Toward Physics-Guided Safe Deep Reinforcement Learning for Green Data Center Cooling Control

Ruihang Wang
ruihang001@ntu.edu.sg
Nanyang Technological University
Singapore

Xinyi Zhang
zh0031yi@ntu.edu.sg
Nanyang Technological University
Singapore

Xin Zhou
zhouxin@ntu.edu.sg
Nanyang Technological University
Singapore

Yonggang Wen
ygwen@ntu.edu.sg
Nanyang Technological University
Singapore

Rui Tan
tanrui@ntu.edu.sg
Nanyang Technological University
Singapore

ABSTRACT

Deep reinforcement learning (DRL) has shown good performance in tackling Markov decision process (MDP) problems. As DRL optimizes a long-term reward, it is a promising approach to improving the energy efficiency of data center cooling. However, enforcement of thermal safety constraint during DRL's state exploration is a main challenge. The widely adopted reward shaping approach adds negative reward when the exploratory action results in unsafety. Thus, it needs to experience sufficient unsafe states before it learns how to prevent unsafety. In this paper, we propose a safety-aware DRL framework for single-hall data center cooling control. It applies offline imitation learning and online post-hoc rectification to holistically prevent thermal unsafety during online DRL. In particular, the post-hoc rectification searches for the minimum modification to the DRL-recommended action such that the rectified action will not result in unsafety. The rectification is designed based on a thermal state transition model that is fitted using historical safe operation traces and able to extrapolate the transitions to unsafe states explored by DRL. Extensive evaluation for chilled water and direct expansion cooled data centers in two climate conditions shows that our approach saves 22.7% to 26.6% total data center power compared with conventional control, reduces safety violations by 94.5% to 99% compared with reward shaping.

KEYWORDS

Data center, safe reinforcement learning, energy efficiency, thermal safety

1 INTRODUCTION

Data centers (DCs) form the computing backbone of Internet. The DC market has ever been growing to meet the cloud computing demands. With the current compound annual growth rate of 13.4% [3], the global DC market is projected to be doubled in about 5 years. DCs are energy-intensive. From a survey in 2016, the DC industry uses 2% of the world's electricity production [26]. Given the fast DC market increase, it is important to improve DC energy efficiency in the pursuit of carbon neutrality. A DC is a cyber-physical system consisting of the information technology (IT) system and the cooling system. The IT equipment uses electricity for computing and generates heat that needs to be moved and dissipated to the ambient. This moving process, i.e., cooling, uses more than 40% of DC's electricity supply [26]. Therefore, perpendicular to the design and adoption of new energy-efficient IT equipment, proper control of the cooling system based on distributed sensing and cyber intelligence is critical to improving DC energy efficiency.

In this paper, we consider the problem of DC cooling control that aims at reducing the DC energy usage subject to the IT equipment's thermal safety constraint. Any IT device specifies the highest temperature that it can tolerate (e.g., 32°C for ASHRAE Class A1 servers [5]). Crossing the temperature upper limit may cause device shutdown and service disruption. Many DC operators adopt an operation scheme of maintaining the temperature in the hot zone of the data hall (referred to as zone temperature) at a certain setpoint that is sufficiently lower than the IT equipment's temperature upper limits. In the presence of dynamic IT workload, the operating point of the computer room air conditioning (CRAC) units, i.e., the temperature and mass flow rate of the cold supply air, need to be periodically adjusted to maintain the zone temperature. This can be achieved by conventional feedback controls [30].

The DC cooling control can be also viewed as a Markov decision process (MDP). Deep reinforcement learning (DRL) has shown good performance in tackling various MDP problems [27]. Recent studies [7, 12] have also applied DRL to learn the energy-efficient policies for operating the heating, ventilation, and air conditioning (HVAC) systems of human-centric buildings. The learning process is steered by a reward function that jointly captures the cumulative penalty of process deviations from the setpoint and the long-term average energy efficiency of the HVAC system. Thus, compared with the conventional feedback controls that only focus on maintaining the temperature at the setpoint, DRL additionally admits the goal of energy efficiency optimization. The existing results show that the adequately trained DRL agents achieve up to 16.7% HVAC energy savings over long runs [7]. Such energy efficiency gains achieved for HVAC control motivate us to develop DRL for DC cooling control. However, DC cooling control faces more dynamics in heat load and more stringent requirement on the thermal safety.

In online DRL (including the on-policy and off-policy schemes), the agent interacts with the controlled system iteratively and learns from positive and negative rewards caused by the performed actions. For an intricate MDP problem, the convergence of the DRL often requires experiencing a large number of action-state trials. For instance, the DRL for HVAC control in [12] performs 500,000 interactions. To apply DRL for DC, it is critical to avoid the data hall's excursions to thermal unsafety during the learning process, forming a constrained MDP (CMDP) problem. To tackle CMDP in the general context, recent studies (e.g., [17, 25]) adopt a *reward shaping* approach that applies a penalty in the reward function when the constraint is violated. However, this approach, which is essentially a Lagrangian relaxation [14], does not explicitly enforce the constraint. *Post-hoc rectification* is another approach that explicitly addresses the constraint of CMDP. Specifically, in each control period, the approach finds the smallest rectification to the control action suggested by the DRL agent such that the rectified action will not drive the system to the unsafe region. The studies [8, 11] have derived the closed-form rectifications when the controlled system follows linear state transition. However, the thermal state transition in DC is nonlinear. The solutions [8, 11] based on the linear approximation of the thermal state transition will inevitably lead to degradation of thermal safety compliance.

In this paper, we propose a safety-aware reinforcement learning framework (Safari) for single-hall DC cooling control. The single-hall scheme is often adopted in enterprise DCs. Safari takes a holistic design that enables the adoption of DRL to pursue DC energy savings while effectively preventing excursions to thermal unsafety. Safari comprises an offline stage and an online stage. First, Safari adopts offline imitation learning to initialize the DRL agent. The imitation learning is based on the historical traces when the CRAC is operated by the conventional controller that empirically assures thermal safety. Such data traces are in general available in the DC infrastructure management (DCIM) system. The imitation learning can reduce the DRL agent's unsafe attempts in the online stage. Second, for the online stage, we design a new post-hoc rectification approach based on a state transition model that captures the data hall thermodynamics. The model fitted with historical traces generated by the safe conventional controller can accurately extrapolate the state transitions that are unseen in the historical traces and explored by the DRL agent. Thus, a salient advantage of Safari lies in the low overhead and low demand on data (i.e., only safe data are needed) when fitting the state transition model. In contrast, as shown in this paper, the domain-agnostic approach of using a neural network to model the state transitions requires unsafe training data, which is in general unavailable and contradictory to the original goal of ensuring safety.

The contributions of this paper are summarized as follows.

- We formulate DC cooling control as an MDP problem and design a DRL agent. Then, we conduct extensive measurements using the EnergyPlus simulator [10] to show the DC energy savings achieved by the DRL agent. The study also shows that the agent designed without rigorous thermal safety consideration produces excessive unsafe events, even when the temperature setpoint is conservatively low.
- We design Safari that applies imitation learning and post-hoc rectification to holistically prevent thermal unsafety during online DRL. We develop a DC-specific post-hoc rectification approach that exploits thermodynamic laws and outperforms the existing domain-agnostic rectification approaches.
- We conduct extensive simulations for two DCs with chilled water and direct expansion cooling systems in two climate conditions. When IT workload pattern is simple, Safari saves 22.7% to 26.6% power compared with conventional control, reduces safety violations by 94.5% to 99% compared with reward shaping. When IT workload pattern is complex, the power saving and violation reduction are 25.7% and 99%, respectively.

Paper organization: §2 reviews related work. §3 presents the background and preliminaries. §4 presents a measurement study. §5 presents the design of Safari. §6 presents evaluation results. §7 discusses several relevant issues. §8 concludes this paper.

2 RELATED WORK

This section reviews the existing studies on machine learning (ML)-based DC cooling control and safe reinforcement learning. Table 1 categorizes the existing approaches, summarizes their requirements and implementation properties for safety consideration. In what follows, we discuss the details of these existing studies.

■ **ML-based DC cooling control.** DC cooling control is a CMDP problem. The existing ML-based solutions can be categorized into *model-free* [9, 17, 25, 28] and *model-based* [16, 31] approaches.

The model-free approaches learn the control policy by directly interacting with the controlled system, which in general follow online DRL. The study [17] applies the deep deterministic policy gradient (DDPG) to learn the cooling control policy for a two-zone DC. The studies [25] and [9] adopt the parameterized deep Q-network (DQN) and the proximal policy optimization (PPO), respectively, to learn the policy for joint control of cooling and IT (e.g., via compute job allocation). The study [28] applies DQN to learn the policy for air free cooling control. After adequate learning, the DRL agents in [9, 17, 25, 28] achieve energy savings. During the learning phase, they all follow the reward shaping strategy to relax the constrained optimization problem to an unconstrained one. Thus, they only address the thermal safety constraint in a *semi-explicit* manner. Differently, our proposed approach directly and explicitly addresses the thermal safety constraint via post-hoc rectification. As indicated in Table 1, the reward shaping approach needs *exploratory data* that cover the unsafe region to learn from the penalty in a *reactive* manner. Therefore, the learning phase of reward shaping in general experiences unsafe states.

