Green Data Center Cooling Control via Physics-Guided Safe Reinforcement Learning

RUIHANG WANG, ZHIWEI CAO, XIN ZHOU, YONGGANG WEN, and RUI TAN, Nanyang Technological University, Singapore

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 constraints 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 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 show that our approach saves 18% to 26.6% of total data center power compared with conventional control and reduces safety violations by 94.5% to 99% compared with reward shaping. We also extend the proposed framework to address data centers with non-uniform temperature distributions for detailed safety considerations. The evaluation shows that our approach saves 14% power usage compared with the PID control while addressing safety compliance during the training.

CCS Concepts: • **Hardware → Enterprise level and data centers power issues**; • **Computing methodologies → Reinforcement learning**.

Additional Key Words and Phrases: Data center, safe reinforcement learning, energy efficiency, computational fluid dynamics, proper orthogonal decomposition

ACM Reference Format:
Ruihang Wang, Zhiwei Cao, Xin Zhou, Yonggang Wen, and Rui Tan. 2022. Green Data Center Cooling Control via Physics-Guided Safe Reinforcement Learning. *ACM Trans. Cyber-Phys. Syst.* 1, 1 (January 2022), 26 pages. https://doi.org/XXXXXXX.XXXXXXX

A preliminary version of this work appears in The 13th ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS) held virtually in May 2022. 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. The authors would like to acknowledge the support from Xinyi Zhang during the early stage of this research.
Authors' address: Ruihang Wang, ruihang001@ntu.edu.sg; Zhiwei Cao, zhiwei003@ntu.edu.sg; Xin Zhou, zhouxin@ntu.edu.sg; Yonggang Wen, ygwen@ntu.edu.sg; Rui Tan, tanrui@ntu.edu.sg, Nanyang Technological University, School of Computer Science and Engineering, 50 Nanyang Avenue, Singapore, 639798.

2378-962X/2022/1-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Data centers (DCs) form the backbone of the Internet ecosystem. The DC market has ever been growing to meet the demands of cloud computing and storage services. In the ongoing COVID-19 pandemic, the DC industry needs to expand to support the surging online activities. As such, the global DC market value is forecast to reach 143.4 billion U.S. dollars by 2027 according to the current compound annual growth rate of 13.4% [4]. However, DCs are energy-intensive. According to a survey in 2022, the sector of the DC industry uses about 1.8% of the electricity in the U.S. and contributes 0.3% global carbon emissions [10]. Given the fast DC market growth, it is important to improve DC energy efficiency in the pursuit of carbon neutrality. A DC is a cyber-physical system consisting of information technology (IT) equipment and cooling systems. 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 [40]. 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.

This paper considers the problem of DC cooling control that aims at reducing the DC energy usage subject to the IT equipment's thermal safety constraints. Any IT device specifies the highest temperature that it can tolerate (e.g., 32°C for ASHRAE Class A1 servers [8]). 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 workloads, the operating point, i.e., the temperature and mass flow rate, of the computer room air conditioning (CRAC) units need to be periodically adjusted to maintain the zone temperature. This can be achieved by conventional feedback controls [45].

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 [29, 41]. Recent studies [11, 16] 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 [11]. 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 thermal safety requirements.

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 action. For an intricate MDP problem, the convergence of the DRL often requires experiencing a large number of action-state trials. For instance, model-free DRL for HVAC control in [16] performs 500,000 interactions to converge. To apply DRL for DC, it is critical to avoid the data hall's excursions to thermal unsafety during the learning process, which forms a constrained MDP (CMDP) problem. To tackle CMDP in the general context, recent studies (e.g., [24, 37]) adopt a *reward shaping* approach that applies a penalty in the reward function when the constraints are violated. However, this approach, which is essentially a Lagrangian relaxation [18], does not explicitly enforce the constraints. *Post-hoc rectification* is another approach that explicitly addresses the constraints of

CMDP. Specifically, in each control period, the approach aims to find the smallest rectification to the potential unsafe recommendation made by the DRL agent such that the rectified action will not drive the system to the unsafe region. The study [15] has derived the closed-form rectifications when the controlled system follows a linear state transition. Under the same linear assumption, the study [12] incorporates the rectification into the DRL training with a differentiable projection layer. However, the thermal state transition in DC is nonlinear and non-differentiable in terms of the control action. As such, the domain-agnostic solutions based on the linear approximation of the thermal state transition will inevitably lead to degradation of thermal safety compliance.

In this work, we first present a safety-aware reinforcement learning framework (Safari) for DC cooling control. We consider both single-hall and multi-hall schemes. The single-hall and multi-hall schemes are often adopted in enterprise [22] and co-located DCs [43], respectively. 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 and accelerate the convergence in the online stage. Second, for the online stage, we design a new post-hoc rectification approach based on state transition models that capture 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 for 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 exploratory training data, which is in general unavailable and contradictory to the original goal of ensuring safety.

To capture the thermal state transition, the above setting assumes that the air of the considered data hall is well-mixed to exhibit uniform spatial temperature distribution. Under this assumption, the thermal transition is modeled using an ordinary differential equation (ODE) that can be solved with low computation overhead for online action rectification. For a DC that hosts diverse IT equipment for different computing tasks, the spatial temperature distribution can be non-uniform due to the heterogeneous equipment placements and workload distributions. Therefore, it is necessary to extend the transition model with fine-grained spatial temperature prediction capabilities for safety considerations. The computational fluid dynamics and heat transfer (CFD/HT) [36] is a typical technique to characterize the full-fledged temperature distribution of a given space by solving the Navier-Stokes (NS) and energy balance equations [6]. It has been adopted in offline optimization for reducing the DC energy cost and preventing thermal risk [37]. However, the vanilla CFD/HT model does not meet the online rectification requirement due to its compute-intensive and non-differentiable nature. At the start of each control period, the online rectification is expected to perform in time to catch up with the system state transition. Unfortunately, the iterative rectification of the DRL recommended action is computationally prohibitive based on the CFD/HT model.

To address the above computation challenges, we propose a reduced-order modeling approach to accelerate the rectification based on the proper orthogonal decomposition (POD). The POD method aims to describe a full field profile with a linear combination of a set of spatial basis functions, i.e., the POD modes and the corresponding coefficients. Specifically, for the offline stage, we first derive the POD modes and the relationship between the boundary conditions and POD coefficients based on the data generated from a calibrated CFD/HT model [44]. After that, we apply the POD model for online post-hoc rectification. In this paper, we develop the closed-form and heuristic search-based rectifications based on two forms of POD models, respectively. The proposed methods extend the Safari framework for data hall safety considerations with fine-grained spatial temperature modeling.

As the development of the POD model only requires the simulation data from the CFD/HT model, it eliminates the safety concern of collecting exploratory data from the physical DC.

In summary, this paper proposes the Safari framework that enables the adoption of DRL to pursue DC energy savings while effectively preventing excursions to thermal unsafety. 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 [14] to show the DC energy savings achieved by the DRL agent. The study also shows that the agent designed without rigorous thermal safety considerations 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 DCs with chilled water and direct expansion cooling systems in two climate conditions. When the IT workload pattern is simple, Safari saves 22.7% to 26.6% power compared with conventional control and reduces safety violations by 94.5% to 99% compared with reward shaping. When the IT workload pattern is complex, the power savings and violation reductions are 25.7% and 99%, respectively.
- We extend Safari to address DCs with non-uniform temperature distributions for detailed safety considerations. With the extended formulation, we derive the post-hoc rectifications based on two forms of the POD model. Through evaluation, Safari saves 14% power compared with the PID control and addresses the thermal safety constraint compared with other DRL controls.

The rest of this paper is organized as follows. §2 reviews the 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 relevant approaches and summarizes their requirements and implementation properties for safety considerations. 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* [13, 24, 37, 42] and *model-based* [22, 46] approaches.

The model-free approaches learn the control policy by directly interacting with the controlled system, which in general follows the online DRL scheme. The study [24] applies the deep deterministic policy gradient (DDPG) to learn the cooling control policy for a two-zone DC. The studies [37] and [13] adopt the parameterized deep Q-network (DQN) and the DDPG, respectively, to learn the policy for joint control of cooling and IT (e.g., via compute job allocation). The study [42] applies DQN to learn the policy for air free cooling control. After adequate learning, the DRL agents in [13, 24, 37, 42] 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 constraints in a *semi-explicit* manner. Differently, our proposed approach directly and explicitly addresses the thermal safety constraints via post-hoc rectification. As indicated in Table 1, the reward shaping approach needs *exploratory data* that covers the unsafe

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

Category	Approach	Ref.	Application	Requirements for safety		Consideration	
				Exploratory data	Transition model	When?	How?
Model-free*	Simplex	[27, 35]	Load balancing, etc.	Required	Not required	Reactive	Explicit
	Reward shaping	[13, 24, 37, 42]	DC cooling				Semi-explicit
	Post-hoc rectification	[12, 15]	HVAC	Not required†	Linear model	Proactive	Explicit
		Safari	DC cooling	Not required	Physics model		
Model-based*	Reward shaping	[46]	DC cooling	Required	LSTM	Reactive	Semi-explicit
	MPC	[22]		Required	Linear model	–	Implicit
		[20]	Building	Required	Gaussian	–	Implicit
Offline	Regularization	[26]	Building	Not required	Not required	–	Implicit

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

†If the state transition is linear, [12, 15] do not require exploratory data. However, for nonlinear DC thermodynamics, although [12, 15] can be extended to use DNNs capturing nonlinear transitions, our experiments in §5.3 show that exploratory data will be needed to train the DNNs.

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., [20, 22, 46]) 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 dynamics. The study [22] 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 constraints. The study [20] 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 [46] 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. 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 [27, 35] 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. A recent work adopts the offline reinforcement learning algorithm for HVAC control. To address safety concern, a Kullback-Leibler regularization term is incorporated to penalize policies that deviate far from the one learned from historical data. This mechanism reduces the randomness of exploration that could damage the system.

