Power Management of Wireless Sensor Nodes with Coordinated Distributed Reinforcement Learning

Shaswot Shresthamali, Masaaki Kondo and Hiroshi Nakamura
The University of Tokyo
{shaswot, kondo, nakamura}@hal.ipc.i.u-tokyo.ac.jp

***Abstract*—Energy Harvesting Wireless Sensor Nodes (EHWSNs) require adaptive energy management policies for uninterrupted perpetual operation in their physical environments. Contemporary online Reinforcement Learning (RL) solutions take an unrealistically long time exploring the environment to converge on working policies. Our work accelerates learning by partitioning the state-space for simultaneous exploration by multiple agents. We achieve this by using a novel coordinated ϵ-greedy method and implement it via Distributed RL (DiRL) in an EHWSN network. Our simulation results show a *four-fold increase* in state-space penetration and reduction in time to achieve optimal operation by an order of magnitude (50x). Moreover, we also propose methods to reduce instances of disastrous outcomes associated with learning and exploration. This translates to reducing the downtimes of the nodes in simulations corresponding to a real-world scenario by one thirds.**

***Keywords*-Distributed Reinforcement Learning, Deep Reinforcement Learning, Energy Harvesting Wireless Sensor Nodes, Energy Neutral Operation, Internet of Things, ϵ-greedy exploration**

I. Introduction

Energy-Harvesting Wireless Sensor Nodes (EHWSNs) and their networks constitute an integral part of the Internet of Things (IoT) ecosystem. Adaptive power management policies ensure uninterrupted operation of EHWSNs without the need for human intervention even when the working environment is complex and unpredictable. When a node consumes all of the energy harvested without exceeding its battery limits, Energy Neutral Operation (ENO) is achieved [1]. Judicious regulation of duty cycles for ENO has been achieved through Reinforcement Learning (RL) methods [2]. For the sake of example, we base our discussions on a simulated solar EHWSN system [2], shown in Fig. 1, hereafter referred to as ENO-RL. The ENO-RL evaluates various duty cycling schemes by trial-and-error. A scalar reward acts as a feedback that is used to further optimize its policies and maximize node utility (by maximizing its duty cycles without violating ENO.) This learning-based method is adaptive by nature and dispenses with the need for hand-crafted optimizations which makes it very suitable for a wide range of application scenarios.

During learning, the RL agent has to explore a vast state-action space (for instance, the space of all the combinations of different duty cycles, battery levels and harvested energy) in search of optimal policies. On the other hand, it is also equally important for the agent to maximize its utility by behaving optimally. ϵ-greedy methods are a simple yet practical method to manage the exploration-exploitation trade-off. Using this method, the agent acts greedily according to its learned policy, but with a probability ϵ, it takes a random exploratory action.

Motivation: Ideally we would prefer EHWSNs to acclimatize in their working environment and reach maximum utility as soon as possible. However, acting greedily prematurely leads to weak policies that cannot deal with anomalous and rare states that ultimately reduces the overall utility. Robust policies require longer exploration periods which translates to lost opportunities to maximize utility and higher probability of disastrous outcomes. Here disastrous outcomes refer to battery violations when the node goes out of operation due to insufficient energy (*downtimes*) or when there is an excess of harvested energy that cannot be utilized nor stored in the battery (*overflows*). Downtimes result in reduced coverage and rerouting which directly affects the network quality of service. Overflows decrease the energy efficiency of the node.

Good exploration of the working environment during RL produces better policies. However simply having a higher exploratory rate does not necessarily translate to good exploration. This is because some regions of the state-action space may not be easily accessible through naive undirected exploration. For example, the chances of an agent with very low battery reserves exploring higher states of battery level are extremely slim if the duty cycles are randomly chosen. As a result of insufficient and inefficient exploration, the sub-optimal solutions obtained are not robust. A possible workaround is to use simulators and historical data with offline training to extensively pretrain the nodes. However this is not always a

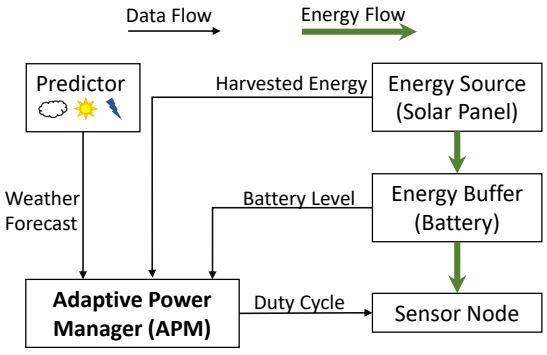

Fig. 1. The ENO-RL system for a solar-EHWSN. The Adaptive Power Manager (APM) uses RL to regulate the node's energy consumption via duty cycling.

feasible solution, especially for unknown environments.

Contribution: In this work, we propose to coordinate the many nodes of the sensor network to explore different regions of the vast problem state-space. For this, we base our work on a conventional Distributed RL (DiRL) [3] system. In a DiRL system (Section IV-A), multiple nodes interact with the environment concurrently and their experiences are pooled together in a central server. The server learns from these experiences and broadcasts the updated policies back to the nodes. A naive DiRL implementation does not however imply state-space partitioning or efficient exploration. We propose a novel exploration method, ϵ-pref, to partition the state-space among sensor nodes for efficient exploration. This implicitly results in higher risk of disastrous outcomes for some nodes. We propose ϵ-safe to deal with the trade-off between efficient exploration via state-space partitioning and the risk it entails. Additionally, since each node of the sensor network generally deals with its unique environment, it is important that it adapts to perform optimally as fast as possible. This is possible by decreasing instances of unnecessary exploration. To this end, we present ϵ-adapt to dynamically adjust the exploration rate. Our main contributions in this work are:

- We propose ϵ-pref to partition the state-space for efficient exploration by coordinating the ϵ-greedy behavior in a DiRL system. We achieve this by allotting each node in an EHWSN network with a preferred exploratory action such that each node explores a specific region of the state-space (Section IV-B).
- We also propose ϵ-safe, a technique to distribute the risk associated with exploration uniformly among all nodes so as to increase the collective performance without compromising heavily on exploration efficiency (Section IV-C).
- Additionally, we propose ϵ-adapt, to automatically adapt the exploration rate so as to minimize unnecessary exploration and decrease the learning time and cost (Section IV-D).

The rest of the paper is organized as follows. Section II gives a brief overview of the related research. Section III presents the theoretical background on ENO, RL and DiRL. Our novel exploration schemes for the DiRL framework are explained in Section IV. Section V describes our evaluation methodology and comparison metrics. Section VI contains the results of our experiments and we conclude with Section VII.

II. Related Work

ENO for EHWSNs was first formally described in [1] where the authors use a predictive system coupled with linear programming optimization methods. RL-based approaches for optimizing communication policies in EHWSNs were reported in [4]–[6]. Tabular RL methods for ENO of EHWSNs, discussed in [2], [7], have very limited applications because they require discrete states and actions with very long training times. The authors in [8] improved upon this to using function approximation with RL policy-gradient methods. Our work is based on model-free value-function approximation for RL using Deep Q Networks (DQNs) [9], described in Section III-B. Our work can also be easily extended to policy gradient [10], [11] and actor-critic RL [12] methods.