The model-based approaches (e.g., [15, 16, 31]) aim at reducing the *sampling complexity* (i.e., the number of interactions with the controlled system) by allowing the ML-based controller to interact with a computational model of the system. The study [16] presents the model-predictive control (MPC) of DC cooling based on a linearized thermodynamic model. However, the MPC formulation does not explicitly address the thermal constraint. The study [15] also applies MPC and uses a Gaussian process model for the state transition. It continuously updates the model with online data that are sampled by following an optimal experiment design strategy. The study [31] constructs a deep neural network (DNN) to capture the thermodynamics and uses it to reduce the sampling complexity of a DRL agent designed with reward shaping. However, the training of the DNN requires a large amount of exploratory data.

■ **Safe reinforcement learning.** Various safe reinforcement learning techniques have been proposed to address the CMDP problem under the general context, which can be categorized into the simplex, reward shaping, and post-hoc rectification approaches.

Table 1: Categorization and summary of the existing studies relevant to ML-based DC cooling control.

Policy learning category	Approach	Studies	Applications	Requirements for safety		Safety implementation	
				Exploratory data	Transition model	When?	Explicitness
Model-free*	DRL (simplex)	[19, 24]	Load balancing, etc.	Required	Not required	Reactive	Explicit
	DRL (reward shaping)	[9, 17, 25, 28]	DC cooling, etc.				Semi-explicit
	DRL (post-hoc rectification)	[8, 11]	HVAC control, etc.	(Not) required†	Linear model	Proactive	Explicit
		Safari	DC cooling control	Not required	Thermodynamics		
Model-based*	DRL (reward shaping)	[31]	DC cooling control	Required	Neural network	Reactive	Semi-explicit
	MPC	[16]		Required	Linear model	–	Implicit
		[15]	Building	Required	Gaussian process	–	Implicit

*The "model" refers to that needed/used for learning the control policy toward the optimization objective, not for safety consideration. The three categories of model-free approaches are used as baselines for comparison when evaluating Safari in §6.

†If the state transition is linear, [8, 11] do not require exploratory data. However, for nonlinear DC thermodynamics, although [8, 11] can be extended to use DNNs capturing nonlinear transitions, our experiments in §5.3 shows that exploratory data will be needed to train the DNNs.

As the reward shaping approach has been reviewed earlier in the context of DC cooling control, we will focus on the remaining two. The studies [19, 24] follow the simplex architecture that executes the DRL as the high-performance learner to maximize the reward and falls back to a safe controller once the system enters the unsafe region. For each fallback, the simplex approach requires and reacts to at least one unsafe state. Although the use of the safe controller renders the safety implementation explicit, the frequent interruptions to the DRL may adversely affect its learning efficiency.

The post-hoc rectification approach searches for the minimum modification to the control action generated by an ML-based controller to *proactively* prevent the system from entering the unsafe region. In [11], based on a linear state transition model, a closed-form rectification is found by solving a convex constrained optimization problem. The work [8] extends the above approach by augmenting the DRL policy network with a projection layer that projects the action onto a predefined safety set and applies the extended approach to HVAC and power grid inverter control. However, the effectiveness of the approaches in [8, 11] depends on the linearity of the controlled system. In this paper, we will analytically show the nonlinear property of the thermodynamics in DC. This paper further advances the post-hoc rectification approach by accommodating a nonlinear model governing the DC thermodynamics to enforce thermal safety. The model can be fitted with historical non-exploratory data produced under the control of a safe controller. As the fitted model remains accurate in the unsafe region, our approach does not require the undesirable exploratory data.

3 PRELIMINARIES

This section presents the preliminaries of DC cooling and DRL. The symbols used in this paper are summarized in Appendix A.

3.1 DC Cooling Control Model

This paper considers both chilled water (CW) and direct expansion (DX) cooling systems. Fig. 1 illustrates a typical CW-cooled DC consisting of a cooling tower, a chiller, two water pumps (i.e., chilled water pump and condensed water pump), and a data hall hosting multiple CRAC units and many servers. This paper focuses on the single-hall scheme, which is often adopted in enterprise DCs. In §7, we will discuss how to extend Safari to address the multi-hall scheme. The heat generated by the IT equipment is moved out of the DC via three cycles. In the *indoor air cycle*, the CRAC units supply cold air to the data hall cold aisle, draws hot air from the zone, and cools the hot air by their internal air-water heat exchangers. In the *chilled water cycle*, the chilled water pump supplies chilled water to the CRAC units. The return warm water from CRAC is cooled by the chiller via a vapor-compression refrigeration process. In the *condenser water cycle*, the chiller transfers heat to the cooling tower by the condenser. The cooling tower dissipates the heat to the outdoor environment. The total power usage of the cooling system, denoted by P_c, comprises the power usages of the CRAC units, the chiller, the cooling tower, and the water pumps. A component's power usage depends on its working status. The EnergyPlus simulator contains realistic power usage models of the cooling components. The IT power usage (denoted by P_{IT}) comprises the powers used by computing and the IT equipment's internal fans, where the former mainly depends on the utilization of the IT equipment (denoted by U_{IT}) and the latter mainly depends on the data hall's cold aisle temperature (denoted by T_{in}). Therefore, we model $P_{\text{IT}} = p(U_{\text{IT}}, T_{\text{in}})$. In the simulations conducted in this paper, we configure the EnergyPlus to use a model $p(U_{\text{IT}}, T_{\text{in}})$ from

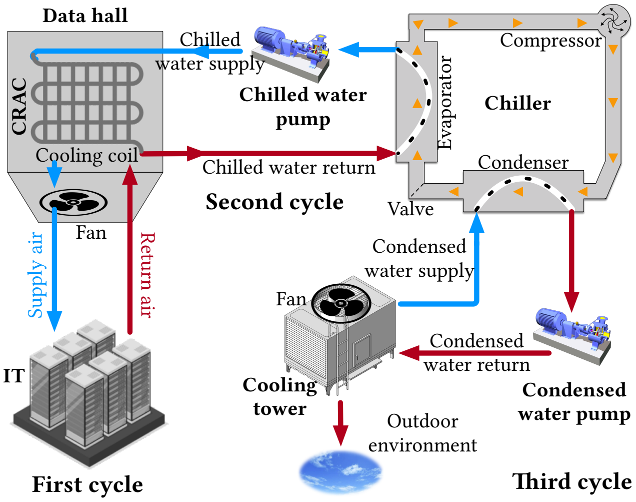

Fig. 1: Typical chilled water cooled DC system.

[21]. As the design of Safari does not require the power usage models discussed above, we omit introducing their details. Compared with CW, the DX cooling system is simpler – it consists of two cycles only. Appendix B provides a brief introduction of DX. Note that Safari is agnostic to the type of cooling system. In §6, we will evaluate the performance of Safari for both CW and DX cooling.

Then, we describe the heat process in the data hall. We consider a scenario where 1) the CRAC units adopt the same setpoint for the supply air temperature and 2) the zone temperature has a uniform spatial distribution. The zone temperature, denoted by T_{z}, is governed by the following thermodynamic model derived from the law of the conservation of energy [4]:

$$\frac{\mathrm{d}T_{\mathrm{z}}(t)}{\mathrm{d}t} = \frac{f(t)}{\rho V_{\mathrm{s}}}\left(T_{\mathrm{in}}(t) - T_{\mathrm{z}}(t)\right) + \frac{1}{\alpha V_{\mathrm{s}}} Q(t), \qquad (1)$$

where t is time, $f(t)$ is the instantaneous total mass flow rate of the supply air from all CRAC units, ρ is the density of air, V_{s} is the data hall volume, α is a system dependent parameter that is relevant to the thermal capacitance of air, and $Q(t)$ is the instantaneous sensible heat load. In practice, Q comprises the portion of P_{IT} converted to heat, the heats emitted from lighting and human workers temporarily in the data hall, and the external heat transferred into the data hall via walls. As the IT-generated heat usually dominates $Q(t)$, to simplify the discussion in this paper, we assume $Q(t) = P_{\mathrm{IT}}(t)$. Note that in the EnergyPlus simulations conducted in this paper, we account for lighting heat. To achieve the uniform spatial distribution of the zone temperature, thermal-aware load balancing [20] can be applied. In addition, the total mass flow rate $f(t)$ can be attributed to the CRAC units properly to help equalize the IT racks' outlet temperatures. In this paper, we will not detail the zone temperature equalization. Instead, we focus on the main challenge of improving DC energy efficiency while maintaining the overall thermal safety in the hot zone.

As discussed in §1, to maintain $T_{\mathrm{z}}(t)$ at a setpoint, the DC cooling control periodically adjusts the setpoints for $f(t)$ and $T_{\mathrm{in}}(t)$. Let τ denote the control period. A typical setting for τ is 15 minutes [31]. Let $\hat{f}[k]$ and $\hat{T}_{\mathrm{in}}[k]$ denote the setpoints applied at $t = k\tau$ for the kth control period of $t \in (k\tau, (k+1)\tau)$. The cooling system implements $\hat{f}[k]$ and $\hat{T}_{\mathrm{in}}[k]$ via the primary controls of its components. Due to the uncertain evolution of $P_{\mathrm{IT}}(t)$, the cooling process is a continuous-time stochastic process. To make the analysis tractable, we make the following simplifying assumptions, while the simplified model still captures the main challenges of DC cooling control. Note that these assumptions will be relaxed in the performance evaluation.