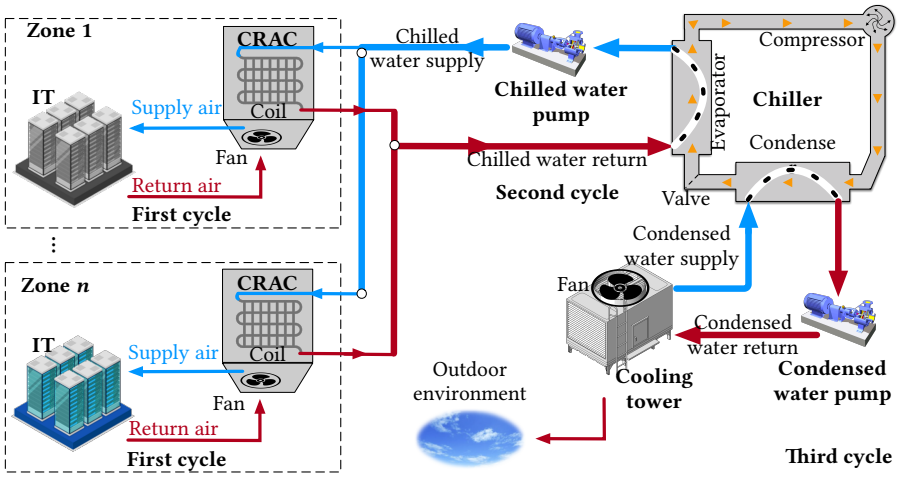

Fig. 1. A typical chilled water-cooled DC system.

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 [15], based on a linear state transition model, a closed-form rectification is found by solving a convex constrained optimization problem. The work [12] 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 [12, 15] depends on the linearity and differentiability of the system dynamics. In this paper, we will analytically show the nonlinear property of the thermodynamics in DC. This paper further advances the post-hoc rectification approach with state transition models incorporating the knowledge of DC thermodynamics to enforce thermal safety. The models can be either fitted with historical non-exploratory data produced under the control of a safe controller or simulation data generated by CFD/HT. As the fitted model remains accurate in the unsafe region, our approach does not require undesirable exploratory data.

3 PRELIMINARIES

This section presents the preliminaries of DC cooling control and DRL. The notations used in this paper are summarized in Table 2, which are grouped into six categories of DC configurations, POD-related, power/heat-related, air volume-related, temperatures, and DDPG-related. The default vectors are in column forms unless particularly specified.

3.1 DC Model and Cooling Control

3.1.1 DC system overview. 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 multiple data halls hosting multiple CRAC units and servers. The single-hall and multi-hall schemes are often adopted in enterprise [22] and co-located DCs [43], respectively. 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, draw hot air from the zone, and cool 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

Table 2. Summary of Notations

Sym.	Definition	Sym.	Definition
l	number of CRACs	N, H	number of field points and POD modes
m	number of servers	$\mathbf{\Phi} \in \mathbb{R}^{N\times H}$	matrix of POD modes
n	number of temperature sensors	$\boldsymbol{\beta} \in \mathbb{R}^{H}$	vector of POD coefficients
P_{IT}	total IT power usage	f_{in}	total mass flow rate of supply air
P_{c}	total cooling power usage	$\hat{f}_{\text{in}}$	setpoint for f_{in}
P_{DC}	DC total power usage, $P_{\text{DC}} = P_{\text{IT}} + P_{\text{c}}$	f_{IT}	mass flow rate of IT equipment
U_{IT}	IT utilization	V_s	volume of the data hall
Q	sensible heat load, $Q = P_{\text{IT}}$ in analysis	α	a system dependent parameter
T_{in}	cold aisle temperature	τ	control period
$\hat{T}_{\text{in}}$	setpoint for T_{in}	$\boldsymbol{\mu} \in \mathbb{R}^{2l}$	action $\boldsymbol{\mu} = (\hat{T}_{\text{in}_1}, \hat{f}_{\text{in}_1}, \ldots, \hat{T}_{\text{in}_l}, \hat{f}_{\text{in}_l})$
T_{z}	zone temperature	$\mathbf{s} \in \mathbb{R}^{n+3}$	state $\mathbf{s} = (T_{\text{z}_1}, \ldots, T_{\text{z}_n}, P_{\text{c}}, P_{\text{IT}}, T_{\text{w}})$
$\overline{T}_{\text{z}}$	thermal safety upper bound for T_{z}	r	reward function
T_{C}	setpoint for T_{z} in DDPG	$T_{\text{L}}, T_{\text{U}}$	bounds for T_{z} for reward shaping
$\widetilde{T}_{\text{z}}$	predicted T_{z} by transition model	$\lambda_{\text{T}}, \lambda_{\text{P}}\ \lambda_{\text{S}}, \lambda_1$	coefficients of reward function
T_{w}	outdoor weather temperature	$\mathcal{G}, \mathcal{T}$	Gaussian, and Trapezoid functions

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 usage 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 [31]. 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. It directly cools the air through the evaporation and condensation of refrigerant. We add a brief description of the DX-cooled system in supplementary material. 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.

3.1.2 Data hall heat process model. Then, we describe the heat process in the data hall. In this paper, we consider two modeling methods of the data hall heat process with 1) nodal dynamics and 2) fine-grained spatial heat transfer, respectively. Their modeling principles are as follows.

■ **Nodal dynamics model.** The nodal dynamics is a simplified modeling method adopted by the EnergyPlus simulation. The model considers a scenario where 1) the CRAC units adopt the same setpoint for the supply air temperature and 2) the zone air temperature has a uniform spatial distribution. The zone temperature of a data hall, denoted by T_{z}, is governed by the following thermodynamic model derived from the law of the energy conservation [7]:

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

where t is time, $f_{\text{in}}(t)$ 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 heat 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 $Q(t) = P_{\text{IT}}(t)$. Note that in the EnergyPlus simulations conducted in this paper, we account for lighting heat as well. In practice, thermal-aware load balancing [28] can be applied to achieve the uniform spatial distribution of the hot zone air temperature. In addition, the total mass flow rate $f_{\text{in}}(t)$ can be attributed to the CRAC units properly to help equalize the IT racks' outlet temperatures. For this scenario, we will not detail the zone temperature distribution. Instead, we focus on the main challenge of improving DC energy efficiency while maintaining the overall thermal safety in the hot zone.

■ **Fine-grained spatial model.** Although today's DCs are often equipped with hot aisle containments to prevent air re-circulation, the temperature distributions can be non-uniform due to the heterogeneous server placements and workload distributions. The CFD/HT is a typical method to characterize fine-grained thermodynamics by solving the NS and energy balance equations. However, the vanilla CFD/HT model is compute-intensive due to the iterative solving process. To reduce the computation overhead, we adopt the POD technique to approximate the CFD/HT-generated temperature field. Let $\mathbf{T}_{\text{z}}(t) \in \mathbb{R}^N$ denote the vector of temperature field containing N discrete points at a time step. With the POD approximation, the temperature field can be expressed by the linear combination of H orthogonal basis functions (i.e., POD modes $\mathbf{\Phi} = (\mathbf{\phi}_1, \mathbf{\phi}_2, \ldots, \mathbf{\phi}_H) \in \mathbb{R}^{N\times H}$) and the corresponding coefficients (i.e., POD coefficients $\mathbf{\beta}(t)=(\beta_1, \beta_2, \ldots, \beta_H) \in \mathbb{R}^H$) as:

$$\mathbf{T}_{\text{z}}(t) = \overline{\mathbf{T}}_{\text{o}} + \sum_{i=1}^{H} \beta_i(t)\mathbf{\phi}_i, \tag{2}$$

where $\overline{\mathbf{T}}_{\text{o}}$ is a vector of the average temperature field from CFD/HT simulation. In practice, the POD modes can be efficiently derived by the snapshot method [19]. The POD coefficients are determined by the boundary conditions of the hosted facilities, i.e., $\mathbf{\beta}(t)=\mathcal{F}(\mathbf{T}_{\text{in}}(t), \mathbf{f}_{\text{in}}(t), \mathbf{P}_{\text{IT}}(t), \mathbf{f}_{\text{IT}}(t))$ where $\mathcal{F}$ is a function that models the relationship between the boundary conditions and the POD coefficients, $\mathbf{T}_{\text{in}}$, $\mathbf{f}_{\text{in}}$, $\mathbf{P}_{\text{IT}}$, and $\mathbf{f}_{\text{IT}}$ are the vectors of the CRAC supply temperatures, CRAC mass flow rates, IT powers, and IT mass flow rates, respectively. Similarly, we assume that the total IT-generated heat equals the total sensible heat load. In practice, $\mathcal{F}$ can be modeled using heat flux matching [38] or spline interpolation [39]. To apply POD for action rectification, we will design two forms of $\mathcal{F}$ and evaluate their performance in meeting the safety constraints. For this scenario, we will consider the detailed temperature distribution of a DC and focus on specific spatial locations for safety evaluation.