DQNs are known to be sample inefficient and therefore suffer from lengthy learning times. They require a memory pool from which the neural network (NN) extracts minibatches during training. Populating this memory pool is expensive in terms of time and performance. The authors in [3], [13] propose DiRL to populate this pool via concurrent interaction of multiple agents with the environment. However since their application domain (playing computer games) and ENO-RL are very different, naive implementation of their method is not optimal. Computer games are artificial deterministic environments where disastrous outcomes incur no real costs unlike ENO-RL. We thus propose novel ϵ-greedy exploration strategies to not only minimize the learning time but also reduce the costs associated with learning. We would like to note that, in our work, the DiRL agents do not interact with each other, either directly or indirectly. This is not an unreasonable assumption to make for EHWSNs because harvesting ambient energy and sensing environmental data has practically no effect on the working environment. RL agents whose actions influence the environment experienced by other agents lie in the domain of multi-agent RL and is not in the scope of our research.

In [14]–[16], the authors present solutions to simultaneously manage the power and QoS constraints. Doing so increase the complexity of the RL problem, especially in regards to the RL reward function which has been discussed in [17]. The authors in [18] take a step further and optimize the operation of EHWSN while taking into account the *dynamic utility* of the data gathered by the sensor node. Model-based approaches are discussed in [19].

The exploration-exploitation dilemma in RL is still an open problem. ϵ-greedy and softmax action selection [10] are popular methods of exploration via introducing noise in the action space. The author in [20] gives an overview of different exploration techniques in action space. In contrast, Noisy-nets [21] and evolutionary strategies [22]–[24] inject noise into the parameters of the neural network to achieve exploration. While the above methods tackle *how* to explore, methods based on concepts of surprise [25], curiosity [26], selective attention [27], disagreement [28] and gradient ascent in information [29] try to answer the question of *when* to explore. These sophisticated exploration methods, while effective in selective domains, require non-trivial computational requirements and therefore have limited application in resource-constrained EHWSNs. In the end, ϵ-greedy methods give the best bang-for-buck with good exploration at minimal computational cost over other methods. In [30], the authors compare convergence-based and ϵ-greedy exploration strategies for single agent RL systems for EHWSNs.

Adaptive ϵ-greedy methods have been proposed in [31] where the authors propose a fusion of softmax and ϵ-greedy methods depending on the temporal difference error. However, this method is not applicable for DQNs. Our adaptive ϵ-

TABLE I
KEY TERMS USED

Term	Description	Term	Description
b_t	Battery Level	s_t	state
d_t	Duty Cycle	a_t	action
h_t	Harvested Energy	r_t	reward
z_t	Node Energy Utilization	π	policy
p_t	Energy Neutral Performance	e_t	experience
f_E	Weather Forecast	ϵ	exploration rate

greedy method follows similar logic but adapts the *change* in exploration rate as a function of the immediate reward. As to our knowledge, this is the first work that exploits ϵ-greedy methods for accelerating DiRL by introducing preferential exploratory actions.

III. BACKGROUND

This section presents the theoretical background behind ENO, RL and DQNs. For the rest of our discussion, we assume a discrete time model with discrete time steps t. T consecutive timesteps constitute an episode E. We consider the ENO-RL System (Fig. 1) that consists of a generic sensor node powered by a battery that is charged by a solar panel. The node's power consumption is determined by its operating duty cycle that is regulated by the adaptive power manager (APM). The APM uses RL to learn power management policies for duty cycling to achieve ENO. The APM also takes into account a rough weather prediction when choosing the duty cycles. Table I lists some of the key terms used throughout this paper.

A. ENO

Let us consider an EHWSN with an ideal battery of capacity b_{max}, capable of varying its power consumption via N_D discrete duty cycles $d_t \in [d_{min}, d_{max}]$, and can harvest a maximum of h_{max} energy per timestep t. We further assume that all of the harvested energy is first stored in the battery before being consumed, i.e., harvest-store-use system. At time t, the reserve battery level is $b_t \in [0, b_{max}]$, the harvested energy is $h_t \in [0, h_{max}]$ and the energy consumed by the node is $z_t = d_t \times z_{avg}$ where $0 < d_t \leq 1$ and z_{avg} is the average energy consumption of the node. The battery at the start of $t+1$ timestep is given by:

$$b_{t+1} = b_t + h_t - z_t \tag{1}$$

The least energy that can be consumed when the node is operational is $z_{min} = d_{min} \times z_{avg}$ corresponding to the lowest duty cycle. The node is operational at time t if $b_t \geq z_{min}$. *Downtimes* are instances when $b_t < z_{min}$ and the node is out of operation. We assume that higher duty cycles imply higher node utility. Energy is irrecoverably lost and *overflow* occurs when $b_t \geq b_{max}$.

We now define the ENO condition for the EHWSN. ENO requires that the amount of energy harvested equals the amount of energy consumed by the node in some time interval. This ensures that the node always has enough energy to operate (no downtimes) and harvested energy is not wasted due to overflow. The objective of the APM is therefore to optimize the duty cycles of the node such that downtimes and overflows are minimized and the utility (energy consumption) of the node is maximized. We lump battery inefficiencies and the power consumed by the APM together with the node's power consumption without any loss of generality. This simplifies the objective of ensuring ENO to:

$$b_t + h_t > z_{min} \tag{2}$$

$$b_t + h_t - z_t < b_{max} \tag{3}$$

From the above equations we can conclude that as long as $z_{min} < b_t < b_{max}$ for all t, the node is consuming all of the energy that it has harvested and is therefore energy neutral (assuming no downtimes or overflows). The energy neutral performance (ENP) gives a measure of the deficit or surplus of energy required to maintain ENO. If we assume that there are no downtimes or overflows, then the ENP at time t is given by $p_t = b_t - b_{init}$ where b_{init} is the battery level at the beginning of an episode E.

It must be noted that during some episodes, the total harvested energy may not be sufficient to power the node even at its lowest duty cycle. Alternatively, sometimes the energy harvested may be so plentiful that even with the highest duty cycle, some surplus energy would remain unused. The value of b_{init} plays an important role in such cases to ensure ENO. Although it is difficult to express b_{init} formally to guarantee ENO, in practice, the range of suitable b_{init} values can be roughly estimated using guidelines in [1]. Here, we assume that as long as the maximum deviation of b_{init} from b_{opt} is less that $b_{margin} \geq 0$, i.e., $|b_{init} - b_{opt}| \leq b_{margin}$, there exists a power management policy to achieve ENO. b_{opt} and b_{margin} are hyperparameters that can be estimated using the formulations in [1].

B. DQN

We consider a model-free RL agent that interacts with its environment in sequential discrete time steps. At each timestep, the agent observes its state s_t, executes an action $a_t \in \mathcal{A}$ where $\mathcal{A}$ is a set of possible actions, according to a policy $\pi(a|s)$ where π maps states s_t to actions a_t. Consequently the agent receives a scalar reward r_t and the environment changes its state to s_{t+1}. The return is defined as $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$ which is the total cumulative return from time t discounted by a factor $\gamma \in (0, 1]$.