Assumption 1. $P_{\mathrm{IT}}(t)$ only changes at the start of each control period and $P_{\mathrm{IT}}[k] \triangleq P_{\mathrm{IT}}(t)|_{t\in((k-1)\tau,k\tau)}$ is Markovian.

Assumption 2. At the end of each control period, the DC system has converged to a steady state and the cooling components' primary controls have zero steady-state control errors.

Assumption 1 follows from the time-slotted treatment that has been widely adopted to convert a continuous-time problem to its discrete-time counterpart [22]. Under Assumption 2, the setpoints $\hat{f}[k-1]$ and $\hat{T}_{\mathrm{in}}[k-1]$ are implemented when $t \to k\tau^{-}$. Formally, $f(t)|_{t\to k\tau^{-}} = \hat{f}[k-1]$, $T_{\mathrm{in}}(t)|_{t\to k\tau^{-}} = \hat{T}_{\mathrm{in}}[k-1]$, $\left.\frac{\mathrm{d}T_{\mathrm{z}}(t)}{\mathrm{d}t}\right|_{t\to k\tau^{-}} = 0$. By substituting the above simplification-induced results into Eq. (1) and by defining $T_{\mathrm{z}}[k] \triangleq T_{\mathrm{z}}(t)|_{t\to k\tau^{-}}$, we obtain the following steady state transition model:

$$T_{\mathrm{z}}[k] = \hat{T}_{\mathrm{in}}[k-1] + \frac{\rho P_{\mathrm{IT}}[k]}{\alpha \hat{f}[k-1]}. \qquad (2)$$

3.2 Deep Reinforcement Learning

DRL is a deep learning-based approach that learns a policy function $\mu_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$ to tackle an MDP problem. The DRL agent uses the policy to select the action $\mu[k]$ based on the current system state $s[k]$, i.e., $\mu[k] = \mu_{\boldsymbol{\theta}}(s[k])$. The action drives the system to the next state $s[k+1]$, while the agent receives an immediate reward $r[k]$. Let γ denote a discounted factor. The agent uses an algorithm to learn the optimal policy $\boldsymbol{\theta}^{*}$ for the following unconstrained optimization problem: $\boldsymbol{\theta}^{*} = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{s}\left[\sum_{k=0}^{\infty} \gamma^{k} r[k] \middle| \mu_{\boldsymbol{\theta}}\right]$.

In this paper, we use the DDPG [18] learning algorithm to deal with the continuous action space in DC cooling control. It concurrently learns $\mu_{\boldsymbol{\theta}}(s)$ and a Q-function $Q_{\boldsymbol{\phi}}(s,\mu)$ parameterized with parameters $\boldsymbol{\phi}$ and differentiable with respect to action μ. To learn the Q-function, the agent samples a batch of N transition data samples $\{s_i, \mu_i, s_{i+1}, r_i | i = 1, \ldots, N\}$ through interacting with the controlled system. Then, it updates $\boldsymbol{\phi}$ by minimizing the loss function $\mathcal{L}(\boldsymbol{\phi}) = \frac{1}{N}\sum_{i=1}^{N}(Q_{\boldsymbol{\phi}}(s_i,\mu_i) - y_i)^2$, where y_i is the target Q value given by $y_i = r_i + \gamma Q'_{\boldsymbol{\phi}}(s_{i+1}, \mu'_{\boldsymbol{\theta}}(s_{i+1}))$. The $Q'_{\boldsymbol{\phi}}$ and $\mu'_{\boldsymbol{\theta}}$ are two target networks copied from the original networks and updated once per main network update. To learn the policy function, it updates $\boldsymbol{\theta}$ by maximizing $\mathcal{J}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} Q_{\boldsymbol{\phi}}(s_i, \mu_{\boldsymbol{\theta}}(s_i))$.

4 PERFORMANCE OF REWARD SHAPING

This section formulates the DC cooling control as an MDP problem with reward shaping for thermal safety consideration. Then, we measure a DDPG solution's energy savings with respect to a conventional controller and its effectiveness in preventing thermal unsafety. The results motivate the pursuit of better solutions in §5.

4.1 MDP Formulation with Reward Shaping

IT workload and outdoor environment condition are two exogenous factors to DC cooling control. Let $T_{\mathrm{o}}[k]$ denote the outdoor air temperature at $t = k\tau$. We assume both $P_{\mathrm{IT}}[k]$ and $T_{\mathrm{o}}[k]$ are Markovian. The data hall zone temperature T_{z} should be kept within a thermal safety upper bound denoted by $\overline{T}_{\mathrm{z}}$, i.e., $T_{\mathrm{z}}[k] \le \overline{T}_{\mathrm{z}}, \forall k$. In the simulations conducted in this paper, we set $\overline{T}_{\mathrm{z}} = 32°\mathrm{C}$, which is the temperature upper limit for ASHRAE A1 servers [5].

We define the action, state, and reward of the MDP formulation with reward shaping as follows.

Action: The action applied in the kth control period, denoted by $\boldsymbol{\mu}[k]$, consists of the setpoints of the CRAC's supply air temperature and mass flow rate, i.e., $\boldsymbol{\mu}[k] = (\hat{T}_{\mathrm{in}}[k], \hat{f}[k])$.

State: Besides the notation defined in §3.1, we also define $T_{\mathrm{in}}[k] \triangleq T_{\mathrm{in}}(t)|_{t\to k\tau^{-}}$ and $P_{\mathrm{c}}[k] \triangleq P_{\mathrm{c}}(t)|_{t\to k\tau^{-}}$. The state $\mathbf{s}[k]$ is defined as $\mathbf{s}[k] \triangleq (T_{\mathrm{z}}[k], T_{\mathrm{in}}[k], P_{\mathrm{c}}[k], P_{\mathrm{IT}}[k], T_{\mathrm{o}}[k])$. When the action $\boldsymbol{\mu}[k]$ is to be chosen at $t = k\tau$, $\mathbf{s}[k]$ is fully observable. From Eq. (2) and the assumption that the two exogenous state components $P_{\mathrm{IT}}[k]$ and $T_{\mathrm{o}}[k]$ are Markovian, the probability distribution of the transition from $\mathbf{s}[k]$ to $\mathbf{s}[k+1]$ under an action $\boldsymbol{\mu}[k]$ is conditioned

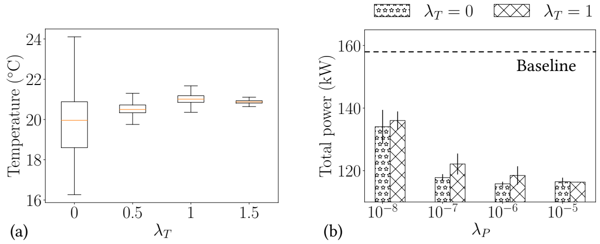

Fig. 2: Impact of λ_T and λ_P on performance of DDPG. (a) Data hall zone temperature; (b) DC total power, with error bar represents standard deviation over multiple DDPG agents.

on the probability distributions of $\mathbf{s}[k]$ and $\boldsymbol{\mu}[k]$ only. Thus, the control process is an MDP.

Reward: According to [13], a good DC cooling controller should maintain the data hall air temperature at a certain setpoint denoted by T_C and reduce energy usage of the whole DC. We adopt the following reward function that incorporates the above two goals and also includes a shaping term for thermal safety consideration:

$$\begin{aligned} r[k] = & -\lambda_T \exp\left(-\lambda_1 (T_z[k] - T_C)^2\right) - \lambda_P P_{DC}[k] \quad \cdots\cdots \textit{goals} \\ & -\lambda_2 \left((T_z[k] - T_U)^+ + (T_L - T_z[k])^+\right), \qquad \cdots \textit{shaping} \end{aligned} \tag{3}$$

where λ_1, λ_2, λ_T and λ_P are several hyperparameters, $P_{DC}[k]$ is the DC's total power usage (i.e., $P_{DC}[k] = P_{IT}[k] + P_c[k]$), $[T_L, T_U]$ specifies a desirable range for $T_z[k]$, $(x)^+ = \max\{0, x\}$. The shaping term adds a penalty when T_z is out of $[T_L, T_U]$. The T_U can be set lower than $\overline{T}_z$ to better address the thermal safety consideration. The objective of the MDP problem is to find the policy parameters to maximize the long-term accumulative reward, i.e., $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{P_{IT}, T_o} \left[\sum_{k=0}^{\infty} \gamma^k r[k] \middle| \boldsymbol{\mu}_{\boldsymbol{\theta}} \right]$, where P_{IT} and T_o are two stochastic processes.

4.2 Performance Measurements

We conduct a set of simulations to evaluate the performance of the DDPG solution. We implement DDPG in PyTorch [23] and integrate the EnergyPlus 8.8.0 simulator with the OpenAI gym [6] interface. Thus, the DDPG agent can learn the control policy for a CW-cooled DC simulated by EnergyPlus. The control period τ is 15 minutes. Other hyperparameter settings of the DDPG can be found in Table 4 of Appendix C. To drive the simulations, we use the historical weather trace of Singapore, which is provided by EnergyPlus. We adopt a simple IT utilization variation pattern for each simulated day: $U_{IT} = 0.5$ from 00:00 to 06:00; $U_{IT} = 0.75$ from 06:00 to 08:00; $U_{IT} = 1.0$ from 08:00 to 18:00; $U_{IT} = 0.8$ from 18:00 to 24:00. We set the first 50 days as the learning phase. After that, we disable the policy update and the system enters a 1-year testing phase. We compare the testing-phase performance of DDPG with an EnergyPlus' built-in controller [2] (referred to as *baseline controller*) that only aims at maintaining $T_z[k]$ at T_C.