3.1.3 DC cooling control. As discussed in §1, to maintain $T_{\text{z}}(t)$ at a setpoint, the DC cooling control periodically adjusts the setpoints for $f_{\text{in}}(t)$ and $T_{\text{in}}(t)$. Let τ denote the control period. A typical setting for τ is 15 minutes [46]. Let $\hat{f}[k]$ and $\hat{T}_{\text{in}}[k]$ denote the setpoints applied at $t = k\tau$ for the k-th control period of $t \in (k\tau, (k+1)\tau)$. The cooling system implements $\hat{f}_{\text{in}}[k]$ and $\hat{T}_{\text{in}}[k]$ via the primary controls of its components. Due to the uncertain evolution of $P_{\text{IT}}(t)$, 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. $P_{\text{IT}}(t)$ only changes at the start of the control period and $P_{\text{IT}}[k] \triangleq P_{\text{IT}}(t)|_{t\in((k-1)\tau,k\tau)}$ 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 [33]. Under Assumption 2, the setpoints $\hat{f}_{\text{in}}[k-1]$ and $\hat{T}_{\text{in}}[k-1]$ are implemented when $t \to k\tau^-$. Formally, $f_{\text{in}}(t)|_{t\to k\tau^-} = \hat{f}_{\text{in}}[k-1]$, $T_{\text{in}}(t)|_{t\to k\tau^-} = \hat{T}_{\text{in}}[k-1]$, $\left.\frac{\mathrm{d}T_{\text{z}}(t)}{\mathrm{d}t}\right|_{t\to k\tau^-} = 0$. By substituting the above simplification-induced results into Eq. (1) and by defining $T_{\text{z}}[k] \triangleq T_{\text{z}}(t)|_{t\to k\tau^-}$, we obtain the following steady-state transition of the nodal dynamics model of a data hall by:

$$T_{\text{z}}[k] = \hat{T}_{\text{in}}[k-1] + \frac{\rho P_{\text{IT}}[k]}{\alpha \hat{f}_{\text{in}}[k-1]}. \tag{3}$$

Under the same assumptions, the steady state transition of the fine-grained POD model is obtained by:

$$\mathbf{T}_{\text{z}}[k] = \overline{\mathbf{T}}_{\text{o}} + \Phi\mathcal{F}\left(\hat{\mathbf{T}}_{\text{in}}[k-1], \hat{\mathbf{f}}_{\text{in}}[k-1], \mathbf{P}_{\text{IT}}[k], \mathbf{f}_{\text{IT}}[k]\right). \tag{4}$$

3.2 Deep Reinforcement Learning

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

In this paper, we use the DDPG [25] learning algorithm to deal with the continuous action space in DC cooling control. It concurrently learns $\mu_\theta(\mathbf{s})$ and a Q-function $Q_\psi(\mathbf{s}, \boldsymbol{\mu})$ parameterized with parameters ψ and differentiable with respect to action $\boldsymbol{\mu}$. 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 ψ by minimizing the loss function $\mathcal{L}(\psi) = \frac{1}{N}\sum_{i=1}^{N}\left(Q_\psi(s_i, \mu_i) - y_i\right)^2$, where y_i is the target Q value given by $y_i = r_i + \gamma Q'_\psi\left(s_{i+1}, \mu'_\theta(s_{i+1})\right)$. The Q'_ψ and μ'_θ are two target networks copied from the original networks and updated once per main network update. To learn the policy function, it updates θ by maximizing $\mathcal{J}(\theta) = \frac{1}{N}\sum_{i=1}^{N} Q_\psi(s_i, \mu_\theta(s_i))$.

4 PERFORMANCE BENCHMARK AND MOTIVATIONS

This section formulates the DC cooling control as an MDP problem with reward shaping for thermal safety considerations. Then, we benchmark a DDPG's performance on energy savings in comparison with a conventional controller. We also evaluate the effectiveness of reward shaping in thermal unsafety prevention during learning. The results motivate the pursuit of better solutions in §5.

4.1 MDP Formulation for DC Cooling Control

We consider a DC hosting l CRACs, m servers, and n temperature sensors deployed in the cold and hot aisles, respectively. The IT workload and outdoor environment temperature are two exogenous factors to DC cooling control. Let $P_{\text{IT}}[k] = \sum_i^m P_i[k]$ and $T_{\text{w}}[k]$ denote the total IT workload and outdoor weather temperature at $t = k\tau$, respectively. We assume both $P_{\text{IT}}[k]$ and $T_{\text{w}}[k]$ are Markovian. The zone air temperature at specific locations should be kept within a thermal safety upper bound denoted by $\overline{T}_{\text{z}}$, i.e., $T_{\text{z}_i}[k] \le \overline{T}_{\text{z}}, \forall k, i = 1, 2, \ldots, n$. In the simulations conducted in

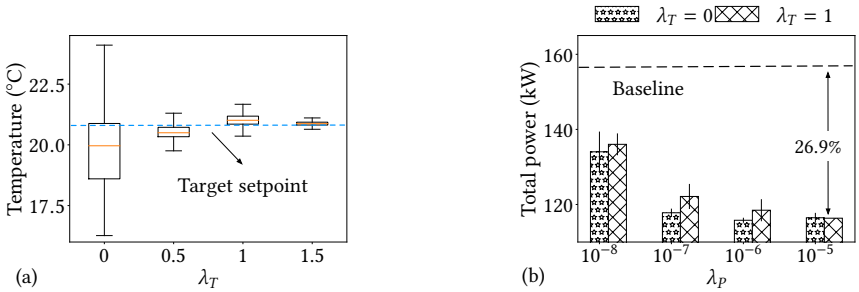

Fig. 2. Impact of λ_{T} and λ_{P} on performance of DDPG over 1-year test. (a) Data hall zone temperature; (b) DC average total power usage, with error bar representing the standard deviation over multiple DDPG agents.

this section, we set $\overline{T}_{\mathrm{z}} = 32$°C. With the above considerations, we define the control action, system state, and reward of the MDP as follows.

Control action: The action applied in the k-th control period, denoted by $\boldsymbol{\mu}[k]$, consists of the setpoints of the l CRACs' supply air temperature and mass flow rate. Formally, the control action is a vector defined as $\boldsymbol{\mu}[k] = (\hat{T}_{\mathrm{in}_1}[k], \hat{f}_{\mathrm{in}_1}[k], \hat{T}_{\mathrm{in}_2}[k], \hat{f}_{\mathrm{in}_2}[k], \ldots, \hat{T}_{\mathrm{in}_l}[k], \hat{f}_{\mathrm{in}_l}[k]) \in \mathbb{R}^{2l}$.

System State: The state at the k-th time step consists of the indoor/outdoor temperature measurements and the total power usage of the cooling and IT systems, respectively. Besides the notation defined in §3.1, we also define $P_{\mathrm{c}}[k] \triangleq P_{\mathrm{c}}(t)|_{t\to k\tau^-}$. Formally, the system state is a vector defined as $\mathbf{s}[k] = (T_{\mathrm{z}_1}[k], T_{\mathrm{z}_2}[k], \ldots, T_{\mathrm{z}_n}[k], P_{\mathrm{c}}[k], P_{\mathrm{IT}}[k], T_{\mathrm{w}}[k]) \in \mathbb{R}^{n+3}$. When the action $\boldsymbol{\mu}[k]$ is to be chosen at $t = k\tau$, $\mathbf{s}[k]$ is fully observable. From the assumption that the two exogenous state components $P_{\mathrm{IT}}[k]$ and $T_{\mathrm{w}}[k]$ 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 on the probability distributions of $\mathbf{s}[k]$ and $\boldsymbol{\mu}[k]$ only. Thus, the control process is an MDP.

Reward function: According to [17], a good DC cooling controller should maintain the average data hall air temperature at a certain setpoint denoted by T_{C} and reduce total energy usage. We adopt the following reward function that incorporates the above two goals and also includes a penalty term for thermal safety considerations:

$$r(\mathbf{s}[k]) = \underbrace{\lambda_{\mathrm{T}}\mathcal{G}\left(\mathbf{T}_{\mathrm{z}}[k], T_{\mathrm{C}}\right) - \lambda_{\mathrm{P}}P_{\mathrm{DC}}[k]}_{\text{optimization goals}} + \underbrace{\lambda_{\mathrm{S}}\mathcal{T}\left(\mathbf{T}_{\mathrm{z}}[k], T_{\mathrm{U}}, T_{\mathrm{L}}\right)}_{\text{penalty term}}, \tag{5}$$

where $\mathcal{G}$ is a Gaussian function defined as $\mathcal{G} = \sum_i^n \exp\left(-\lambda_1\left(T_{\mathrm{z}_i}[k] - T_{\mathrm{C}_i}\right)^2\right)$, $\mathcal{T}$ is a trapezoid penalty function defined as $\mathcal{T} = \sum_i^n \left(\left[T_{\mathrm{z}_i}[k] - T_{\mathrm{U}}\right]^+ + \left[T_{\mathrm{L}} - T_{\mathrm{z}_i}[k]\right]^+\right)$, λ_{T}, λ_{P}, λ_{S} and λ_1 are several hyperparameters, $P_{\mathrm{DC}}[k]$ is the DC's total power usage (i.e., $P_{\mathrm{DC}}[k] = P_{\mathrm{IT}}[k] + P_{\mathrm{c}}[k]$), $[T_{\mathrm{L}}, T_{\mathrm{U}}]$ specifies a desirable range for $T_{\mathrm{z}}[k]$ and $[x]^+ = \max\{0, x\}$. The Gaussian function regulates the average zone air temperature to be close to the target setpoint and the penalty term penalizes the reward when the temperature is out of $[T_{\mathrm{L}}, T_{\mathrm{U}}]$. The T_{U} can be set lower than $\overline{T}_{\mathrm{z}}$ to better address the thermal safety considerations in practice. The objective of the MDP problem is to find the policy parameters to maximize the long-term accumulative reward, i.e., $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{P_{\mathrm{IT}}, T_{\mathrm{w}}}\left[\sum_{k=0}^{\infty}\gamma^k r[k] \middle| \boldsymbol{\mu}_{\boldsymbol{\theta}}\right]$, where P_{IT} and T_{w} are two stochastic processes.

4.2 Performance Measurements

In this section, we adopt the nodal dynamics to model a single-hall DC and conduct a set of simulations in EnergyPlus to evaluate the performance of the DDPG solution. We implement DDPG

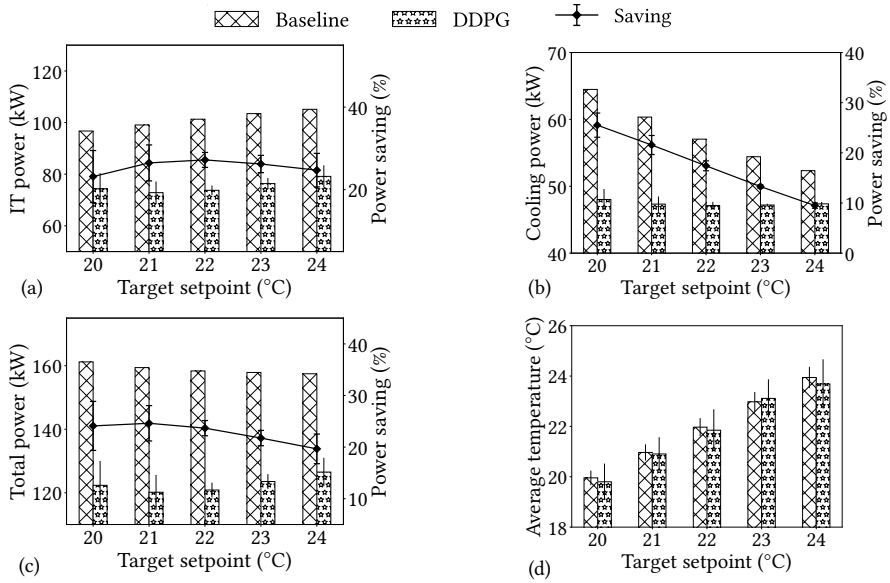

Fig. 3. Comparison between EnergyPlus' built-in controller (baseline) and converged DDPG over 1-year testing. (a)-(c) IT, cooling and total power consumption; (d) average zone air temperature.

in PyTorch [34] and integrate the EnergyPlus 8.8.0 simulator with the OpenAI gym [9] 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. 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: $U_{\mathrm{IT}} = 0.5$ from 00:00 to 06:00; $U_{\mathrm{IT}} = 0.75$ from 06:00 to 08:00; $U_{\mathrm{IT}} = 1.0$ from 08:00 to 18:00; $U_{\mathrm{IT}} = 0.8$ 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 [3] (referred to as *baseline controller*) that only aims at maintaining $T_{\mathrm{z}}[k]$ at T_{C} by adjusting the supply air temperature.