For an agent following policy π, the Q-value function $Q^\pi(s,a) = \mathbb{E}[R_t | s_t = s, a_t = a]$ gives the *expected* return when executing action a from state s. The greedy action $\tilde{a} = \arg\max_a Q^\pi(s,a)$ maximizes this expected return. The optimal action-value function $\max_\pi Q^\pi(s,a)$, gives the maximum action-value for the state-action pair (s,a) that can be achieved by any policy.

We use a value-based RL using function approximation with NNs in this work. Specifically we use a variant of DQN called Double DQN [32] with dueling architecture [33]. On the i-th iteration, the action-value function, $Q(s,a;\theta_i)$, is approximated by a DQN with parameters θ_i. The agent stores its experiences $e_t = (s_t, a_t, r_t, s_{t+1})$ at each timestep t in a

memory pool $\mathcal{M} = \{e_1, e_2....e_t\}$. During learning, random minibatches of experiences $(s, a, r, \hat{s}) \sim \mathcal{Z}(\mathcal{M})$ are selected for updating θ (refer Appendix B).

IV. Proposed System

In this section, we describe the distributed ENO-RL system and propose our novel exploration strategies to accelerate learning.

A. Distributed RL

We first formulate the RL problem by defining the state-action space, the reward function and then extend it to a distributed system.

1) RL Formulation: The APM uses RL to learn policies to optimize the duty cycles of the node. For an episode E, the agent's state at time t is $s_t = (b_t, p_t, h_t, f_E)$. $f_E \in [1, 2...N_F]$ is a rough estimate of the predicted energy harvesting opportunity for episode E that can be easily obtained from the internet, e.g., weather forecast websites. The agent's actions a_t correspond to the possible duty cycles $d \in [\, d_{min}, d_{max}]$. The reward $r_E \in [-1, 1]$ is calculated only at the end of the episode E and all the state-action pairs that were encountered during E are equally credited with that amount. This reduces reward sparsity and stabilizes the learning process. r_E is based on the mean battery level b_E, for episode E (Fig. 10). We do not shape the reward to prefer any particular value of b_E as long as it is within the working range of $b_{opt} \pm b_{margin}$. The rewards decrease linearly as b_E deviates further from this range. A large b_{margin} implies greater uncertainty about the optimal battery level. While this increases the complexity of the problem, it greatly relaxes the effect of the choice of hyperparameter b_{opt} which is not easily obtainable. r_E is given by:

$$r_E = \begin{cases} 1, & \text{if } |b_{opt} - b_E| \leq b_{margin} \\ 1.5 - 5 \times \frac{|b_{opt} - b_E|}{b_{max}}, & \text{otherwise} \end{cases} \quad (4)$$

This reward function encourages the agent to maintain the battery levels within a safe working range $b_{opt} \pm b_{margin}$, while allowing the battery levels to fluctuate sufficiently during an episode. The constants used in the Equation (4) were determined empirically.

Motivating Example: Let us take two RL agents, *Red* and *Blue*, attempting to solve the ENO-RL system in a non-distributed framework. Red always chooses d_{max} during ϵ-greedy exploration and Blue always chooses d_{min}. Figure 2 shows the combined scatter plot of the state-space (ENP state not shown for clarity) visited by each of the agents after one year of training. Agent Red explores using a high duty cycle and consequently spends more time in regions of low battery levels. Similarly Blue has a lower duty cycle and therefore experiences higher battery levels. From this figure, we observe that it is possible to divide the state-action space between agents by biasing their preferred exploratory actions. Furthermore, we can expect each of these agents to learn from each others' experience to enhance their collective intelligence.

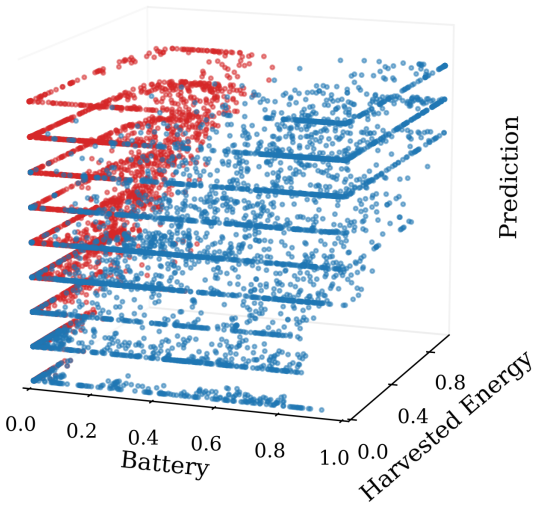

Fig. 2. A clear state-space partitioning as a result of different exploration policies of two agents, Red (high duty cycles) and Blue (low duty cycles). The discrete tiers correspond to each of the discrete weather prediction states (top one corresponds to a sunny day).

We make use of these insights and propose our coordinated exploration methods for the following DiRL Framework.

2) DiRL Framework: We consider a distributed EHWSN network that consists of one central *learner* and a fleet of N_w *workers*. The EHWSNs are the workers whereas the central network server is the learner. During the i-th episode a worker $w_k, k = 1, 2, ...N_w$, interacts with its unique environment according to a policy $\pi(w_k|\theta_i)$ and uploads its stream of experiences $(s_{k1}, a_{k1}, r_{k1}, \hat{s}_{k1}), ..., (s_{kT}, a_{kT}, r_{kT}, \hat{s}_{kT})$ to a global memory pool $\mathcal{M}$. The learner randomly gathers a minibatch of experiences $\mathcal{Z}(\mathcal{M})$ from this pool and performs N_l learning steps. The updated parameters θ_{i+1}, are then broadcast back to the workers. The workers do not execute any learning steps. We make this assumption because EHWSNs are typically too constrained in their computational resources to perform gradient-based learning. All workers receive the same set of parameters from the learner. Their policies differ only with respect to their exploratory behavior.

B. Partitioned ϵ-greedy Exploration: ϵ-pref

Let us define a function $\Omega(w_k) = k$ that maps each worker w_k to a unique real number k. Let $\tilde{a}$ denote the greedy action. For a given node w, the probability of taking an action $a \in \mathcal{A}$ is:

$$p(a|w) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{N_A}, & \text{if } a = \tilde{a} \\ C(a, w)\frac{\epsilon}{N_A}, & \text{otherwise} \end{cases} \quad (5)$$

For naive ϵ-greedy method, $C(a, w) = 1$ and all actions are equally probable to be the exploratory action. This acts as our baseline method for comparison between DiRL methods. We refer to it as ϵ-naive method. In this method, the diversity in experiences is assumed to be a consequence of the stochasticity of the unique environments experienced by each worker node. In cases when the stochasticity of the environment is insufficient to induce diverse exploratory behavior, the

experiences gathered are redundant. Consequently, while ϵ-naive methods increase the quantity of experiences to learn from, these experience are not necessarily diverse. We describe several methods to overcome this problem below.