4.2.1 Impact of λ_T and λ_P. In Eq. (3), the hyperparameters λ_T and λ_P are the weights for combining the goals of maintaining temperature and reducing total power usage. We fix the other hyperparameters (i.e., λ_1=0.5, λ_2=0.1, T_C=21°C, $T_L = T_C - 1.5$°C, $T_U = T_C + 1.5$°C) and vary λ_T and λ_P. Fig. 2(a) shows the distribution of T_z versus λ_T when $\lambda_P = 10^{-5}$. We train a separate DDPG agent for each λ_T setting. Each error bar shows the distribution of T_z during testing. When $\lambda_T \neq 0$, the T_z fluctuates around T_C and the variation of T_z decreases with λ_T. When $\lambda_T = 0$, T_z has large variations. Next, we fix λ_T to a certain setting and vary λ_P. For each λ_P, we train multiple DDPG agents. For each agent, we obtain the average P_{DC} during testing. Each error bar in Fig. 2(b) shows the standard deviation of the average P_{DC} over the multiple agents. The DC power usage shows a decreasing trend when λ_P increases. In addition, under the same setting for λ_P, the setting of $\lambda_T = 0$ leads to lower P_{DC} compared with the setting $\lambda_T = 1$. This is because the DDPG agent with $\lambda_T = 0$ can focus on reducing P_{DC}. The horizontal dash line in Fig. 2(b) shows the average P_{DC} during testing when the baseline controller is used. We can see that the DDPG controllers bring DC power savings. The results in Fig. 2 show that λ_T and λ_P affect the trade-off between data hall temperature stability and DC power efficiency. In the rest of this section, we set $\lambda_T = 1$ and $\lambda_P = 10^{-5}$.

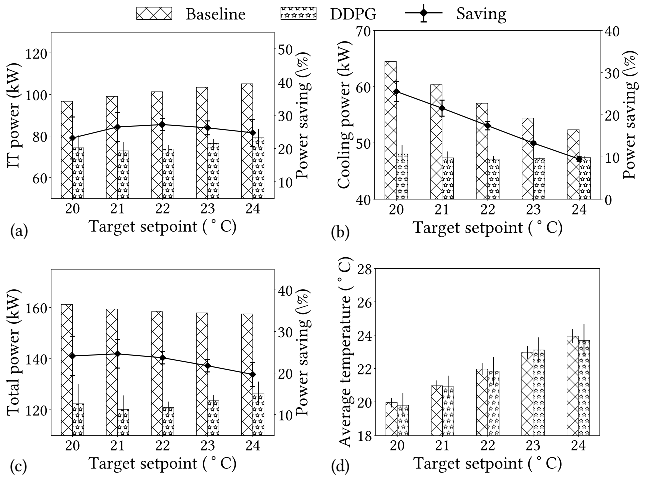

Fig. 3: Comparison between EnergyPlus' built-in controller (baseline) and converged DDPG over 1-year testing. (a)-(c) IT, cooling and total power consumption; (d) average zone air temperature.

4.2.2 Comparison of DDPG and baseline controllers under various T_C settings. Zone temperature setpoint is an important operation setting. We vary T_C from 20°C to 24°C with a step size of 1°C. For each setpoint, we train multiple DDPG agents and measure the averages of P_{IT}, P_c, and P_{DC} during testing for each of the agents. Figs. 3(a)-(c) show the power measurements versus T_C. The error bar shows the standard deviation over the multiple agents. The figures also show the power measurements when the baseline controller is adopted, as well as the relative savings achieved by DDPG. We can see that with the baseline controller, the IT power increases with T_C. With DDPG, the IT power also shows a slight increasing trend. However, DDPG saves more than 20% IT power. Although both controllers maintain T_z at the setpoint with small deviations as shown in Fig. 3(d), our investigation shows that, compared with the baseline controller, DDPG recommends lower $\hat{T}_{in}$ and $\hat{f}$ such that the T_{in} can be maintained lower, according to Eq. (2). As such, the IT power is lower since the server fans rotate slower.

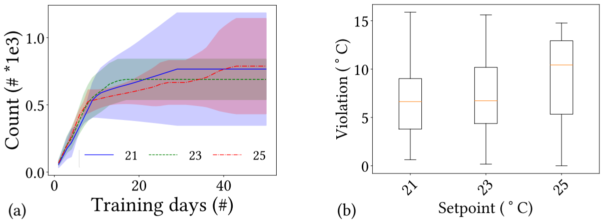

Fig. 4: DDPG's training phase under various temperature setpoints. (a) Cumulative count of safety violations; (b) violation magnitude: mid line, box, and whisker represent median, interquartile range, and degree of dispersion.

From Fig. 3(b), the cooling power decreases with T_C under the baseline controller. A key reason is that, with hotter return air, the temperature difference between the hot air and the chilled water in the CRAC is larger, which allows the CRAC fan to rotate slower while exchanging the same amount of heat. Differently, for DDPG, the cooling power changes slightly when T_C increases. This is because the optimized system under DDPG control has almost hit the minimum cooling power needed to move a certain amount of heat generated by the IT equipment. Fig. 3(c) shows the sum of the results in Figs. 3(a) and (b). Compared with the baseline controller, the DDPG agent can save 20% to 25% total power. In particular, when T_C is 21°C, the relative saving achieves the peak. Note that 21°C is one of the typical zone temperature setpoints in DCs [29].

The above results show that, under a certain T_C setting, the DDPG agent achieves substantial power savings compared with the baseline controller. In addition, under the conventional control that maintains T_z at T_C, running hotter data center (i.e., by setting higher T_C) can be beneficial to energy efficiency [13], due primarily to the saving in cooling power. However, from Fig. 3(c), under the DDPG control, this understanding may not be true, since the proposed DDPG agent jointly considers the impacts of $\hat{T}_{in}$ and $\hat{f}$ on the IT/cooling power and minimizes the DC total power.

4.2.3 *Thermal safety compliance of DDPG.*

We evaluate the thermal safety compliance in terms of the cumulative count and magnitude of the violations to the constraint $T_z[k] \le \overline{T}_z$. Specifically, in the kth control period, the cumulative count is $\sum_{i=0}^{k} H(T_z[i] - \overline{T}_z)$, where $H(\cdot)$ is the unit step function; the violation magnitude is $(T_z - \overline{T}_z)^+$. For each temperature setpoint T_C, we conduct multiple independent experiments and record the two metrics over time. In Fig. 4(a), a curve shows the average of the cumulative counts produced by multiple DDPG agents under a certain T_C setting in the learning phase; the shaded area in the same color shows the corresponding standard deviation. We can see significant increases of the cumulative violation counts up to more than 1,000. In particular, in the first 10 days, there are sharp increases. Fig. 4(b) shows the box plots of the violation magnitude in the 50 days under three settings of T_C. We can see that the violation magnitude can be more than 15°C even when T_C is 21°C, which is 11°C lower than $\overline{T}_z$. These results show that DDPG with reward shaping generates excessive, serious safety violations. In addition, simply adjusting the temperature setpoint T_C does not solve the problem.

5 THE SAFARI APPROACH

5.1 CMDP Formulation & Approach Overview

From the results in §4.2, DDPG achieves energy savings. However, as reward shaping addresses the thermal constraint implicitly, it is weak in preventing thermal unsafety. The excessive, serious safety violations during the learning phase will impede the adoption of DRL for DC. In this paper, we aim at explicitly enforcing the thermal safety constraint of the following CMDP problem:

$$\begin{aligned} &\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{P_{\mathrm{IT}},T_{\mathrm{o}}}\left[\sum_{k=0}^{\infty} \gamma^k r[k] \middle| \boldsymbol{\mu}_{\boldsymbol{\theta}} \right], \\ &\text{s.t. } \Pr\left(T_{\mathrm{z}}[k] \le \overline{T}_{\mathrm{z}} \right) > 1 - \epsilon, \forall k, \end{aligned} \tag{4}$$

where ϵ is a small enough number for high confidence in ensuring the thermal safety requirement. Note that the constraint in Eq. (4) is expressed in the probabilistic form because $T_z[k]$ is stochastic due to the stochasticity of $P_{\mathrm{IT}}[k-1]$.

The proposed Safari approach that aims at addressing the CMDP problem in Eq. (4) consists of the following two stages.

■ *Offline imitation learning*: Before the DDPG agent is applied, it is trained offline to imitate an existing conventional safe controller using the historical data traces generated by the safe controller. Meanwhile, these traces are also used to fit a state transition model (e.g., Eq. (2)) that will be used for the online stage. With imitation learning, the DDPG agent produces much less safety violations when interacting with the DC.