4.2.1 Impact of λ_{T} and λ_{P}. In Eq. (5), 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_{S}=0.1, T_{C}=21°C, $T_{\mathrm{L}} = T_{\mathrm{C}} - 1.5$°C, $T_{\mathrm{U}} = T_{\mathrm{C}} + 1.5$°C) and vary λ_{T} and λ_{P}. Fig. 2(a) shows the distribution of T_{z} versus λ_{T} when $\lambda_{\mathrm{P}} = 10^{-5}$. We train a separate DDPG agent for each λ_{T} setting. Each error bar shows the distribution of T_{z} during testing. When $\lambda_{\mathrm{T}} \neq 0$, the T_{z} fluctuates around T_{C} and the variation of T_{z} decreases with λ_{T}. When $\lambda_{\mathrm{T}} = 0$, 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 $\lambda_{\mathrm{T}} = 0$ leads to lower P_{DC} compared with the setting $\lambda_{\mathrm{T}} = 1$. This is because the DDPG agent with $\lambda_{\mathrm{T}} = 0$ 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 $\lambda_{\mathrm{T}} = 1$ and $\lambda_{\mathrm{P}} = 10^{-5}$.

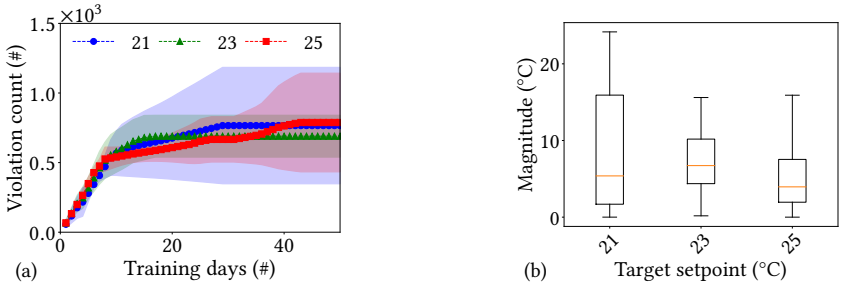

Fig. 4. DDPG's training phase under various setpoints. (a) Cumulative count of safety violations; (b) violation magnitude: midline, box, and whisker represent the median, interquartile range, and dispersion degree.

4.2.2 Comparison of DDPG and baseline controllers under various T_C settings. The 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 slightly 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. (3). As such, the IT power is lower since the server fans rotate slower.

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 [43].

Under a certain T_C setting, the above results show that 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 [17], 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 $T_z[k] \leq \overline{T}_z$. Specifically, in the k-th 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 $\left[T_z - \overline{T}_z\right]^+$. 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

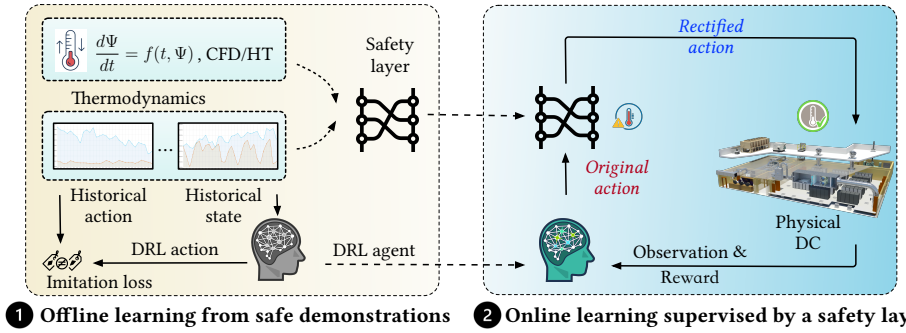

Fig. 5. The Safari framework consists of two stages. In the offline stage, the agent initializes its policy by learning from the demonstrations of an existing safe controller. Meanwhile, the historical data is also used to fit a thermal transition model that captures the knowledge of thermodynamics. In the online stage, the agent interacts with the physical DC to improve its policy supervised by the physics-based transition model. This model is adopted as a safety layer that continuously rectifies the potential unsafe DRL recommendations.

setting in the learning phase; the shaded area in the same color shows the corresponding standard deviation. We can see significant increases in 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 $\overline{T}_{\mathrm{z}}$. 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

This section formulates the constrained optimization problem and proposes the Safari framework to address the CMDP problem. Then, four variants of Safari with different transition models and prior knowledge are presented for online action rectification.

5.1 CMDP Formulation & Approach Overview

From the results in §4.2, DDPG achieves energy savings. However, as reward shaping addresses the thermal constraints 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 to explicitly enforce the thermal safety constraints of the following CMDP problem:

$$\begin{aligned} \theta^* \triangleq & \arg\max_{\theta} \mathbb{E}_{P_{\mathrm{IT}}, T_{\mathrm{w}}} \left[\sum_{k=0}^{\infty} \gamma^k r[k] \middle| \mu_\theta \right], \\ \text{s.t. } & \Pr\left(T_{\mathrm{z}_i}[k] \le \overline{T}_{\mathrm{z}_i} \right) > 1 - \epsilon, \ \forall k, \ i = 1, 2, \ldots, n, \end{aligned} \tag{6}$$

where ϵ is a small enough number for high confidence in ensuring the thermal safety requirement. Note that the constraints in Eq. (6) are expressed in the probabilistic form in general because $T_{\mathrm{z}_i}[k]$ is stochastic due to the stochasticity of $P_{\mathrm{IT}}[k-1]$.

Fig. 5 illustrated the proposed Safari framework. The proposed approach consists of the following two stages that aim to address the CMDP problem in Eq. (6).

■ *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.

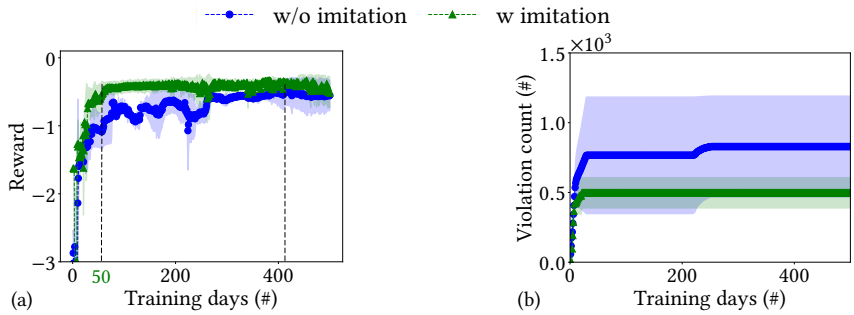

Fig. 6. Effectiveness of imitation learning. (a) Per-day reward average; (b) cumulative count of safety violations.

Meanwhile, these traces are also used to fit a state transition model (e.g., Eq. (3)) that will be used for the online stage. With imitation learning, the DDPG agent produces much fewer 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 constraints in Eq. (6), 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 $\boldsymbol{\mu}_{\boldsymbol{\theta}}[k]$ at the end of the control period. Let $\widetilde{T}_{z_i}[k+1]$ denote the i-th predicted temperature by the state transition model as $\widetilde{T}_{z_i}[k+1] = h_i(\boldsymbol{\mu}[k], P_{\mathrm{IT}}[k], \cdots)$, where "$\cdots$" represents the other factors that the prediction needs to consider. To ensure the system constraints, we solve the following problem to find the minimal rectified action $\boldsymbol{\mu}^*[k]$:

$$\begin{aligned} \boldsymbol{\mu}^*[k] \triangleq & \arg\min_{\boldsymbol{\mu}'[k]} \|\boldsymbol{\mu}'[k] - \boldsymbol{\mu}_{\boldsymbol{\theta}}[k]\|_2^2 /2, \\ & \text{s.t. } h_i(\boldsymbol{\mu}'[k], P_{\mathrm{IT}}[k], \cdots) \le \overline{T}_{z_i},\ i = 1, 2, \ldots, n. \end{aligned} \tag{7}$$

The ℓ_2 norm minimization in Eq. (7) 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. We derive the optimal rectification of Eq. (7) using the Karush-Kuhn-Tucker (KKT) conditions as:

$$\begin{cases} \boldsymbol{\mu}^*[k] - \boldsymbol{\mu}_{\boldsymbol{\theta}}[k] + \sum_{i=1}^{n} \lambda_i^* \nabla_{\boldsymbol{\mu}} h_i(\boldsymbol{\mu}^*[k], P_{\mathrm{IT}}[k], \cdots) = \mathbf{0}, \\ \lambda_i^* h_i(\boldsymbol{\mu}^*[k], P_{\mathrm{IT}}[k], \cdots) = 0, \\ \lambda_i^* \ge 0,\ i = 1, 2, \ldots, n, \end{cases} \tag{8}$$

where λ_i is the optimal Lagrange multiplier in terms of the i-th temperature constraint. With the above conditions, the rectified actions can be derived as $\boldsymbol{\mu}^*[k] = \boldsymbol{\mu}_{\boldsymbol{\theta}}[k] - \lambda_{i^*}^* \nabla_{\boldsymbol{\mu}} h_{i^*}(\boldsymbol{\mu}^*[k], P_{\mathrm{IT}}[k], \cdots)$, where $i^* \triangleq \arg\max_i \lambda_i^*$. The existing studies on post-hoc rectification [12, 15] adopt the linear state transition models such that the problem in Eq. (7) is a tractable convex quadratic program. Unfortunately, the thermal state transition in DC is nonlinear and non-differentiable in terms of $\boldsymbol{\mu}$. To address the challenges, §5.3 will present various surrogate 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 K consecutive control periods: $\{\mathbf{s}_{\mathrm{safe}}[k], \boldsymbol{\mu}_{\mathrm{safe}}[k] \mid k = 1, \ldots, K\}$, where $\boldsymbol{\mu}_{\mathrm{safe}}[k]$ is the action performed by the conventional safe controller on the state $\mathbf{s}_{\mathrm{safe}}[k]$ in the k-th control period. Such a dataset can be retrieved from the DCIM. The