The state values corresponding to the battery level (and therefore ENP) are highly correlated with the choice of actions as we saw in Fig. 2. In contrast, the states for harvested energy and weather prediction are completely independent of the node duty cycles. We can hope to divide the state-space only on the basis of battery level. We do so by dedicating each node with a preferred duty cycle for exploration by a *preference* factor of $C(a, w)$ as follows:

$$C(a,w) = \begin{cases} u + \frac{1-u}{N_w}, & \text{if } a = \Omega(w) \\ \frac{1-u}{N_w}, & \text{otherwise} \end{cases} \quad (6)$$

Equations (5) and (6) describe a coordinated exploratory scheme, referred to as ϵ-pref, where a node w is instructed by the central learner to prefer action $a = \Omega(w)$ for exploration by a probability u over other actions. As a result, even if all the nodes experience the same state, the state-action space is maximally explored because the nodes try out all the different possible actions (assuming $N_A \leq N_w$) which may possibly lead the nodes to different regions of the state-space.

C. Safe ϵ-greedy Exploration: ϵ-safe

ϵ-pref maximizes state-space exploration but as a result, some nodes always end up taking risky actions and facing disastrous results. This reduces the performance of the nodes and affects the overall system performance. To distribute the risks without compromising exploration, we propose a safe exploration method that we call ϵ-safe, where nodes switch their preferred exploratory action every iteration (or episode). If we redefine the condition for a in (6) for the i-th iteration as $a = (\Omega(w_k) + i) \mod N_A$, the preferred actions of all nodes change with every iteration. This way, each node explores an action for one iteration (T timesteps). Doing so allows the node to explore that action sufficiently without incurring too much risk. Furthermore, we also dynamically change the preference of exploration, u, such that $u \propto \epsilon$. This means that when the node is exploring at a high rate, it tends to prefer one particular action for exploration during an episode. In the next episode, it will explore with preference to another action. This way, during exploration, the node cycles through all its actions, trying one preferentially in each episode. As ϵ decreases, the preferences become less pronounced.

D. Adaptive ϵ-greedy exploration: ϵ-adapt

In ϵ-greedy methods, like the ones discussed above, a typical strategy is to start with a high value of ϵ and gradually anneal it as training progresses. This ensures high exploration at early stages of training. Naive ϵ annealing methods may result in sub-optimal learning because the agent may start acting greedily before it has explored enough and converge sub-optimally; or it may explore for a longer time than required thus missing out on chances for maximizing utility. Furthermore, one fixed annealing method cannot be expected to be optimal for all environments. We propose an adaptive scheme, referred to as ϵ-adapt, to automatically adapt the rate of exploration during learning based on achieved rewards. The basic idea is to explore more when rewards are low (i.e., the agent is behaving sub-optimally) and to behave progressively greedily as rewards increase. We propose an adaptive method where positive rewards reduce ϵ by a factor of β and negative rewards increase it by the same factor. Expressed mathematically, if the i-th episode acquires a reward r_i using an exploration rate ϵ_i, the exploration rate for next episode, $\epsilon_{i+1} \in [\epsilon_{min}, \epsilon_{max}]$, is given by:

$$\epsilon_{i+1} = \epsilon_i (1 - \beta \frac{r_i}{|r_i|}) \quad (7)$$

This method of exploration results in faster adaptation to the nodes' unique working environments and increase in utility.

V. Evaluation Methodology

We simulate the ENO-RL system using realistic values (see Appendix A) based on a TMote Sky [34], with solar radiation data for Tokyo from the Japanese Meteorological Agency. Their website [35] provides hourly data and therefore we assume each timestep to have a duration of one hour. An episode lasts for one day and so contains $T = 24$ time steps. We split up the days into $N_F = 10$ categories based on their total solar radiation and use it to simulate the coarse predictor of future energy, f_E. Every hour, the ENO-RL APM identifies its state s_t and chooses an action a_t corresponding to a duty cycle d_t from ten equally spaced discrete duty cycles ($N_D = 10, d_{min} = 10\%$). At the end of the day (episode), the APM calculates its reward r_E, and uploads the experience into its memory pool. The APM collect experiences for a week before it starts learning, after which learning takes place at every time step.

The DQN uses the hyperparameters listed in Appendix B. For DiRL systems, the nodes receive an update of their DQN parameters from the central server along with an ϵ-greedy policy directive. This directive sets the value of ϵ and $C(a, w)$ for each node. The nodes then interact with their respective environments and upload their experiences at the end of 24 hours (one episode). On receiving the experiences from the different nodes, the central server trains for $N_l = 1000$ iterations and then broadcasts the updated parameters and exploration directives back to the nodes.

We determined the annealing rate for ϵ empirically. In single node non-DiRL systems, we anneal ϵ from 0.9 to 0.01 at a rate of 0.1 per year. For the DiRL systems, ϵ_i for episode (or day) E_i is given by:

$$\epsilon_i = 0.9 - \frac{E_i}{E_i + 40} \quad (8)$$

We compare between different methods on the basis of battery violations (downtimes or overflows) because this is a direct indicator of ENO. We define the *learning time* to be the time required for a policy to achieve ENO. ENO is said to be achieved if the number of violations is less than

TABLE II
ENO-RL POLICIES

Policies	ϵ-greedy Type	RL Type	Preference, u
B-ENO	ϵ-naive	non-DiRL	N/A
D-ENO	ϵ-naive	DiRL	0.0
P-ENO	ϵ-pref	DiRL	0.5
S-ENO	ϵ-safe	DiRL	ϵ
A-ENO	ϵ-adapt, $\beta = 0.1$	DiRL	ϵ

24 in a span of 365 consecutive days. We also define the *learning cost* as the number of violations committed by a node before achieving ENO. For DiRL system, we sum the violation hours of *all* the nodes. Ideally we would like to minimize both learning cost and time. To compare the state-space penetration in a given time interval between different methods, we take the covariance $\sigma(b, h)$, of the state-values of battery b, and harvested energy h. This makes sense because the other state variables, ENP and prediction, are dependent on b and h. High values of $\sigma(b, h)$ indicate more state space coverage.

We want to answer the following questions through our experiments.

- What speedups can we expect to gain from DiRL methods?
- How does partitioning the state-space with ϵ-pref affect learning time and cost?
- Can we minimize the learning costs with ϵ-safe?
- Can an adaptive ϵ-adapt lower learning time and cost?

We compare four different policies for the ENO-RL system, summarized in Table II. All DiRL solutions use ten workers ($N_w = 10$) and one separate learner. We use B-ENO and D-ENO as baselines for comparison. The values of u, β and the annealing rate of ϵ were determined empirically.

Since the nodes of a DiRL system are expected to experience different environments, we simulate this by allowing each worker to interact with an environment based on solar data of Tokyo from different years [1]. B-ENO trains with the solar data starting from 1995. Once a system achieves ENO, we test its robustness by implementing it greedily for a period from 1995 to 2018. We also conduct analytical simulations where all the nodes in DiRL experience identical environments based on solar data for 2000. This is to remove the effect of the environmental stochasticity during exploration for fair analysis of our proposed exploration schemes,

VI. RESULTS

In this section, we present the results of our experiments.