■ *Online post-hoc rectification*: After the DDPG agent is applied, it learns the optimal policy by interacting with the DC. To ensure the constraint in Eq. (4), after an action $\boldsymbol{\mu}[k]$ is recommended by the DDPG agent at $t = k\tau$, we use the state transition model obtained in the offline stage to predict the zone temperature resulted from $\boldsymbol{\mu}[k]$ at the end of the control period. Denote by $\widetilde{T}_z[k+1]$ the predicted temperature and by $\widetilde{T}_z[k+1] = h(\boldsymbol{\mu}[k], P_{\mathrm{IT}}[k], \cdots)$ the state transition model, where we use the "$\cdots$" to represent the other factors that the prediction needs to consider. If $\widetilde{T}_z[k+1]$ exceeds $\overline{T}_z$, we solve the following problem to find a rectified action $\boldsymbol{\mu}^*[k]$:

$$\boldsymbol{\mu}^*[k] = \arg\min_{\boldsymbol{\mu}'} \|\boldsymbol{\mu}', \boldsymbol{\mu}[k]\|_2^2/2, \quad \text{s.t. } h(\boldsymbol{\mu}', P_{\mathrm{IT}}[k], \cdots) \le \overline{T}_z. \tag{5}$$

The ℓ_2 norm minimization in Eq. (5) aims at preserving the policy learned by DDPG. The accuracy of the state transition model $h(\boldsymbol{\mu}[k], P_{\mathrm{IT}}[k], \cdots)$ is critical to the safety compliance of the post-hoc rectification. The existing studies on post-hoc rectification [8, 11] adopt linear state transition models such that the problem in Eq. (5) is a tractable convex optimization problem. Unfortunately, the thermal state transition in DC is nonlinear. §5.3 will present various state transition models and analyze their efficacy for the safety-oriented post-hoc rectification.

5.2 Offline Imitation Learning

The imitation learning uses a training dataset over M consecutive control periods: $\{\mathbf{s}_{\mathrm{safe}}[m], \mathbf{a}_{\mathrm{safe}}[m] | m = 1, \ldots, M\}$, where $\mathbf{a}_{\mathrm{safe}}[m]$ is the action performed by the conventional safe controller on the state $\mathbf{s}_{\mathrm{safe}}[m]$ in the mth control period. Such a dataset can be retrieved from the DCIM. The DDPG agent's parameters $\boldsymbol{\theta}$ is trained using the dataset to minimize the following loss function:

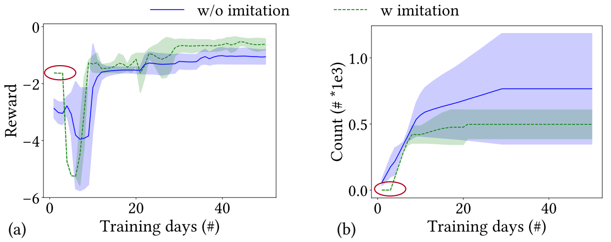

Fig. 5: Effectiveness of imitation learning. (a) Per-day reward average; (b) cumulative count of safety violations.

$\mathcal{L}_{\text{imit}}(\boldsymbol{\theta}) = \frac{1}{M}\sum_{m=0}^{M} \|\boldsymbol{\mu}_{\boldsymbol{\theta}}(\mathbf{s}_{\text{safe}}[m]) - \mathbf{a}_{\text{safe}}[m]\|_2^2$. On the completion of the offline imitation learning, the DDPG agent captures the control policy of the conventional safe controller.

Now, we present an experiment to investigate the effectiveness of the offline imitation learning. In this experiment, two groups of DDPG agents, with and without imitation learning respectively, are deployed to interact with the DC and further updated with online data according to the reward function in Eq. (3). Figs. 5(a) and (d) show the traces of reward and cumulative safety violation count of the two groups of DDPG agents, respectively. From Figs. 5(a) and (b), the agents with imitation learning have high rewards and no safety violations in the first three days. However, from the 4th to the 20th day, these agents start to generate safety violations when they explore better policies. Nevertheless, from Fig. 5(b), imitation learning can reduce the cumulative violation count.

In summary, imitation learning accelerates DRL convergence and alleviates the safety concern of DRL. §5.3 will further develop online post-hoc rectification aiming at eliminating safety violations.

5.3 Online Post-hoc Rectification

As discussed in §5.1, the accuracy of the state transition model $h(\boldsymbol{\mu}[k], P_{\text{IT}}[k], \cdots)$ is critical to the safety compliance of post-hoc rectification. In this section, we first discuss a possible design that uses a long short-term memory (LSTM) network to model the transition. Our experiments show that it requires exploratory data. Then, we present three designs of Safari, i.e., Safari-1, Safari-2, and Safari-3, with different transition models that progressively integrate more prior knowledge and run-time information. Safari-1 uses the steady state transition model in Eq. (2). Safari-2 uses the transient model in Eq. (1) and is unleashed from Assumption 2. Based on Safari-2, Safari-3 applies the maximum ramp-up trajectory of P_{IT} observed in history as the predicted trajectory within the next control period and is further unleashed from Assumption 1.

5.3.1 *A possible design of LSTM-based rectification.*

LSTM networks can model complex and nonlinear temporal correlations. However, the non-exploratory data generated by the conventional safe controller may not support fitting an LSTM to capture the transitions to unsafe states explored by DDPG. To investigate this issue, we build a three-layer LSTM that predicts the next state based on a candidate action and the state, action traces in the past 20 control periods. We conduct experiments to investigate the LSTM's requirement on training data. Fig. 6 shows the distributions of *non-exploratory data* (produced by the baseline controller), *random exploratory data* (produced by a controller performing random actions), and *marginally safe exploratory data* (produced by a controller performing clipped random actions), which have increasing coverage in the state and action spaces. The mean absolute errors (MAEs) of the predictions made by the LSTMs trained using these datasets are shown by the histograms labeled "LSTM" in Fig. 7. The LSTM trained with the random exploratory data achieve MAEs lower than 0.5°C, indicating the LSTM design is satisfactory. The LSTMs trained with non-exploratory and marginally safe exploratory data have high MAEs, due to their poor performance in characterizing the transitions to unsafe states. Fig. 7 includes results for both a CW cooling system and a DX cooling system. From the above results, this LSTM-based design requires exploratory data including the unsafe states, which are in general unavailable and contradictory to the original goal of ensuring safety.

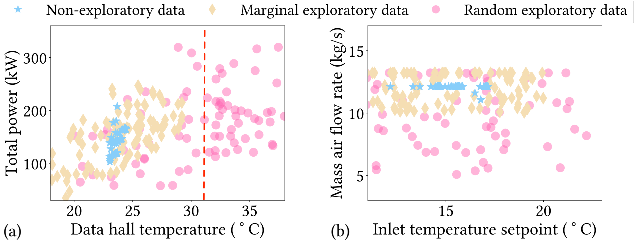

Fig. 6: Non-exploratory data, random exploratory data, and marginally safe exploratory data. (a) State; (b) action.

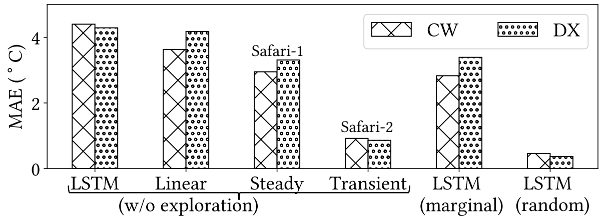

Fig. 7: Test MAE of different state transition models. The test data are randomly sampled, including the unsafe state.

5.3.2 *Safari-1: Steady state transition-based rectification.*

Safari-1 uses the non-exploratory data produced by the conventional safe controller to fit the parameter α in Eq. (2). Then, Safari-1 uses Eq. (2) as the prediction model $\widetilde{T}_{\text{z}}[k+1] = h(\boldsymbol{\mu}[k], P_{\text{IT}}[k], \cdots)$. If $\widetilde{T}_{\text{z}}[k+1]$ exceeds $\overline{T}_{\text{z}}$, Safari-1 solves the convex optimization problem in Eq. (5) using its Karush-Kuhn-Tucker (KKT) condition:

$$\begin{cases} \hat{T}_{\text{in}}^*[k] - \hat{T}_{\text{in}}[k] + \lambda = 0, \\ \hat{f}^*[k] - \hat{f}[k] - \lambda \dfrac{P_{\text{IT}}[k+1]}{\alpha(\hat{f}^*[k])^2} = 0, \\ \lambda \left(\hat{T}_{\text{in}}^*[k] + \dfrac{P_{\text{IT}}[k+1]}{\alpha \hat{f}^*[k]} - \overline{T}_{\text{z}} \right) = 0, \end{cases} \tag{6}$$

where λ is the Lagrangian multiplier, $\boldsymbol{\mu}^*[k] = (\hat{T}_{\text{in}}^*[k], \hat{f}^*[k])$ is the rectified action. Under the definition $P_{\text{IT}}[k+1] \triangleq P_{\text{IT}}(t)|_{t\in(k\tau,(k+1)\tau)}$, $P_{\text{IT}}[k+1]$ is unknown when the DDPG agent chooses the action at

$t = k\tau$. However, pragmatically, the controller can wait for a short while until $P_{\mathrm{IT}}[k+1]$ is observable and then solve Eq. (6).

