TX2 module is the accelerating hardware module and the TensorRT is the software framework for accelerating the DNN inference engines. TensorRT is comprised of an inference optimizer and a runtime which together deliver low-latency and high-throughput for resource allocation on SDRs. In Fig. 6, the flowchart of converting the DNN models to the optimized TensorRT format is depicted.

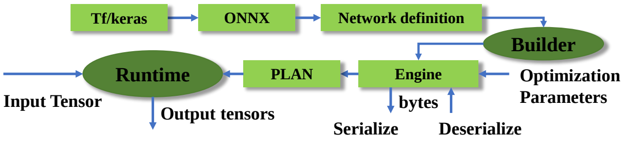

Figure 6: Flowchart of TensorRT model optimization.

Firstly, the Tensorflow based DNN model of the DRL network which is trained in the ns3-gym environment is saved. Consequently, this model is converted to the ONNX format. Then, the inference engine is built by TensorRT which can be serialized to PLAN file. Our SDR-based resource allocation method utilizes TensorRT to control the GPU-based DNN inference engine. In this paper, for the first time, the performance of a SDR-based power allocation algorithm which relies on TensorRT and CUDA to optimize the DNN inference engine is presented. In Table 1 a comparison between the original model(.h5 format), ONNX format and PLAN (optimized by TensorRT) is demonstrated with respect to the average inference time of 30000 runs. The inference time of the original model, and ONNX model are measured on GPP. However, the inference time of the PLAN file (optimized by TensorRT) is measured on the GPU (cannot run on GPP). As the results show, the total inference time of the optimized model is significantly lower than the original, and ONNX models, respectively, while the precision of model has not been compromised.

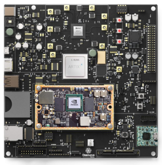

Figure 7: AIR-T SDR

Format	Inference time
Original	2266 μs
ONNX	14.6 μs
PLAN	0.23 μs

Table 1: Inference time of a DNN model in different formats

5.4 DRL Over-the-air Results

To deploy the models on the AIR-T SDRs, the PLAN files are required to be linked to the DS-CDMA physical layer to perform inference. Our DS-CDMA physical layer is developed purely in C++, however, the PLAN inference engines are CUDA and GPU friendly. To manage the complexity in a large program like ours, it is essential to break down the large program into smaller components. In fact, the CUDA-based PLAN inference engines and C++ codes are compiled separately and are linked together using the linking tools that have been made available since CUDA 5.0. The experiments were conducted in the 2.45 GHz ISM band. The power of the first packet is set by the user (can be set to the max value to ensure reliability). Upon the successful delivery of the first packet at the receiver, an ACK (acknowledgment) message is transmitted from the receiver to the transmitter which contains two key elements (i) SINR, and (ii) the interference sensed at the receiver. The information required for the DRL which included SINR, interference, buffer length, and the distance of each transmitter from its intended receiver and other receivers as well, are fed to the DRL agents to choose optimal action (optimal power between 0 to 5 mW) for the transmission of the next packet. For each experiment 1000 packets were transmitted and in total 16 experiments were performed where each experiment corresponded to a different combination of

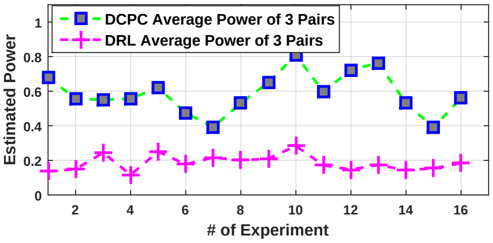

Figure 8: Estimated power for DRL and DCPC.

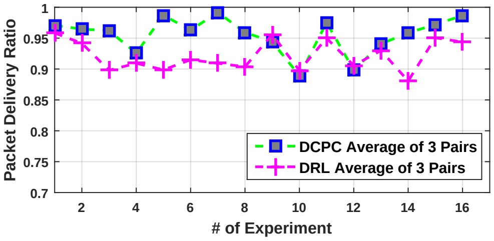

Figure 9: The average PDR for DRL and DCPC.

Format	Power	PDR
DRL	0.19mW	.922
DCPC	0.59mW	.955

Table 2: Performance Comparison of DRL agents and DCPC Algorithm

the transmitter and receiver gains to represent different operation constraints. To evaluate the performance of the DRL agents, we compare their performance against DCPC[4, 11], in terms of PDR and total power consumption.

Fig. 8 depicts the estimated average power of 1000 packet transmission for different experiments. As it was mentioned, each experiment corresponds to a different combination of the transmitter and receiver gains. Fig. 9 shows the average PDR of DCPC and DRL. The power consumption of DCPC is significantly higher than of DRL without a significant gain in PDR. In Table II, the average performance metrics of DCPC and DRL are presented. The results indicate that the power consumption of DCPC is $\approx$ 3 times higher.

6 CONCLUSION AND FUTURE WORK

Machine learning has been rapidly emerging as a tool to accelerate the strides towards realizing the next-generation of radios. While there have been several advancements in this domain, the majority of them have been limited to simulations or relied on host-PC with ample computational resources. Due to the inherent need for mobility, energy-efficiency, and other strict Quality-of-Service requirements, it is inevitable that machine learning models will have to live on embedded devices (at the edge) to ensure AI-enabled radios can make real-world impacts. In this work, for the first time in literature, we have used GPU-embedded SDR to demonstrate real-time (<μs) distributed decision making for a resource allocation problem in distributed wireless networks. To accomplish this, we proposed the MR-iNet Gym framework that orchestrates a cohesive development process. This leverages ns3-gym simulation for training, optimizing the trained model for hardware deployment, and achieving superior performance on GPU-embedded SDR hardware in our over-the-air experiments.