A. Acceleration in Learning due to DiRL

Figure 3 shows the learning time for the different policies in a log plot. As expected, when we scale up from a single agent to a DiRL system with ten nodes, there is a corresponding dramatic decrease in learning time. With our proposed methods, we are able to achieve even better results and achieve speedups of up to 49.5x in the case of A-ENO. This means that with

[1]Ideally, we would like to have weather data from the same year but different locations in Tokyo - but due to unavailability of such data, we resort to this scheme.

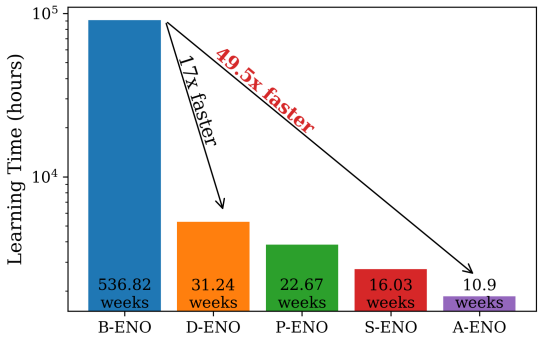

Fig. 3. A naive DiRL system (D-ENO) accelerates learning by 17x compared to a single agent RL (B-ENO). By coordinating the agents to explore efficiently and using an adaptive exploration rate, learning is accelerated by almost 50x (A-ENO). The numbers in the bars correspond to the time required to achieve ENO.

ϵ-adapt methods, an EHWSN network can start performing optimally within 10 weeks of deployment *without any prior training or human intervention*.

Table III lists the violation instances during training and testing of the different policies. Single node B-ENO has the largest number of violations - longer learning period means more violation instances. D-ENO commits fewer violations due to accelerated learning but we observe that it is not as robust as B-ENO during testing. Both B-ENO and D-ENO use ϵ-naive exploration. The results indicate that simply increasing the number of experiences using DiRL and relying on the stochasticity of the environment to provide diversity may not necessarily result in better policies. Due to inefficient exploration, *D-ENO learns fast but not enough*. The poor testing performance of D-ENO likely is the result of overfitting and early convergence to local optima.

B. State-space partitioning with ϵ-pref

P,S,A-ENO coordinate node exploration to explore a wider state-space to overcome the limitations of D-ENO. This is achieved by assigning each node a different preferential exploratory action. As a result, P-ENO fares better in the test compared to D-ENO (Table III); however, more exploration comes with higher learning costs. *P-ENO learns better but costs more.*

To further analyze the effects of coordinated exploration, we look at Fig. 4 and Fig. 5 obtained through analytical simulations where all nodes experience identical environments. Figure 4 shows the battery profiles for the different policies in the first week of training. In the case of D-ENO, all nodes have very similar battery profiles. However the nodes with P,S,A-ENO policies experience widely different battery levels as a result of coordinated exploration (ϵ-pref, ϵ-safe, ϵ-adapt). This diversity in the state-space exploration is further illustrated

TABLE III
NUMBER OF TRAINING AND TESTING VIOLATIONS

	B-ENO	D-ENO	P-ENO	S-ENO	A-ENO
Training	17722	7930	8817	**5392**	5921
Testing	0	20	16	8	**0**

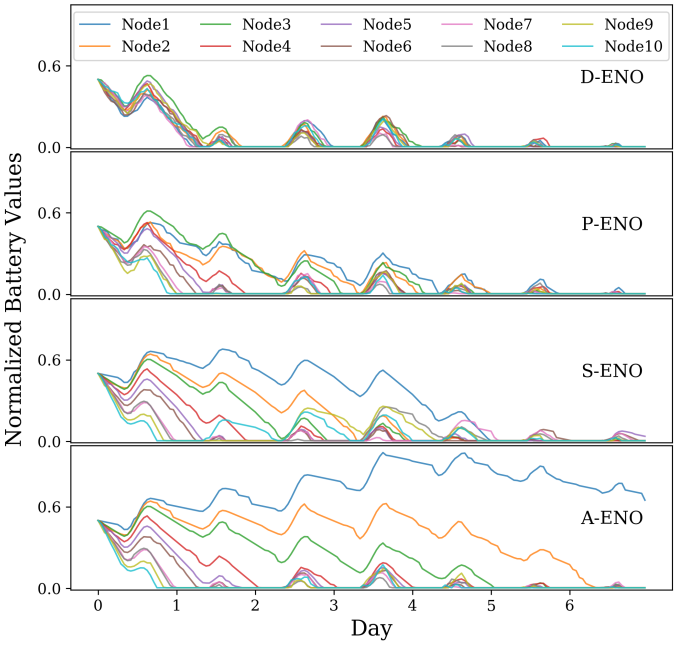

Fig. 4. Battery profiles for the first week of training. P,S,A-ENO experience diverse battery states compared to D-ENO as a result of coordinated exploration.

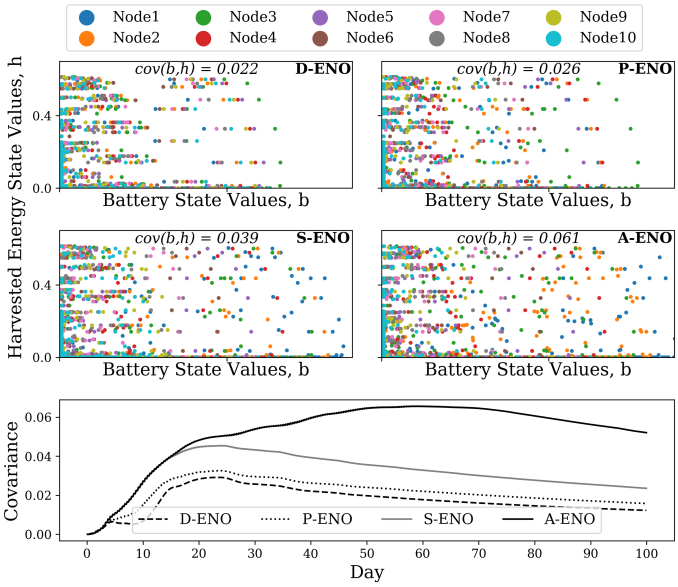

Fig. 5. Scatter plot of the states visited by different policies in the first two weeks of training. A-ENO has the most spread, i.e., better exploration compared to other methods. A-ENO has the highest covariance between the state values corresponding to battery levels and harvested energy during the first 100 days of training (bottom figure).

in Fig. 5 as a scatter plot of the states visited. The bottom part of the figure shows the covariance between battery levels and harvested energy, for each day, for the first hundred days. We observe that D-ENO has the least spread (and therefore least covariance) resulting in lesser than par performance. All the other policies cover a wider state-space. A-ENO has the highest variance (four times that of B-ENO) and experiences

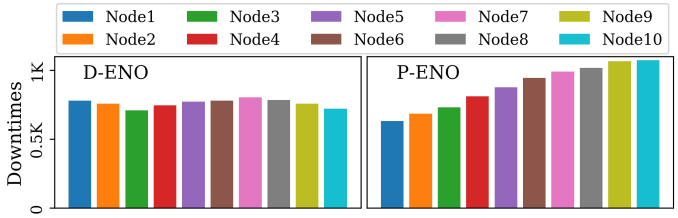

Fig. 6. The figure shows the cumulative downtimes of all the nodes for D-ENO and P-ENO in the first month. All nodes experience about the same number of downtimes in D-ENO. In contrast, owing to the risks of exploration by partitioning the state-space, the number of downtimes committed by each of the nodes of P-ENO vary significantly (higher duty cycle nodes experience more downtimes).

a wider range of battery levels (downtimes and overflows) in its very first week of training, resulting in reduced learning costs and superior acceleration. As expected, the covariance decreases as exploration rates decrease and the nodes act more greedily towards the end of the 100 day period.