DDPG agent's parameters θ is trained using the dataset to minimize the following loss function: $\mathcal{L}_{\text{imit}}(\theta) = \frac{1}{K}\sum_{k=0}^{K} \|\mu_\theta(\mathbf{s}_{\text{safe}}[k]) - \mu_{\text{safe}}[k]\|_2^2$. 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. (5). Figs. 6(a) and (b) show the traces of reward and cumulative safety violation count of the two groups of DDPG agents, respectively. From Fig. 6(a), the average reward of the agents with imitation learning converges after around 50 days of online training. For another group without imitation learning, it takes a longer time, i.e., around 400 days, to reach a similar performance. From Fig. 6(b), imitation learning can also reduce the cumulative violation count since these agents behave like the conventional safe controller at the start of online training. In summary, imitation learning accelerates DRL convergence and alleviates the safety concern of DRL. §5.3 will further develop online post-hoc rectification approaches aiming at eliminating safety violations during DDPG exploration.

5.3 Online Post-hoc Rectification

As discussed in §5.1, the accuracy of the state transition model $h(\mu[k], P_{\text{IT}}[k], \cdots)$ 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 four designs of Safari, i.e., Safari-1, Safari-2, Safari-3, and Safari-4 with different transition models that progressively integrate more prior knowledge and run-time information. Safari-1, -2, and -3 are designed for DCs with uniform temperature distributions, while Safari-4 considers the detailed spatial temperature modeling and can be applied for DCs with non-uniform temperature distributions. Specifically, Safari-1 uses the steady state transition model in Eq. (3). 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. Safari-4 uses the POD-based transition model in Eq. (4) for fine-grained thermal safety considerations.

5.3.1 A pure data-driven design of LSTM-based rectification. LSTM networks can model complex and nonlinear temporal correlations with satisfied accuracy [46]. 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 for training data. Fig. 7 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 increased 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. 8(a). The LSTM trained with the random exploratory data achieves 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. 8(a) 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

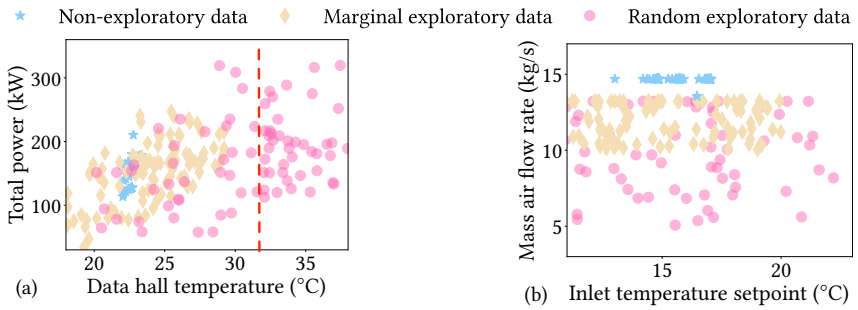

Fig. 7. Non-exploratory data, random exploratory data, and marginally safe exploratory data. (a) System state; (b) Control action.

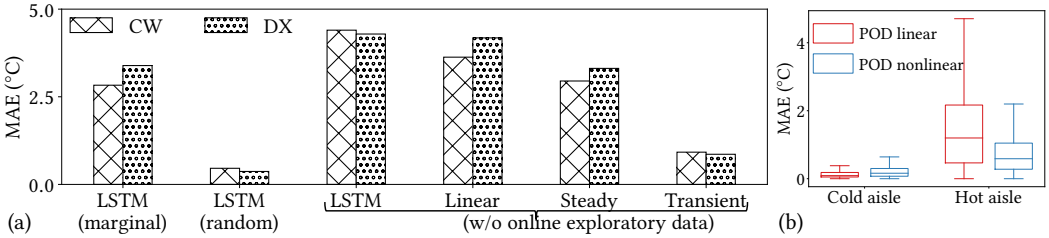

Fig. 8. Test MAE of different state transition models. The test data are randomly sampled, including the unsafe state. (a) MAE of zone air temperature for DCs with uniform temperature distributions; (b) MAE distributions of cold and hot aisles of a CW-cooled DC with non-uniform temperature distribution.

safety. This motivates us to explore transition models that incorporate physical knowledge and require less exploratory data to fit.

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. (3). Then, Safari-1 uses Eq. (3) as the prediction model for one data hall, i.e., $\widetilde{T}_z[k+1] = h(\boldsymbol{\mu}[k], P_{IT}[k], \cdots)$. If $\widetilde{T}_z[k+1]$ exceeds $\overline{T}_z$, Safari-1 solves the optimization problem in Eq. (7). For this scenario, the optimization problem of Eq. (7) is not disciplined quasiconvex programming and thus cannot be directly solved by existing convex optimization tools. To derive the optimal solution, we solve the following equation system derived from its KKT conditions:

$$\begin{cases} \hat{T}^*_{in}[k] - \hat{T}_{in}[k] + \lambda^* = 0, \\ \hat{f}^*[k] - \hat{f}[k] - \lambda^* \dfrac{P_{IT}[k+1]}{\alpha(\hat{f}^*[k])^2} = 0, \\ \lambda^* \left(\hat{T}^*_{in}[k] + \dfrac{P_{IT}[k+1]}{\alpha \hat{f}^*[k]} - \overline{T}_z \right) = 0, \end{cases} \tag{9}$$

where λ is the Lagrange multiplier, $\boldsymbol{\mu}^*[k] = (\hat{T}^*_{in}[k], \hat{f}^*[k])$ is the rectified action. Under the definition $P_{IT}[k+1] \triangleq P_{IT}(t)|_{t \in (k\tau,(k+1)\tau)}$, $P_{IT}[k+1]$ is unknown when the DDPG agent chooses the action at $t = k\tau$. However, pragmatically, the controller can wait for a short while until $P_{IT}[k+1]$ is observable and then solve Eq. (9).

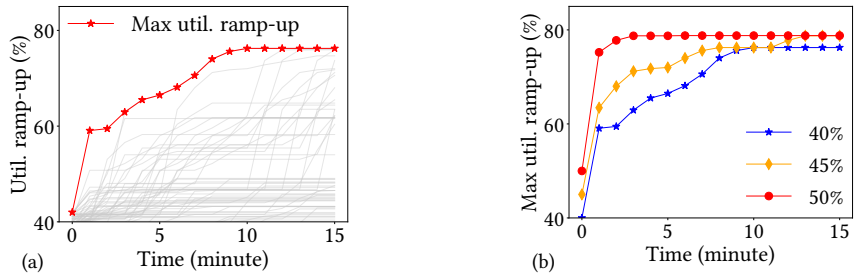

Fig. 9. (a) IT utilization ramp-ups (gray curves) and the max ramp-up (red curve) when starting utilization is 40%; (b) max ramp-ups when starting IT utilization is 40%, 45%, and 50%, respectively.

If the state evolution strictly follows the steady state transition in Eq. (3), the solution to Eq. (9) 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. (9). 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 $T_z[k+1]$ 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 $T_z(t)$ depends on the trajectories of $Q(t)$, $T_{in}(t)$, and $f(t)$. From Assumption 1, the $Q(t)|_{t\in[k\tau,(k+1)\tau)}$ remains constant at $P_{IT}[k+1]$. For $T_{in}(t)$ and $f(t)$, we adopt their setpoints as their approximations. Specifically, we set $T_{in}(t)|_{t\in[k\tau,(k+1)\tau)} = \hat{T}_{in}[k]$ and $f(t)|_{t\in[k\tau,(k+1)\tau)} = \hat{f}[k]$. Then, with the initial condition $T_z(k\tau) = T_z[k]$, we can solve $T_z(t)$ from Eq. (1) as $T_z(t) = W[k] + (T_z[k] - W[k])\,\mathrm{e}^{-\hat{f}[k](t-k\tau)/V_s}$, $t \in [k\tau, (k+1)\tau)$, where $W[k] = \hat{T}_{in}[k] + \frac{P_{IT}[k+1]}{\alpha\hat{f}[k]}$ is a constant within the k-th control period. Then, we mitigate the impact of making approximations for $T_{in}(t)$ and $f(t)$ by adopting the average of $T_z(t)$ as the prediction, i.e., $\widetilde{T}_z[k+1] = \frac{1}{\tau}\int_{k\tau}^{(k+1)\tau} T_z(t)\mathrm{d}t$.

Safari-2 uses the above heuristic prediction approach to predict $\widetilde{T}_z[k+1]$ for the action $\boldsymbol{\mu}$ recommended by DDPG. If $\widetilde{T}_z[k+1]$ exceeds $\overline{T}_z$, it applies grid search in the two-dimensional action space to solve the problem in Eq. (7), in which $h(\boldsymbol{\mu}', P_{IT}[k], \cdots)$ 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 $T_z(t)$. Safari-3's prediction algorithm is as follows. First, during offline stage, Safari-3 builds the *maximum ramp-up function* $P^{\nearrow}(\Delta t|P_{IT}^{start})$ 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 the length of τ minutes provided that the starting IT power is P_{IT}^{start}. Then, at $t = k\tau$, Safari-3 adopts $T_{in}(t) = \hat{T}_{in}[k]$, $f(t) = \hat{f}[k]$, and $Q(t) = P^{\nearrow}(t - k\tau|P_{IT}[k])$ to solve $T_z(t)$ from Eq. (1), where $t \in (k\tau, (k+1)\tau]$. Since Safari-3 uses the maximum ramp-up observed in history, the predicted $Q(t)$ is conservatively high, which is beneficial to unsafety prevention.