If the state evolution strictly follows the steady state transition in Eq. (2), the solution to Eq. (6) can ensure safety. However, in practice, the DC cooling system components' primary controls may have a convergence process longer than the control period. This issue may undermine the safety assurance of the solution given by Eq. (6). This motivates us to adopt the original transient model in Eq. (1) to guide the rectification.

5.3.3 Safari-2: Transient-based rectification. To predict $T_{\mathrm{z}}[k+1]$ more accurately, we need to further consider the transient of T_{in} within a control period, which depends on the primary controls of the CRAC units and the back-end cycles (i.e., the chilled water cycle and the condenser water cycle). Thus, the accurate prediction of T_{in} transient requires a precise model of the whole cooling system. The high modeling overhead is undesirable.

In this section, we develop a heuristic prediction approach merely based on Eq. (1). From Eq. (1), the trajectory of $T_{\mathrm{z}}(t)$ depends on the trajectories of $Q(t)$, $T_{\mathrm{in}}(t)$, and $f(t)$. From Assumption 1, the $Q(t)|_{t\in[k\tau,(k+1)\tau)}$ remains constant at $P_{\mathrm{IT}}[k+1]$. For $T_{\mathrm{in}}(t)$ and $f(t)$, we adopt their setpoints as their approximations. Specifically, we set $T_{\mathrm{in}}(t)|_{t\in[k\tau,(k+1)\tau)} = \hat{T}_{\mathrm{in}}[k]$ and $f(t)|_{t\in[k\tau,(k+1)\tau)} = \hat{f}[k]$. Then, with the initial condition $T_{\mathrm{z}}(k\tau) = T_{\mathrm{z}}[k]$, we can solve $T_{\mathrm{z}}(t)$ from Eq. (1) as $T_{\mathrm{z}}(t) = W[k] + (T_{\mathrm{z}}[k] - W[k])\,\mathrm{e}^{-\hat{f}[k](t-k\tau)/V_{\mathrm{s}}}$, $t \in [k\tau, (k+1)\tau)$, where $W[k] = \hat{T}_{\mathrm{in}}[k] + \frac{P_{\mathrm{IT}}[k+1]}{\alpha\hat{f}[k]}$ is a constant within the kth control period. Then, we mitigate the impact of making approximations for $T_{\mathrm{in}}(t)$ and $f(t)$ by adopting the average of $T_{\mathrm{z}}(t)$ as the prediction, i.e., $\widetilde{T}_{\mathrm{z}}[k+1] = \frac{1}{\tau}\int_{k\tau}^{(k+1)\tau} T_{\mathrm{z}}(t)\mathrm{d}t$.

Safari-2 uses the above heuristic prediction approach to predict $\widetilde{T}_{\mathrm{z}}[k+1]$ for the action $\boldsymbol{\mu}$ recommended by DDPG. If $\widetilde{T}_{\mathrm{z}}[k+1]$ exceeds $\overline{T}_z$, it applies grid search in the two-dimensional action space to solve the problem in Eq. (5), in which $h(\boldsymbol{\mu}', P_{\mathrm{IT}}[k], \cdots)$ given any candidate rectified action $\boldsymbol{\mu}'$ is also computed by the above heuristic prediction approach. Since the dimension of the search space is low (i.e., two), the computational overhead of the grid search is acceptable. For instance, our Safari-2 implementation only takes at most 0.2 seconds to complete the search.

5.3.4 Safari-3: Integrate predicted IT power trajectory. Safari-2 and 3 only differ in the algorithm to predict the trajectory $T_{\mathrm{z}}(t)$. Safari-3's prediction algorithm is as follows. First, during offline stage, Safari-3 builds the *maximum ramp-up function* $P^{\nearrow}(\Delta t | P_{\mathrm{IT}}^{\mathrm{start}})$ for IT power from the historical trace of IT power, where Δt represents the relative time. Specifically, it is the upper envelope of all IT power traces with length of τ minutes provided that the starting IT power is $P_{\mathrm{IT}}^{\mathrm{start}}$. Then, at $t = k\tau$, Safari-3 adopts $T_{\mathrm{in}}(t) = \hat{T}_{\mathrm{in}}[k]$, $f(t) = \hat{f}[k]$, and $Q(t) = P^{\nearrow}(t - k\tau | P_{\mathrm{IT}}[k])$ to solve $T_{\mathrm{z}}(t)$ from Eq. (1), where $t \in (k\tau, (k+1)\tau]$. Since Safari-3 uses the maximum ramp-up observed in the history, the predicted $Q(t)$ is conservatively high, which is beneficial to unsafety prevention.

In Fig. 8(a), the gray curves show the aggregated IT utilization ramp ups in a historical trace collected in a real DC (cf. Appendix C) when the starting IT utilization is 40%. The upper envelope of these curves is the maximum ramp up. Fig. 8(b) shows the maximum ramp ups when the starting IT utilization is 40%, 45%, and 50%.

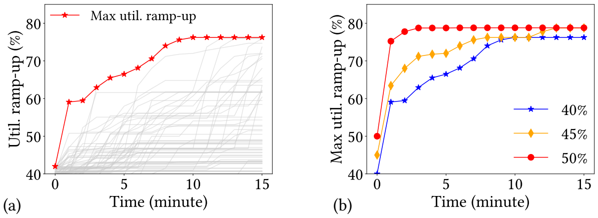

Fig. 8: (a) IT utilization ramp ups (gray curves) and max ramp up (red curve) when starting utilization is 40%; (b) max ramp ups when starting IT utilization is 40%, 45%, and 50%, respectively.

5.3.5 Comparing state prediction models of Safari-1, -2, and [11]. Fig. 7 also shows the MAEs of the state prediction models used by Safari-1 and Safari-2, as well as a linear model used in [11], when $P_{\mathrm{IT}}(t)$ follows Assumption 1. The results are labeled "Steady", "Transient", and "Linear", respectively. The linear transition model uses the design of [11] to predict the next state temperature by $\widetilde{T}_{\mathrm{z}}[k+1] = g_{\omega}(\mathbf{s}[k])^{\top}\boldsymbol{\mu}[k] + T_{\mathrm{z}}[k]$ where g_{ω} is modeled using a three-layer MLP to predict the linear correlation coefficients and each layer has 32 neurons. If only non-exploratory training data produced by the baseline controller are used, Safari-2 achieves the lowest MAEs of less than 0.9°C. In addition, the steady-state transition-based prediction model used by Safari-1 outperforms the linear prediction model in [11]. The performance of the various designs of Safari in preventing safety violations will be extensively evaluated in §6.

6 PERFORMANCE EVALUATION

6.1 Evaluation Methodology and Settings

We use EnergyPlus to simulate the physical processes of a CW-cooled DC and a DX-cooled DC. We use the 1-year weather data of Singapore and Chicago in the tropical and temperate climate zones, respectively. Fig. 12(a) of Appendix C shows the weather temperature traces of the two cities. By default, we consider the tropical condition. Other default settings for the DDPG agent and the simulation environments such as the outdoor condition and IT workload have been described in §4. We have implemented the three designs of Safari presented in §5.3 and the following baseline approaches discussed in §2:

■ **Baseline controller** is the EnergyPlus' built-in controller as described in §4.

■ **Reward shaping** refers to the DDPG agent presented in §4.1 that uses Eq. (3) as the shaped reward function. It captures the essence of [9, 17, 25, 28].

■ **Simplex** follows the essence of [19, 24]. Specifically, when the observed system state is safe, the DDPG agent is applied. Once an unsafe state is observed, the next action is set to the minimum allowable inlet temperature setpoint (i.e., 10°C) and maximum allowable supply air flow rate (i.e., 15 kg/s).

■ **Projection** implements the post-hoc rectification by solving Eq. (5) with the linear transition model described in §5.3.5. It captures the essence of [8, 11].

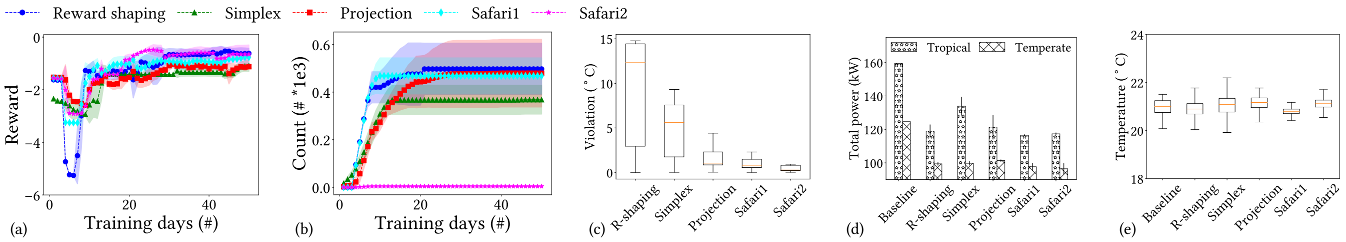

Fig. 9: Performance of various approaches on CW-cooled DC. (a) Per-day reward average during learning; (b) cumulative count of safety violations during learning; (c) violation magnitudes during learning; (d) DC total power during 1-year testing in two climates; (e) zone temperature distribution during 1-year testing.