C. Safe Exploration with ϵ-safe

We now discuss the effects of safe exploration. By assigning each node a different duty cycle with ϵ-pref, some nodes experience more violations than other. For instance, in the battery profiles for P,S,A-ENO in Fig. 4, Nodes 8-10 drain their batteries quicker than other nodes because they explore using higher duty cycles. Figure 6, also obtained through analytical simulations, illustrates this more clearly. It is clear from the figure that some nodes are at more risk of downtimes due to their exploration policy.

While it is inevitable that the nodes suffer through these violations to learn better, we can better distribute these risky situations to reduce learning costs. This is achieved by ϵ-safe in S-ENO. By cycling the exploratory actions with every iteration, the learning costs are reduced: S-ENO has lower costs *and* better results compared to D-ENO and P-ENO as shown in Table III. With safe exploration we can reduce the learning costs by three times compared to B-ENO. This illustrates that it is possible to distribute the risk of exploration without compromising the policy performance.

A comparison of the cumulative learning costs for all DiRL methods is shown in Fig. 7. A-ENO has a slightly larger learning cost compared to S-ENO, a trade-off for better test results, i.e., *S-ENO trades off safety with robustness.* D-ENO is able to decrease its downtimes when compared to P-ENO but commits many more overflows instead. This indicates improper function approximation due to insufficient variety in training experiences. P-ENO has the highest downtimes because the risks associated with ϵ-pref are unevenly distributed.

In D,P,S-ENO, all the nodes follow the same annealing rate for ϵ. This may not be optimal for nodes to adjust their unique working environment. We get better results by dynamically adapting the exploration rate with ϵ-adapt.

D. Adaptive Exploration with ϵ-adapt

For A-ENO, the exploration rate depends on the reward received in the previous episode. A positive reward entices

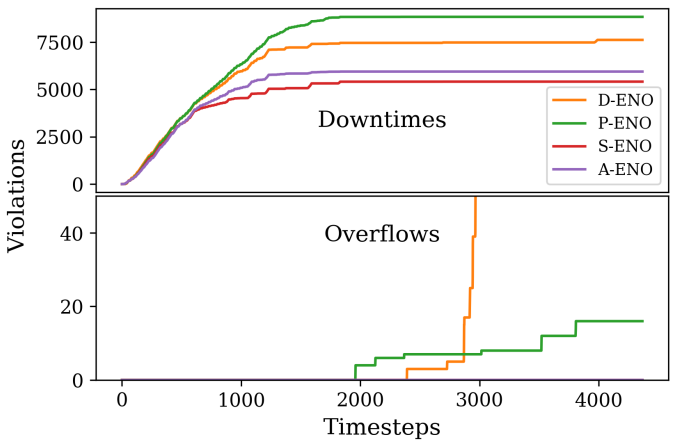

Fig. 7. The figure shows the number of violations for different policies in the first half of the training year. S-ENO has the lowest number of violations followed by A-ENO due to their cyclic preference of exploratory actions. D-ENO has unstable learning illustrated by the sudden increase in overflows around the 3000th timestep.

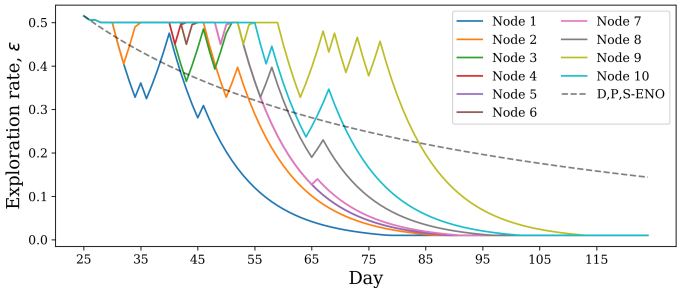

Fig. 8. The figure shows the exploration rates for different nodes of A-ENO during training. ϵ-adapt encourages greedy behavior if rewards are positive. Nodes that start accumulating rewards quickly anneal their ϵ faster. The dashed gray line shows the non-adaptive ϵ-decay for D,P,S-ENO.

the node to act more greedily whereas a negative reward encourages it to explore more. This requires the node to have some minimal amount of exploratory experience before it can start to adapt. Thus in our experiments, A-ENO starts dynamically adapting its exploration rate only after ϵ has annealed to a value below 0.5. Once it starts to adapt ϵ, we limit its value at a maximum of 0.5. We do this so that excessive loss in rewards due to exploration does not cause the node to act in a complete random manner.

Figure 8 shows how ϵ adapts for different nodes of the A-ENO system. The adaptive behavior of the A-ENO system is triggered around the 25th day of training. Node 1 (blue) accumulates rewards faster than its counterparts and quickly anneals its ϵ, i.e., given Node 1's environment, exploration seemed to be redundant. As a result, the node started acting more greedily and accumulating even more rewards.

Node 9, on the other hand, is having difficulty in receiving positive rewards even until the 75th day, probably owing to a difficult environment. Hence, it keeps exploring until the central server learns a good enough policy. In doing so, Node 9 not only increases its rewards but all the other nodes also benefit from its exploratory experiences. By the 115th day, all nodes have started acting mostly greedily. The spikes in their curves correspond to episodes of negative rewards - this may

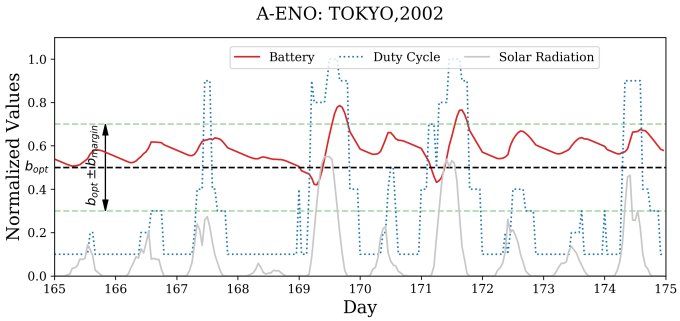

Fig. 9. Intelligent duty cycling policy learned by A-ENO.

have happened due to exploration or a faulty policy.