In Fig. 9(a), the gray curves show the aggregated IT utilization ramp-ups in a historical trace collected in a real DC (cf. Fig. 10(b)) when the starting IT utilization is 40%. The upper envelope of these curves is the maximum ramp-up. Fig. 9(b) shows the maximum ramp-ups when the starting IT utilization is 40%, 45%, and 50%.

5.3.5 Safari-4: POD-based rectification. To predict the temperature field $\widetilde{\mathbf{T}}_z[k+1]$ with fine-grained spatial considerations, we adopt the POD model from Eq. (4). To apply the POD model for rectification, we need to find the boundary-specific POD coefficients $\boldsymbol{\beta}$ such that the error between the POD-predicted temperature and the original calibrated CFD/HT output is minimized as $\boldsymbol{\beta}^* \triangleq \arg\min_{\boldsymbol{\beta}} \left\| \boldsymbol{\Phi}\boldsymbol{\beta} - \mathbf{T}_z^{\text{CFD}} \right\|_2^2$, where $\mathbf{T}_z^{\text{CFD}}$ is the calibrated CFD/HT simulation results. In this section, we develop two forms of the POD model to solve the problem of Eq. (7), respectively.

The first form, termed Safari-4.1, aims to derive a closed-form rectification from the KKT conditions of Eq. (8). In this form, we consider a linear function to model the relationship $\mathcal{F}$ in Eq. (4) between the boundary conditions and the POD coefficients. Specifically, $\mathcal{F}(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{W}_1^\intercal \mathbf{x}[k] + \mathbf{W}_2^\intercal \boldsymbol{\mu}[k]$, where $\mathbf{W}_1 \in \mathbb{R}^{2\times H}$ and $\mathbf{W}_2 \in \mathbb{R}^{2I\times H}$ are matrices of trainable weights, $\mathbf{x}$ consists of the boundary conditions of total IT heat load and flow rate, i.e., $\mathbf{x} = (P_{\text{IT}}, f_{\text{IT}})$. With this form of POD model, the temperature transition is modeled as $\widetilde{\mathbf{T}}_z[k+1] = \overline{\mathbf{T}}_o + \boldsymbol{\Phi}\left(\mathbf{W}_1^\intercal \mathbf{x}[k] + \mathbf{W}_2^\intercal \boldsymbol{\mu}[k]\right)$. Thus, the analytical rectification at the k-th control period is derived from Eq. (8) as:

$$\boldsymbol{\mu}^*[k] = \boldsymbol{\mu}_{\boldsymbol{\theta}}[k] - \lambda_{i^*}^* \mathbf{W}_2 \boldsymbol{\Phi}_{i^*}^\intercal, \tag{10}$$

where $\boldsymbol{\Phi}_i \in \mathbb{R}^{1\times H}$ is the i-th row of the POD modes matrix associated with the i-th temperature constraint and the corresponding Lagrange multiplier is derived as:

$$\lambda_i^* = \left[\frac{\overline{T}_{o_i} + \boldsymbol{\Phi}_i \left(\mathbf{W}_1^\intercal \mathbf{x}[k] + \mathbf{W}_2^\intercal \boldsymbol{\mu}[k]\right) - \overline{T}_{z_i}}{\left(\mathbf{W}_2 \boldsymbol{\Phi}_{i^*}^\intercal\right)^\intercal \left(\mathbf{W}_2 \boldsymbol{\Phi}_{i^*}^\intercal\right)} \right]^+, \ i = 1, 2, \ldots, n. \tag{11}$$

The detailed derivation can be found in the supplementary material.

The second form, termed Safari-4.2, aims to predict $\mathbf{T}_z[k+1]$ more accurately with nonlinear function to model $\mathcal{F}$ in Eq. (4). In this form, we adopt a two-layer MLP to learn the relationship between the boundary conditions and POD coefficients. The MLP consists of 32 and 64 neurons for each layer, respectively. With the nonlinear form of POD model, the temperature transition is modeled as $\widetilde{\mathbf{T}}_z[k+1] = \mathbf{T}_o + \boldsymbol{\Phi}\mathcal{F}(\mathbf{x}[k], \boldsymbol{\mu}[k])$. If the predicted temperature $\widetilde{\mathbf{T}}_z[k+1]$ exceeds $\overline{T}_z$, the rectification of Safari-4.2 can be numerically solved using the OptNet [5] or grid search for low dimensional action space.

5.3.6 Performance of state transition models and data requirements. We first evaluate the nodal form temperature transition models developed for Safari-1, -2 as well as a linear form used in [15]. Fig. 8 also shows the MAEs of these state prediction models when $P_{\text{IT}}(t)$ follows Assumption 1. The results are labeled "Steady", "Transient", and "Linear", respectively. The linear transition model uses the design of [15] to predict the next state temperature by $\widetilde{T}_z[k+1] = g_\omega(\mathbf{s}[k])^\intercal \boldsymbol{\mu}[k] + T_z[k]$ 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 [15]. We next evaluate the two forms of the POD model developed for Safari-4 for fine-grained temperature field predictions. The training data used to extract the POD models contains 96 samples generated from the calibrated CFD simulation. The test data contains 36 samples generated with different boundary conditions. We select the first five modes, i.e., $H = 5$, as the basis functions since they capture the majority of the energy of the temperature field. The results are labeled as "POD linear"

Table 3. Configurations of the evaluated DCs.

DC	Weather	Cooling system	No. of zones	Modeling methods
DC 1	Singapore Chicago	CW	1	EnergyPlus
DC 2	Singapore	DX	1	EnergyPlus
DC 3	Singapore	CW	2	EnergyPlus
DC 4	Singapore	CW	1	CFD/HT

Table 4. Hyperparameter settings

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

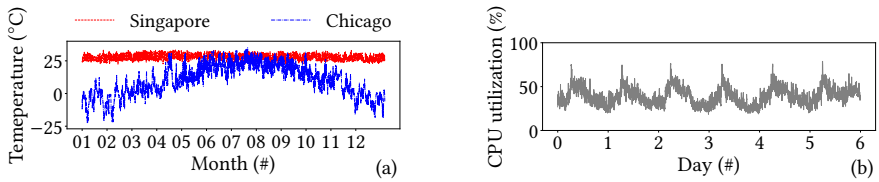

Fig. 10. (a) Historical weather data at Singapore and Chicago; (b) aggregated IT utilization trace in a real DC hosting 4,000 servers [2].

and "POD nonlinear" as shown in Fig. 8(b). The overall MAEs for the linear and nonlinear form of POD models are 0.54°C and 0.46°C, respectively. From the figure, we observe the linear form POD performs better for cold aisle temperature prediction while the nonlinear form produces lower MAE in predicting hot aisle temperature. As the considered DC is equipped with a hot aisle containment to prevent air re-circulation, the cold aisle temperature is linearly related to the CRAC settings. The temperature distribution in the hot aisle is more complex due to the heterogeneous heat load generated by the IT equipment. As such, the nonlinear form of the POD model performs better for hot aisle temperature prediction. Note the training data for extracting the POD modes are generated from the calibrated CFD model. Thus, the online exploratory data is not required. The performance of the various designs of Safari will be extensively evaluated in §6.

6 PERFORMANCE EVALUATION

This section applies the proposed Safari approach to optimize different DCs and presents the performance in energy saving and compliance with thermal safety constraints. We evaluate Safari on four DCs with different configurations. The DC configurations and hyperparameter settings of DDPG are summarized in Table 3 and 4, respectively.

6.1 Evaluation Methodology and Testbed Settings

6.1.1 EnergyPlus-based simulation testbed. We use EnergyPlus to simulate the physical processes of a CW-cooled and a DX-cooled DC with uniform temperature distribution, respectively. Figs. 10(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. 10(b) shows the aggregated CPU utilization trace collected from a real Internet DC hosting 4,000 servers [2]. 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:

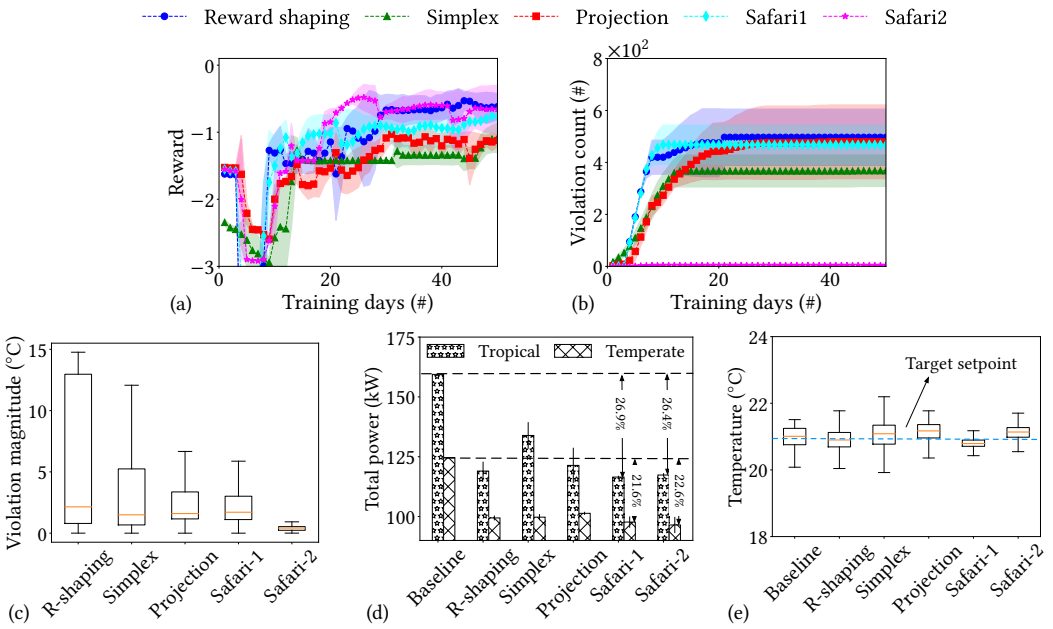

Fig. 11. Performance of various approaches on a CW-cooled DC with uniform temperature distribution. (a) Per-day reward average during learning; (b) cumulative count of safety violations during learning; (c) violation magnitudes during learning; (d) DC average total power usage and (e) zone air temperature distribution during 1-year testing.