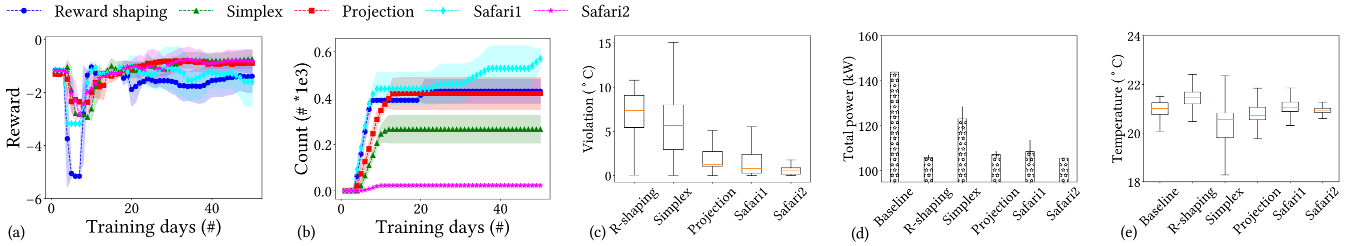

Fig. 10: Performance of various approaches on a DX-cooled DC under tropical climate.

6.2 Evaluation Results for a CW-cooled DC

We conduct simulations based on the simple IT utilization pattern described in §4.2, which satisfies Assumption 1. Fig. 9(a) shows the per-day reward averages of various approaches in the first 50 days. The high rewards in the first several days are due to imitation learning. The rewards stabilize after about 20 days training. Fig. 9(b) and (c) show the cumulative count and distribution of the violation magnitudes during DRL. The reward shaping exhibits the poorest performance in terms of either violation count or magnitude. In Fig. 9(b), the simplex, projection, and Safari-1 produce hundreds of violations in the 50 days. In contrast, Safari-2 only produces five violations. From Fig. 9(c), the projection, Safari-1, and Safari-2 produce smaller violation magnitudes compared with the reward shaping and simplex. This suggests that the proactive unsafety prevention measures are better than the reactive ones. Safari-1 produces lower violation magnitudes compared with the projection. This shows that the steady state transition model in Eq. (2) is better than the linear model in [11]. Safari-2 achieves the lowest violation count and magnitudes. Specifically, at the 50th day, the violation count of Safari-2 is only 1.0%, 1.4%, 1.0%, and 1.1% of those of reward shaping, simplex, projection, and Safari-1, respectively. The 3rd quartile of temperature violation magnitudes of Safari-2 is only 0.81°C, lower than the 14.5°C, 7.6°C, 2.3°C and 1.48°C of reward shaping, simplex, projection, and Safari-1. Figs. 9(d) and (e) show the DC's total power and the zone temperature under various controllers during testing. Safari-1 and Safari-2 achieve similar power savings and outperform the other baseline approaches. In summary, Safari-2 achieves 26.4% and 22.7% power savings compared with the baseline controller in the tropical and temperate climates, respectively. It also effectively prevents unsafety during learning and maintains small temperature deviations during testing.

Table 2: Performance under real IT utilization trace.

Approach	DC total power (kw)	Violation count (#)	Violation magnitude (°C)		
			Q1	Q2	Q3
Baseline	110.69	N.A.	N.A.	N.A.	N.A.
R-shaping	79.48	3446	1.86	4.46	7.13
Safari-2	81.27	42	0.42	0.65	1.31
Safari-3	82.22	18	0.14	0.34	0.73

Q1, Q2, Q3 represent the 1st, 2nd, 3rd quartiles.

Next, we conduct a set of simulations using a 6-day real IT utilization trace of 4,000 servers collected from a data center [1]. Fig. 12(b) in Appendix C shows the aggregated utilization trace. The trace is re-sampled with one-minute interval, which is the finest zone time granularity setting of EnergyPlus. Therefore, the P_{IT} changes within each control period of 15 minutes. We choose the first four days to construct the maximum ramp-up function and the remaining two days' data repeatedly to drive the simulations. This set of simulations mainly evaluates the performance of Safari-2 and Safari-3 when Assumption 1 is not strictly followed. Table 2 shows the results. Safari-3 saves 25.7% power usage compared with the baseline controller, reduces thermal violations by 99%, and maintains sub-1°C 3rd quartile of violations.

6.3 Evaluation Results for a DX-cooled DC

We conduct simulations in which the simulated IT power satisfies Assumption 1. Fig. 10 shows the evaluation results. The CW and DX systems generate different impacts on the validness of Assumption 2, because they have different cooling components and the associated primary controls. From Fig. 10(b), Safari-1 produces more violations in the DX-cooled DC than the CW-cooled

DC. This implies that the validness of the steady state transition assumption (i.e., Assumption 2) is weakened in DX-cooled DC. Nevertheless, Safari-2 still performs satisfactorily. At the 50th day, the violation count of Safari-2 is only 5.5%, 8.6%, 4.9%, and 4.0% of those of reward shaping, simplex, projection, and Safari-1, respectively. The 3rd quartile of temperature violation magnitudes of Safari-2 is only 0.99°C, lower than the 9.1°C, 8.0°C, 2.7°C and 2.4°C of reward shaping, simplex, projection, and Safari-1. Safari-2 achieves 26.6% average power savings compared with the baseline controller.

7 DISCUSSION

This section discusses two issues not addressed in this paper.

Multi-hall DC: The MDP formulation can be extended and the DDPG algorithm is still applicable. Specifically, the actions and states of the data halls are concatenated to form the action and state of the whole DC. The temperature-related reward components of all data halls can be aggregated with the reward component $-\lambda_P P_{DC}$ to form the reward for the DC. The data halls may adopt different zone temperature setpoints. When a subset of data halls wish to use the conventional safe CRAC control, they can be excluded from the MDP formulation and viewed exogenous. The post-hoc rectification of Safari-3 can be applied in each data hall independently.

Eliminating thermal violations: From the evaluation results, Safari-3 can effectively prevent thermal violations. Although the ultimate goal of eliminating any thermal violations is desirable, the stochastic nature of the zone temperature as explained in §5.1 makes the guaranteed elimination difficult. To achieve guaranteed elimination, thinking-outside-the-box solutions will be needed. A possible solution is as follows. Typically, redundant CRAC units are deployed for fail-safe operations. A standby CRAC unit is activated when its paired unit fails. The DC operator can build a controllable conduct that can direct the cold supply air to the hot zone when needed. When a nearly unsafe state is detected via close temperature monitoring (e.g., every second), the system can activate the standby CRAC unit and direct the cold air to the hot zone. With Safari-3 deployed, the activations of this standby CRAC unit are rare. Thus, the energy usage of this last line of defense is negligible.

8 CONCLUSION

This paper presents Safari, an approach toward safe DRL for single-hall DC cooling control. By integrating imitation learning and post-hoc rectification designed based on the thermodynamic law governing the heat process in the data hall, Safari can effectively prevent thermal unsafety in the hot zone. Our extensive evaluation that covers both chilled water and direct-expansion cooling systems under two climate conditions shows that, with varying IT workload pattern, Safari saves more than 22% total data center power compared with conventional control, reduces safety violations up to 99% compared with reward shaping. Safari sheds lights on deployment of DRL algorithms to safety-critical cyber-physical systems.

ACKNOWLEDGMENTS

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Energy Research Testbed and Industry Partnership Funding Initiative of the Energy Grid (EG) 2.0 programme and its Central Gap Fund ("Central Gap" Award No. NRF2020NRF-CG001-027) and its NTUitive Gap Fund administrated by the NTUitive Pte Ltd and Ministry of Education.