Dynamic exploration ensures that nodes do not blindly anneal their ϵ on the basis of time. Rather, the performance of the nodes (indicated by rewards) dictates the annealing rates. Consequently, *A-ENO learns efficiently by adjusting ϵ according to the working environment*. We observe that A-ENO achieves perfect ENO 50 times faster than B-ENO and with one third of the learning cost (Fig. 3 and Table III). As mentioned before, this superior performance comes with some increase in risky exploratory behavior.

E. Optimal Duty Cycling for ENO-RL

Figure 9 shows the duty cycling behavior corresponding to a greedy implementation of A-ENO policy for 2018 (during testing). It shows the battery and solar profile as well as the duty cycles for ten consecutive days. We observe that the node adapts its duty cycle to the diurnal variations and the widely different solar energy profiles to maintain ENO. The policy has learned that it is optimal to decrease the duty cycles during nighttime and intelligently increase the duty cycles during the day so that the battery level is within $b_{opt} \pm b_{margin}$. More results are shown in Appendix C (Fig. 11 and 12).

VII. Conclusion

We propose coordinated ϵ-greedy exploration methods to partition the state-space among different agents to explore efficiently during DiRL and accelerate training in ENO-RL systems. To overcome the non-uniform distribution of risk associated with coordinated exploration, we propose methods to trade-off performance and safety. Furthermore, by intelligently adjusting the exploration rates and the preference factors, we decrease instances of unnecessary exploration. Our methods accelerate learning speeds by an order of magnitude and lowers violations during training by upto one thirds. We can conclude that using our methods, it is possible to use a comparatively simple ϵ-greedy exploration to optimize exploration for a large state-space. As a result, EHWSNs can work optimally in an environment that it has never experienced before with less than half a year of training and reduced learning costs.

Acknowledgment

This work was partially supported by JSPS KAKENHI Grant Numbers 18J20946, 17H01708 and Japan Science and Technology Agency (JST) CREST Grant Number JPMJCR1785.

REFERENCES

[1] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 6, no. 4, p. 32, 2007.

[2] S. Shresthamali, M. Kondo, and H. Nakamura, "Adaptive power management in solar energy harvesting sensor node using reinforcement learning," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 5s, p. 181, 2017.

[3] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. Van Hasselt, and D. Silver, "Distributed prioritized experience replay," *arXiv preprint arXiv:1803.00933*, 2018.

[4] A. Ortiz, H. Al-Shatri, X. Li, T. Weber, and A. Klein, "Reinforcement learning for energy harvesting point-to-point communications," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.

[5] P. Blasco, D. Gunduz, and M. Dohler, "A learning theoretic approach to energy harvesting communication system optimization," *IEEE Transactions on Wireless Communications*, vol. 12, no. 4, pp. 1872–1882, 2013.

[6] R. Jia, J. Zhang, X.-Y. Liu, P. Liu, L. Fu, and X. Wang, "Optimal rate control for energy-harvesting systems with random data and energy arrivals," *ACM Transactions on Sensor Networks (TOSN)*, vol. 15, no. 1, p. 13, 2019.

[7] R. C. Hsu, C.-T. Liu, and H.-L. Wang, "A reinforcement learning-based tod provisioning dynamic power management for sustainable operation of energy harvesting wireless sensor node," *IEEE Transactions on Emerging Topics in Computing*, vol. 2, no. 2, pp. 181–191, 2014.

[8] A. Murad, F. A. Kraemer, K. Bach, and G. Taylor, "Autonomous management of energy-harvesting iot nodes using deep reinforcement learning," *arXiv preprint arXiv:1905.04181*, 2019.

[9] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[10] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[11] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International Conference on Machine Learning*, 2015, pp. 1889–1897.

[12] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*, 2016, pp. 1928–1937.

[13] A. Nair, P. Srinivasan, S. Blackwell, C. Alcicek, R. Fearon, A. De Maria, V. Panneershelvam, M. Suleyman, C. Beattie, S. Petersen *et al.*, "Massively parallel methods for deep reinforcement learning," *arXiv preprint arXiv:1507.04296*, 2015.

[14] Y. Xu, H. G. Lee, Y. Tan, Y. Wu, X. Chen, L. Liang, L. Qiao, and D. Liu, "Tumbler: Energy efficient task scheduling for dual-channel solar-powered sensor nodes," in *Proceedings of the 56th Annual Design Automation Conference 2019*, ser. DAC '19. New York, NY, USA: ACM, 2019, pp. 172:1–172:6. [Online]. Available: http://doi.acm.org/10.1145/3316781.3317927

[15] K. Gai and M. Qiu, "Optimal resource allocation using reinforcement learning for iot content-centric services," *Applied Soft Computing*, vol. 70, pp. 12–21, 2018.

[16] Y. Xu, H. G. Lee, X. Chen, B. Peng, D. Liu, and L. Liang, "Puppet: Energy efficient task mapping for storage-less and converter-less solar-powered non-volatile sensor nodes," in *2018 IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 2018, pp. 226–233.

[17] Y. Rioual, Y. Le Moullec, J. Laurent, M. I. Khan, and J.-P. Diguet, "Reward function evaluation in a reinforcement learning approach for energy management," in *2018 16th Biennial Baltic Electronics Conference (BEC)*. IEEE, 2018, pp. 1–4.

[18] K. Geissdoerfer, B. Kusy, R. Jurdak, and M. Zimmerling, "Getting more out of energy-harvesting systems: Energy management under time-varying utility with preact," in *2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 2019, pp. 109–120.

[19] A. E. Braten and F. A. Kraemer, "Towards cognitive iot: Autonomous prediction model selection for solar-powered nodes," in *2018 IEEE International Congress on Internet of Things (ICIOT)*. IEEE, 2018, pp. 118–125.

[20] S. B. Thrun, "E cient exploration in reinforcement learning," Technical Report CMU-CS-92-102, School of Computer Science, Carnegie Mellon , Tech. Rep., 1992.

[21] M. Fortunato, M. G. Azar, B. Piot, J. Menick, I. Osband, A. Graves, V. Mnih, R. Munos, D. Hassabis, O. Pietquin *et al.*, "Noisy networks for exploration," *arXiv preprint arXiv:1706.10295*, 2017.

[22] T. Salimans, J. Ho, X. Chen, S. Sidor, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *arXiv preprint arXiv:1703.03864*, 2017.

[23] S. Khadka and K. Tumer, "Evolution-guided policy gradient in reinforcement learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 1188–1200.

[24] S. Khadka, S. Majumdar, S. Miret, E. Tumer, T. Nassar, Z. Dwiel, Y. Liu, and K. Tumer, "Collaborative evolutionary reinforcement learning," *arXiv preprint arXiv:1905.00976*, 2019.

[25] J. Achiam and S. Sastry, "Surprise-based intrinsic motivation for deep reinforcement learning," *arXiv preprint arXiv:1703.01732*, 2017.

[26] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell, "Curiosity-driven exploration by self-supervised prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 16–17.

[27] I. Sorokin, A. Seleznev, M. Pavlov, A. Fedorov, and A. Ignateva, "Deep attention recurrent q-network," *arXiv preprint arXiv:1512.01693*, 2015.