■ **Baseline controller** is the EnergyPlus' built-in controller as described in §4 that only considers to maintain the temperature at the target setpoint [3].

■ **Reward shaping** refers to the DDPG agent presented in §4.1 that uses Eq. (5) as the shaped reward function. It captures the essence of [13, 24, 37, 42].

■ **Simplex** follows the essence of [27, 35]. 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 allowable minimum inlet temperature setpoint (i.e., 10°C) and maximum supply air flow rate (i.e., 15 kg/s).

■ **Projection** implements the post-hoc rectification with the linear transition model described in §5.3.6. It captures the essence of [12, 15] to solve a convex optimization problem of Eq. (7) with simplified system dynamics.

6.1.2 CFD/HT-based testbed. To simulate the detailed temperature distribution in the data hall, we employ a CFD/HT-based testbed that is built based on OpenFOAM [21]. The testbed captures a data hall that is equipped with one CRAC unit and two rows of racks hosting 299 servers as shown in Fig. 14(a). The rated power of each server is 1,000W and the server air flow rates are calibrated by the method in [44]. A hot aisle containment and rack blanking panels are installed to prevent air mixing caused by re-circulation. To monitor the temperature, 9 sensors are deployed at the cold and hot aisles, respectively. The fluid domain is discretized by OpenFOAM into 396,032 mesh grids. After the mesh is created, the NS and energy balance equations are solved by OpenFOAM to derive the airflow and temperature distributions. The air in the simulated data hall is assumed to be incompressible and the turbulence is modeled using the k-epsilon method [30]. The simulated data hall is equipped with a CW-cooled system based on the default settings adopted from §6.1.1. For this unmixed temperature scenario, the safety constraint should satisfy that the hot exhausted

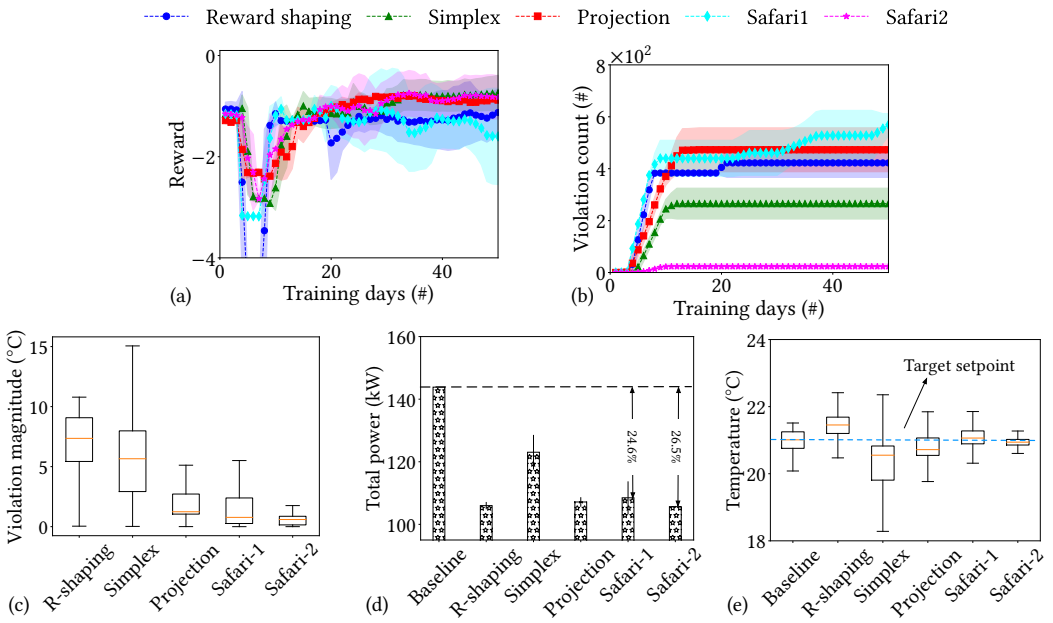

Fig. 12. Performance of various approaches on a DX-cooled DC under tropical climate.

air temperature should not exceed 20°C beyond the supply air temperature as suggested by [1]. On this testbed, we implemented the two forms of Safari-4 and the following baseline approaches for evaluation:

■ **PID controller** that aims to maintain the temperature difference at a target setpoint below the safety constraint. We set this target to 18°C. The parameters for the proportional, integral and derivative terms are $k_{\mathrm{p}} = 1.2$, $k_{\mathrm{i}} = 1$, $k_d = 0.05$, respectively [45].

■ **Vanilla DDPG** that only takes energy saving into the optimization objective without safety considerations.

■ **Reward shaping** that adds a penalty term to the reward function when the temperature difference exceeds 20°C [1].

6.2 Evaluation Results

6.2.1 Performance of Safari-1,-2 on a single-zone CW-cooled DC. We conduct simulations based on the IT utilization pattern described in §4.2, which satisfies Assumption 1. Fig. 11(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 of training. Fig. 11(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. 11(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. 11(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. (3) is better than the linear model in [15]. Safari-2 achieves the lowest violation count and magnitudes. Specifically, on the 50th day, the violation count of

Table 5. 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.

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. 11(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.

6.2.2 Performance of Safari-2,-3 with real-world IT utilization. Next, we conduct a set of simulations using a 6-day real IT utilization trace of 4,000 servers collected from a data center [2]. Fig. 10(b) shows the aggregated utilization trace. The trace is re-sampled with a 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 -3 when Assumption 1 is not strictly followed. Table 5 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.

6.2.3 Performance of Safari-1,-2 on a single-zone DX-cooled DC. In what follows, we conduct simulations in which the simulated IT power satisfies Assumption 1. Fig. 12 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 associated primary controls. From Fig. 12(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. On 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 26.5% average power savings compared with the baseline controller.

6.2.4 Performance of Safari-1,-2 on a two-zone CW-cooled DC. Next, we conduct evaluations on a two-hall CW-cooled DC with distinct temperature setpoints. Specifically, the actions and states of the two data halls are concatenated to form the action and state of the whole DC. The DC is under the control of a centralized DDPG agent. The target zone setpoints for the two halls are 21°C and 23°C, respectively. Fig. 13 shows the evaluation results. From Fig. 13, both Safari-1 and -2 reduce the magnitude of temperature violations compared with reward shaping. Safari-1 produces more violations in hall 1, indicating the steady state transition is weakened in hall 1. Safari-2 keeps sub-1°C 2nd quartile of violations for both data halls. With the centralized agent, Safari-1 and -2 both achieve 18% to 19% average power savings compared with the baseline as shown in Fig. 13(c).

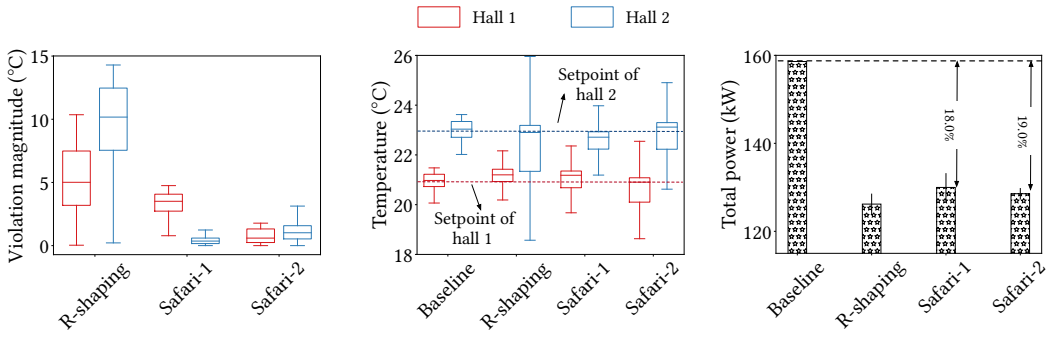

Fig. 13. Performance of Safari-1 and -2 on a two-hall CW-cooled DC.

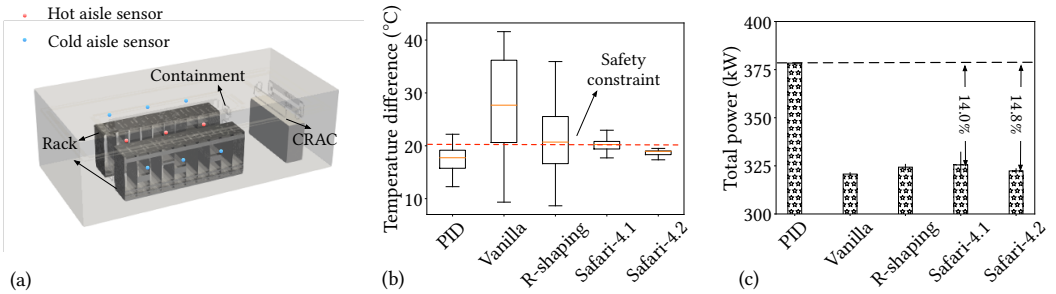

Fig. 14. Performance of Safari-4 on the CFD/HT-based testbed. (a) DC layout, (b) distribution of the maximal temperature difference between the hot and cold aisles during 1-month training, (c) total power during 1-month testing in the tropics.

However, from the results in Fig. 13(b), the temperature variance of each data hall is larger than the single-hall scheme. This implies the centralized DDPG control does not accommodate well to individual targets, which is in agreement with the findings in [23, 32].

6.2.5 Performance of Safari-4 on a single-zone CW-cooled DC. To extend Safari to address DCs with non-uniform temperature distributions, we conduct evaluations on the CFD/HT-based testbed as described in §6.1.2 based on the IT utilization pattern used in §4.2 and the tropical weather trace. Fig. 14(b) shows the distribution of temperature difference between the hot and cold aisles during 1-month training. We observe that Safari-4.1 is able to maintain sub-1°C 3rd quartile of violations and Safari-4.2 can almost keep the maximal temperature difference within 20°C. Safari-4.1 produces a few times violations due to the approximation error of the linear form POD model. Fig. 14(c) shows the DC total power usage under various control over one month. Although the vanilla DDPG achieves the highest saving, i.e., 15.2% compared with the PID control, it produces excessive training violations during the training. Safari-4 achieves about 14% power saving and successfully addresses the thermal safety compliance.