REFERENCES

[1] 2021. Alibaba cluster trace program. https://github.com/alibaba/clusterdata.
[2] 2021. EnergyPlus Setpoint Managers. https://bit.ly/3EtLmZp.
[3] 2021. Global Internet Data Centers Market Report 2021. (2021).
[4] B. Arguello-Serrano and M. Velez-Reyes. 1999. Nonlinear control of a heating, ventilating, and air conditioning system with thermal load estimation. *IEEE Trans. Control Syst. Technol.* 7, 1 (1999), 56–63.
[5] ASHRAE. 2011. 2011 Thermal guidelines for data processing environments–expanded data center classes and usage guidance.
[6] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. 2016. Openai gym. *arXiv:1606.01540* (2016).
[7] B. Chen, Z. Cai, and M. Bergés. 2019. Gnu-RL: A precocial reinforcement learning solution for building hvac control using a differentiable mpc policy. In *ACM BuildSys.* 316–325.
[8] B. Chen, P. Donti, K. Baker, Z. Kolter, and M. Berges. 2021. Enforcing Policy Feasibility Constraints through Differentiable Projection for Energy Optimization. In *ACM e-Energy.* 199–210.
[9] C. Chi, K. Ji, A. Marahatta, P. Song, F. Zhang, and Z. Liu. 2020. Jointly optimizing the IT and cooling systems for data center energy efficiency based on multi-agent deep reinforcement learning. In *ACM e-Energy.*
[10] D. Crawley, L. Lawrie, C. Pedersen, and F. Winkelmann. 2000. EnergyPlus: energy simulation program. *ASHRAE J.* 42, 4 (2000).
[11] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa. 2018. Safe exploration in continuous action spaces. *arXiv:1801.08757* (2018).
[12] X. Ding, W. Du, and A. Cerpa. 2020. MB2C: Model-Based Deep Reinforcement Learning for Multi-zone Building Control. In *ACM BuildSys.*
[13] N. El-Sayed, I. Stefanovici, G. Amvrosiadis, A. Hwang, and B. Schroeder. 2012. Temperature management in data centers: Why some (might) like it hot. In *ACM SIGMETRICS.* 163–174.
[14] P. Geibel and F. Wysotzki. 2005. Risk-sensitive reinforcement learning applied to control under constraints. *J. Artificial Intelligence Research* 24 (2005).
[15] A. Jain, T. Nghiem, M. Morari, and R. Mangharam. 2018. Learning and control using Gaussian processes. In *ACM/IEEE ICCPS.* IEEE, 140–149.
[16] N. Lazic, T. Lu, C. Boutilier, M. Ryu, E. Wong, B. Roy, and G. Imwalle. 2018. Data center cooling using model-predictive control. In *NeurIPS.* 3818–3827.
[17] Y. Li, Y. Wen, D. Tao, and K. Guan. 2019. Transforming cooling optimization for green data center via deep reinforcement learning. *IEEE Trans. Cybern.* 50, 5 (2019), 2002–2013.
[18] T. Lillicrap, J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv:1509.02971* (2015).
[19] H. Mao, M. Schwarzkopf, H. He, and M. Alizadeh. 2019. Towards Safe Online Reinforcement Learning in Computer Systems. In *NeurIPS.*
[20] H. Menon, B. Acun, S.G. De Gonzalo, O. Sarood, and L. Kalé. 2013. Thermal aware automated load balancing for hpc applications. In *IEEE CLUSTER.* 1–8.
[21] T. Moriyama, G. De Magistris, M. Tatsubori, T.-H. Pham, A. Munawar, and R. Tachibana. 2018. Reinforcement learning testbed for power-consumption optimization. In *AsiaSim.* 45–59.
[22] K. Ogata. 1995. *Discrete-time control systems.* Prentice-Hall, Inc.
[23] A. Paszke, F. Gross, S.and Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, L. Gimelshein, N. Antiga, and A. Desmaison. 2019. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS* 32 (2019).
[24] D. Phan, R. Grosu, N. Jansen, N. Paoletti, S. Smolka, and S. Stoller. 2020. Neural simplex architecture. In *NASA Formal Methods Symposium.* Springer, 97–114.
[25] Y. Ran, H. Hu, X. Zhou, and Y. Wen. 2019. DeepEE: Joint optimization of job scheduling and cooling control for data center energy efficiency using deep reinforcement learning. In *IEEE ICDCS.* 645–655.
[26] A. Shehabi, S. Smith, D. Sartor, R. Brown, M. Herrlin, J. Koomey, E. Masanet, N. Horner, I. Azevedo, and W. Lintner. 2016. US data center energy usage report.
[27] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, et al. 2017. Mastering the game of go without human knowledge. *Nature* 550, 7676 (2017), 354–359.
[28] D. Van Le, Y. Liu, R. Wang, R. Tan, Y.-W. Wong, and Y. Wen. 2019. Control of Air Free-Cooled Data Centers in Tropics via Deep Reinforcement Learning. In *ACM BuildSys.* 306–315.
[29] R. Wang, D. Van Le, R. Tan, Y.-W. Wong, and Y. Wen. 2020. Real-Time Cooling Power Attribution for Co-Located Data Center Rooms with Distinct Temperatures. In *ACM BuildSys.* 190–199.
[30] Y.-G. Wang, Z.-G. Shi, and W.-J. Cai. 2001. PID autotuner and its application in HVAC systems. In *ACC*, Vol. 3. IEEE, 2192–2196.
[31] C. Zhang, S. Kuppannagari, R. Kannan, and V. Prasanna. 2019. Building HVAC scheduling using reinforcement learning via neural network based model approximation. In *ACM BuildSys.* 287–296.

A SUMMARY OF NOTATION

Table 3 summarizes the symbols used throughout this paper, which are grouped into four categories of power/heat-related, temperatures, air volume-related, and DDPG-related.

Table 3: Summary of Notation

Symbol	Definition
P_{IT}	IT power usage
P_{c}	cooling power usage
P_{DC}	DC total power usage, $P_{\text{DC}} = P_{\text{IT}} + P_{\text{c}}$
U_{IT}	IT utilization
Q	sensible heat load, $Q = P_{\text{IT}}$ in analysis
T_{in}	cold aisle temperature
$\hat{T}_{\text{in}}$	setpoint for T_{in}
T_{z}	zone temperature
$\overline{T}_{\text{z}}$	thermal safety upper bound for T_{z}
T_{C}	setpoint for T_{z} in DDPG
$\widetilde{T}_{\text{z}}$	predicted T_{z}
T_{o}	outdoor air temperature
f	total volumetric flow rate of cold supply air
$\hat{f}$	setpoint for f
V_s	volume of the data hall
α	a system dependent parameter
τ	control period
$\boldsymbol{\mu}$	action, $\boldsymbol{\mu} = (\hat{T}_{\text{in}}, \hat{f})$
$\mathbf{s}$	state $\mathbf{s} = (T_{\text{z}}, T_{\text{in}}, P_{\text{c}}, P_{\text{IT}}, T_{\text{o}})$
r	reward function
$T_{\text{L}}, T_{\text{U}}$	bounds for T_{z} for reward shaping
$\lambda_1, \lambda_2, \lambda_{\text{T}}, \lambda_{\text{P}}$	coefficients of reward function

B A BRIEF INTRODUCTION OF DX COOLING

Different from the CW system that has three cycles, the DX system has two cycles only as shown in Fig. 11. It directly cools the air through the evaporation and condensation of refrigerant. It consists of a compressor, an evaporator, a condenser, and an expansion valve. The heat is removed via the following process. At the evaporator, hot air is extracted from the data hall and blown through the heat exchange coil by the CRAC fan. The liquid refrigerant in the coil absorbs the heat and expands into vapour. Then, the compressor uses electricity to drive the refrigerant vapour into high pressure gas. At the condenser, heat is dissipated to the outside environment and the refrigerant turns back to liquid.

Table 4: Hyperparameter settings of DDPG.

Hyperparameter	Setting	Hyperparameter	Setting
Training Batch size	1,024	Update per step	96
Actor learning rate	0.001	Critic learning rate	0.001
Actor hidden layer	[32, 32]	Critic hidden layer	[32, 32]
replay buffer size	1×10^7	Discounted factor (γ)	0.99

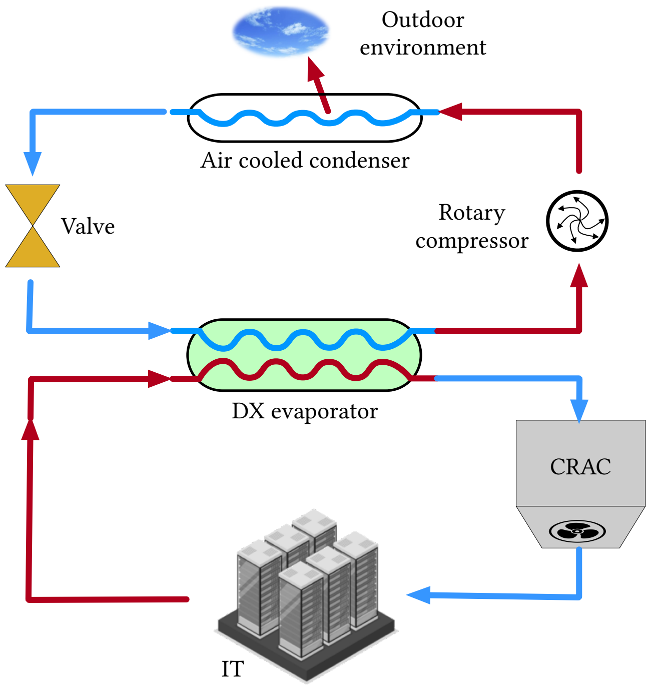

Fig. 11: DX-cooled DC.

C USED SETTINGS AND DATA

Table 4 summarizes the hyperparameter settings of DDPG. Figs. 12(a) and (b) show the outdoor air temperature and IT utilization data used for evaluation. The outdoor temperature data, which are provided by EnergyPlus, were collected from Singapore and Chicago in the tropical and temperate climate zones, respectively. Fig. 12(b) shows the aggregated CPU utilization trace collected from a real Internet DC hosting 4,000 servers [1].

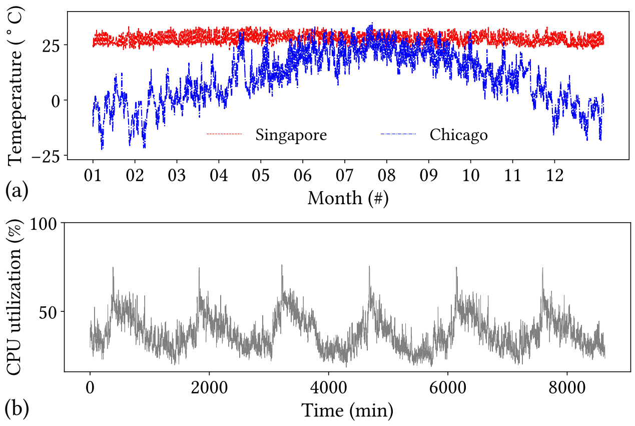

Fig. 12: (a) Historical weather data at Singapore and Chicago; (b) aggregated IT utilization trace in a real DC hosting 4,000 servers [1].