[28] D. Pathak, D. Gandhi, and A. Gupta, "Self-supervised exploration via disagreement," *arXiv preprint arXiv:1906.04161*, 2019.

[29] R. Houthooft, X. Chen, Y. Duan, J. Schulman, F. De Turck, and P. Abbeel, "Vime: Variational information maximizing exploration," in *Advances in Neural Information Processing Systems*, 2016, pp. 1109–1117.

[30] A. Masadeh, Z. Wang, and A. E. Kamal, "Reinforcement learning exploration algorithms for energy harvesting communications systems," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[31] M. Tokic and G. Palm, "Value-difference based exploration: adaptive control between epsilon-greedy and softmax," in *Annual Conference on Artificial Intelligence*. Springer, 2011, pp. 335–346.

[32] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[33] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint arXiv:1511.06581*, 2015.

[34] TMote, "Tmote sky," 2019, [Online; accessed 6-July-2019]. [Online]. Available: https://insense.cs.st-andrews.ac.uk/files/2013/04/tmote-sky-datasheet.pdf

[35] Japan Meteorological Agency, "Japan meteorological agency," 2019, [Online; accessed 6-July-2019]. [Online]. Available: http://www.jma.go.jp/jma/index.html

[36] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[37] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

APPENDIX A
ENO-RL SPECIFICATIONS

Parameter	Value	Description
T	24	Time steps per episode
b_{max}	10.0 Wh	Maximum battery level
b_{opt}	5.0 Wh	Optimal battery level
b_{margin}	$\pm$3.0 Wh	Maximum deviation from b_{opt}
h_{max}	1.0 Wh	Maximum harvested energy per timestep
z_{avg}	0.5 Wh	Mean node energy consumption per timestep
d_{min}	10%	Minimum duty cycle
d_{max}	100%	Maximum duty cycle
N_D	10	No. of duty cycles
N_F	10	No. of weather forecast levels

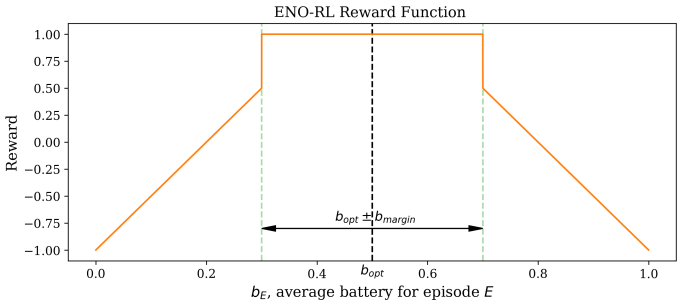

Fig. 10. The reward for episode E depends upon the mean battery level b_E during that episode. The rewards are clipped at ± 1 to ensure stability during training.

APPENDIX B
NEURAL NETWORK SPECIFICATION

TABLE IV
DQN HYPERPARAMETERS

Architecture	Double DQN with Dueling Networks [9], [32], [33]

Hyperparameters	**Single Agent RL**	**Distributed RL**
hidden layers	1	
hidden layer width	50	
activation	ReLu	
initialization	Kaiming, Xavier	
minibatch size	32	
replay memory size	12,096	
target update frequency, N_u	241,920	
discount factor	0.999	
loss function	mean squared error	
optimizer	ADAM	
no. of learners	1	
no. of workers	1	10
learning rate	0.001	
learning steps per day	24	1000

[1] The values of the hyperparameters were obtained empirically using the ENO-RL environment based on weather data for Tokyo, 2000.

[2] The fully connected layer was initialized using Kaiming Uniform method [36] and the value/advantage layers were initialized using Xavier Uniform method [37]. For added stability, the inputs to the neural network are standardized to have a zero mean and a standard deviation of one.

The ENO-RL agent maintains two copies of DQN at each iteration: $Q(s, a; \theta)$ and $Q(s, a; \theta^-)$ for stable learning [9]. θ^- are the parameters of a separate fixed (frozen) target network. While θ is updated at every learning step, θ^- is updated with the values of θ after every N_u learning steps. The agent stores its experiences $e_t = (s_t, a_t, r_t, s_{t+1})$ at each timestep t in a memory pool $\mathcal{M} = \{e_1, e_2 .... e_t\}$. During learning, random minibatches of experiences $(s, a, r, \hat{s}) \sim \mathcal{Z}(\mathcal{M})$ are selected and the Q-learning update is performed using the following loss function ($\hat{s}$ is the state following s).

$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,\hat{s})\sim\mathcal{Z}(\mathcal{M})}\left[\left(y_i^{DDDQN} - Q(s, a; \theta_i)\right)^2\right] \quad (9)$$

where $y_i^{DDDQN} = r + \gamma Q(\hat{s}, \arg\max_{\hat{a}} Q(\hat{s}, \hat{a}; \theta_i); \theta_i^-)$.

APPENDIX C
ENO-RL RESULTS

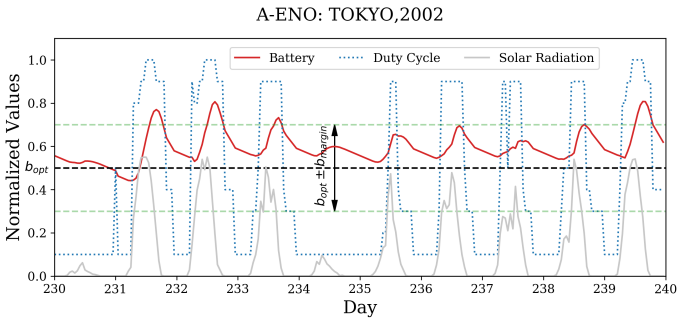

Fig. 11. Battery profile of ENO-RL using A-ENO for Tokyo, 2002 from 230th to 240th day. The battery fluctuates around the 50% mark and hence the node is ENO. The nodes has high duty cycles during sunny periods and low duty cycles during the night and day with low sunshine. Note that the duty cycles never dip below 0.1%. On the 234th and 23rd day, the solar energy is scarce and the APM intelligently lowers the duty cycle.

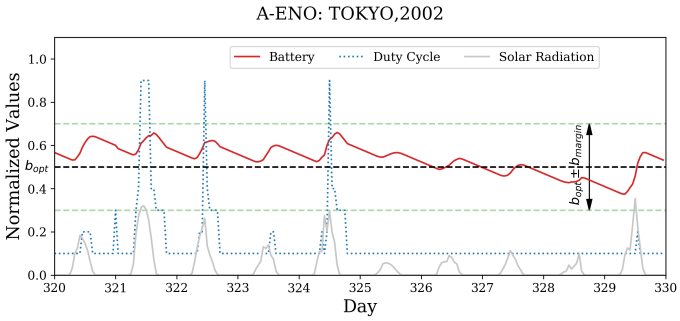

Fig. 12. Battery profile of ENO-RL using A-ENO for Tokyo, 2002 from 320th to 330th day. The APM successfully negotiates consecutive days of low solar energy without any battery violations while maximizing the duty cycle when possible.