In the future, we hope to publish an extended version discussing the physical layer and AI-enabling protocol stack implementation in detail. In 2022, we also hope to release the ns3-gym software since it is the first one developed for the CDMA network. We are actively utilizing MR-iNet Gym to address a wide variety of network optimization problems which we hope to discuss in our future work. Finally, we hope the contribution of this paper has a broader impact that motivates the community to adopt the design ethos of MR-iNet Gym to solve open research challenges in the wireless domain.

ACKNOWLEDGMENT AND DISCLAIMER

This material is based upon work supported by the US Army Contract No. W15P7T-20-C-0006. Any opinions, findings, and conclusions or recommendation expressed in this material are those of the author(s) and do not necessarily reflect the views of the US Army.

REFERENCES

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. 2016. OpenAI Gym. arXiv:1606.01540 [cs.LG]

[2] Tamer ElBatt and Anthony Ephremides. 2004. Joint scheduling and power control for wireless ad hoc networks. *IEEE Transactions on Wireless communications* 3, 1 (2004), 74–85.

[3] Piotr Gawłowicz and Anatolij Zubow. 2019. ns-3 meets OpenAI Gym: The Playground for Machine Learning in Networking Research. In *ACM Intl. Conf. on Modeling, Analysis and Sim. of Wireless and Mobile Systems* (Miami Beach, USA).

[4] Sudheer A Grandhi, Jens Zander, and Roy Yates. 1994. Constrained power control. *Wireless Personal Communications* 1, 4 (1994), 257–270.

[5] Ade Hermawan, Rizki Ginanjar, Dong-Seong Kim, and Jae-Min Lee. 2020. CNN-Based Automatic Modulation Classification for Beyond 5G Communications. *IEEE Communications Letters* 24, 5 (2020), 1038–1041.

[6] Md Habibul Islam, Ying-Chang Liang, and Anh Tuan Hoang. 2007. Distributed power and admission control for cognitive radio networks using antenna arrays. In *2007 2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*. IEEE, 250–253.

[7] Anu Jagannath, Jithin Jagannath, and Tommaso Melodia. 2021. Redefining Wireless Communication for 6G: Signal Processing Meets Deep Learning with Deep Unfolding. *IEEE Transaction on Artificial Intelligence* (2021).

[8] Jithin Jagannath, Nicholas Polosky, Dan Connor, Lakshmi Theagarajan, Brendan Sheaffer, Svetlana Foulke, and Pramod Varshney. 2018. Artificial Neural Network based Automatic Modulation Classifier for Software Defined Radios. In *Proc. of IEEE Intl, Conf. on Communications (ICC)*. Kansas City, USA.

[9] Jithin Jagannath, Nicholas Polosky, Anu Jagannath, Francesco Restuccia, and Tommaso Melodia. 2019. Machine Learning for Wireless Communications in the Internet of Things: A Comprehensive Survey. *Ad Hoc Networks (Elsevier)* (2019).

[10] Jithin Jagannath, Nicholas Polosky, Anu Jagannath, Francesco Restuccia, and Tommaso Melodia. 2020. Neural Networks for Signal Intelligence: Theory and Practice,. In *Machine Learning for Future Wireless Communications*, F.L. Luo (Ed.). John Wiley & Sons, Limited.

[11] Xingjian Li, Jun Fang, Wen Cheng, Huiping Duan, Zhi Chen, and Hongbin Li. 2018. Intelligent power control for spectrum sharing in cognitive radios: A deep reinforcement learning approach. *IEEE access* 6 (2018).

[12] Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. 2019. Applications of Deep Reinforcement Learning in Communications and Networking: A Survey. *IEEE Communications Surveys Tutorials* 21, 4 (2019), 3133–3174.

[13] Yasar Sinan Nasir and Dongning Guo. 2020. Deep Actor-Critic Learning for Distributed Power Control in Wireless Mobile Networks. In *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 398–402.

[14] Keyvan Ramezanpour and Jithin Jagannath. 2022. Intelligent Zero Trust Architecture for 5G/6G Networks: Principles, Challenges, and the Role of Machine Learning in the context of O-RAN. *arXiv preprint arXiv:2105.01478* (2022).

[15] Kaiming Shen and Wei Yu. 2018. Fractional programming for communication systems—Part I: Power control and beamforming. *IEEE Transactions on Signal Processing* 66, 10 (2018), 2616–2630.

[16] Haoran Sun, Xiangyi Chen, Qingjiang Shi, Mingyi Hong, Xiao Fu, and Nicholas D Sidiropoulos. 2018. Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing* (2018).

[17] John Tadrous, Ahmed Sultan, Mohammed Nafie, and Amr El-Keyi. 2010. Power control for constrained throughput maximization in spectrum shared networks. In *IEEE Global Communications Conference (GLOBECOM)*.

[18] Chen Tessler, Nadav Merlis, and Shie Mannor. 2019. Stabilizing off-policy reinforcement learning with conservative policy gradients. (2019).

[19] Mingbo Xiao, Ness B Shroff, and Edwin KP Chong. 2003. A utility-based power-control scheme in wireless cellular systems. *IEEE/ACM Transactions On Networking* 11, 2 (2003), 210–221.