7 DISCUSSION

This section discusses two issues not addressed in this paper.

■ **Multi-agent control:** For the multi-hall scenario with distinct zone temperature setpoints, the DDPG algorithm can be extended to multi-agent control. The multi-agent control is expected

to achieve better temperature maintaining performance for each data hall. Specifically, each DDPG agent will contain a decentralized actor and a centralized critic, respectively. The decentralized actor will recommend action based on the state of the controlled data hall and the centralized critic will observe the global DC state. The Safari approach can be applied in each data hall independently.

■ **Eliminating thermal violations:** From the evaluation results, Safari-4 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-4 deployed, the activation of this standby CRAC unit is rare. Thus, the energy usage of this last line of defense is negligible.

8 CONCLUSION

This paper presents Safari, a safe DRL toward DC cooling control. By integrating imitation learning and post-hoc rectification designed based on the thermodynamics governing the heat process in the data hall, Safari can effectively prevent thermal unsafety. Our extensive evaluation that covers both CW and DX cooling systems under two climate conditions shows that, with varying IT workload patterns, Safari saves 18% to 26.9% total DC power compared with conventional control and reduces safety violations up to 99% compared with reward shaping. With the extended evaluation on a CW-cooled DC with non-uniform temperature distribution, Safari achieves 14% total power saving while ensuring the thermal safety constraint during training. Safari sheds light on the deployment of DRL algorithms to safety-critical cyber-physical systems.

REFERENCES

[1] 2017. How to monitor server room temperature and environmental conditions. https://www.enviromon.net/how-to-monitor-server-room-temperature/

[2] 2021. Alibaba cluster trace program. https://github.com/alibaba/clusterdata.

[3] 2021. EnergyPlus Setpoint Managers. https://bit.ly/3EtLmZp.

[4] 2021. Global Internet data centers market report. https://www.businesswire.com/news/home/20210903005160/en/

[5] B. Amos and J. Z. Kolter. 2017. OptNet: Differentiable optimization as a layer in neural networks. In *ICML*. 136–145.

[6] J.D. Anderson and J. Wendt. 1995. *Computational fluid dynamics*. Vol. 206. Springer.

[7] 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.

[8] ASHRAE. 2011. Thermal guidelines for data processing environments–expanded data center classes and usage guidance.

[9] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. 2016. OpenAI gym. *arXiv:1606.01540* (2016).

[10] Z. Cao, X. Zhou, H. Hu, Z. Wang, and Y. Wen. 2022. Towards a systematic survey for carbon neutral data centers. *IEEE Commun. Surv. Tutor.* 24 (2022), 895–936.

[11] 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.

[12] 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.

[13] 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*.

[14] D. Crawley, L. Lawrie, C. Pedersen, and F. Winkelmann. 2000. EnergyPlus: Energy simulation program. *ASHRAE J.* 42, 4 (2000).

[15] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa. 2018. Safe exploration in continuous action spaces. *arXiv:1801.08757* (2018).

[16] X. Ding, W. Du, and A. Cerpa. 2020. MB2C: Model-based deep reinforcement learning for multi-zone building control. In *ACM BuildSys*.
[17] 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.
[18] P. Geibel and F. Wysotzki. 2005. Risk-sensitive reinforcement learning applied to control under constraints. *J. Artificial Intelligence Research* 24 (2005).
[19] P. Holmes, J.L. Lumley, G. Berkooz, and C. Rowley. 2012. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press.
[20] A. Jain, T. Nghiem, M. Morari, and R. Mangharam. 2018. Learning and control using Gaussian processes. In *ACM/IEEE ICCPS*. 140–149.
[21] H. Jasak, A. Jemcov, and Z. Tukovic. 2007. OpenFOAM: A C++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, Vol. 1000. IUC Dubrovnik Croatia, 1–20.
[22] 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.
[23] J. Li, W. Zhang, G. Gao, Y. Wen, G. Jin, and G. Christopoulos. 2021. Toward intelligent multizone thermal control with multiagent deep reinforcement learning. *IEEE Internet Things J.* 8, 14 (2021), 11150–11162.
[24] 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.
[25] 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).
[26] Hsin-Yu Liu, Bharathan Balaji, Sicun Gao, Rajesh Gupta, and Dezhi Hong. 2022. Safe HVAC Control via Batch Reinforcement Learning. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 181–192.
[27] H. Mao, M. Schwarzkopf, H. He, and M. Alizadeh. 2019. Towards safe online reinforcement learning in computer systems. In *NeurIPS*.
[28] 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.
[29] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (2015), 529–533.
[30] B. Mohammadi and O. Pironneau. 1993. Analysis of the k-epsilon turbulence model. (1993).
[31] 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.
[32] S. Nagarathinam, V. Menon, A. Vasan, and A. Sivasubramaniam. 2020. Marco-multi-agent reinforcement learning based control of building HVAC systems. In *ACM e-Energy*. 57–67.
[33] K. Ogata. 1995. *Discrete-time control systems*. Prentice-Hall, Inc.
[34] A. Paszke, F. Gross, S.and Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison. 2019. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS* 32 (2019).
[35] 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.
[36] A. Radmehr, B. Noll, J. Fitzpatrick, and K. Karki. 2013. CFD modeling of an existing raised-floor data center. In *IEEE SEMI-THERM*. 39–44.
[37] 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.
[38] E. Samadiani and Y. Joshi. 2010. Proper orthogonal decomposition for reduced order thermal modeling of air cooled data centers. *J. Heat Transfer* 132, 7 (2010).
[39] E. Samadiani, Y. Joshi, H. Hamann, M.K. Iyengar, S. Kamalsy, and J. Lacey. 2012. Reduced order thermal modeling of data centers via distributed sensor data. *J. Heat Transfer* 134, 4 (2012).
[40] 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.
[41] 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.
[42] 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.
[43] 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.

[44] R. Wang, X. Zhou, L. Dong, Y. Wen, R. Tan, L. Chen, G. Wang, and F. Zeng. 2020. Kalibre: Knowledge-based neural surrogate model calibration for data center digital twins. In *ACM BuildSys*. 200–209.
[45] 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.
[46] 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 DERIVATION OF THE CLOSED-FORM SOLUTION FOR SAFARI-4.1

For Safari-4.1, we have the following optimization problem:

$$\begin{aligned} \boldsymbol{\mu}^*[k] \triangleq & \arg\min_{\boldsymbol{\mu}'[k]} \|\boldsymbol{\mu}'[k] - \boldsymbol{\mu}_{\boldsymbol{\theta}}[k]\|_2^2 /2, \\ & \text{s.t. } \overline{T}_{\mathrm{o}_i} + \boldsymbol{\Phi}_i \left(\mathbf{W}_1^\intercal \mathbf{x}[k] + \mathbf{W}_2^\intercal \boldsymbol{\mu}[k]\right) \leq \overline{T}_{\mathrm{z}_i},\ i = 1, 2, \ldots, n. \end{aligned} \tag{1}$$

Then, we write the Lagrangian of Eq. (1) as:

$$L\left(\boldsymbol{\mu}[k], \lambda\right) = \|\boldsymbol{\mu}'[k] - \boldsymbol{\mu}_{\boldsymbol{\theta}}[k]\|_2^2 /2 + \sum_{i=1}^{n} \lambda_i \left(\overline{T}_{\mathrm{o}_i} + \boldsymbol{\Phi}_i \left(\mathbf{W}_1^\intercal \mathbf{x}[k] + \mathbf{W}_2^\intercal \boldsymbol{\mu}[k]\right) - \overline{T}_{\mathrm{z}_i}\right). \tag{2}$$

As the objective and constraints in Eq. (1) are convex, the feasible solution should satisfy the KKT conditions. From the KKT conditions, we can get:

$$\nabla_{\boldsymbol{\mu}} L = \boldsymbol{\mu}^*[k] - \boldsymbol{\mu}_{\boldsymbol{\theta}}[k] + \sum_{i=1}^{n} \lambda_i^* \mathbf{W}_2 \boldsymbol{\Phi}_i^\intercal = \mathbf{0}, \tag{3}$$

$$\lambda_i^* \left(\overline{T}_{\mathrm{o}_i} + \boldsymbol{\Phi}_i \left(\mathbf{W}_1^\intercal \mathbf{x}[k] + \mathbf{W}_2^\intercal \boldsymbol{\mu}^*[k]\right) - \overline{T}_{\mathrm{z}_i}\right) = 0,\ i = 1, 2, \ldots, n. \tag{4}$$

Substituting Eq. (4) in Eq. (3), we get:

$$\lambda_i^* = \left[\frac{\overline{T}_{\mathrm{o}_i} + \boldsymbol{\Phi}_i \left(\mathbf{W}_1^\intercal \mathbf{x}[k] + \mathbf{W}_2^\intercal \boldsymbol{\mu}_{\boldsymbol{\theta}}[k]\right) - \overline{T}_{\mathrm{z}_i}}{\left(\mathbf{W}_2 \boldsymbol{\Phi}_{i^*}^\intercal\right)^\intercal \left(\mathbf{W}_2 \boldsymbol{\Phi}_{i^*}^\intercal\right)}\right]^+,\ i = 1, 2, \ldots, n. \tag{5}$$

If the constraints are satisfied, the Lagrange multiplier should be inactive, i.e., $\lambda_i^* = 0, i = 1, 2, \ldots, n$. If the constraints are not satisfied, the DRL recommended action will be rectified by $\boldsymbol{\mu}^*[k] = \boldsymbol{\mu}_{\boldsymbol{\theta}}[k] - \lambda_{i^*}^* \mathbf{W}_2 \boldsymbol{\Phi}_{i^*}^\intercal$, where $i^* \triangleq \arg\max_i \lambda_i^*$.

B A BRIEF INTRODUCTION OF THE DX COOLING SYSTEM

Different from the CW system that has three cycles, the DX system has two cycles only as shown in Fig. 1. 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.

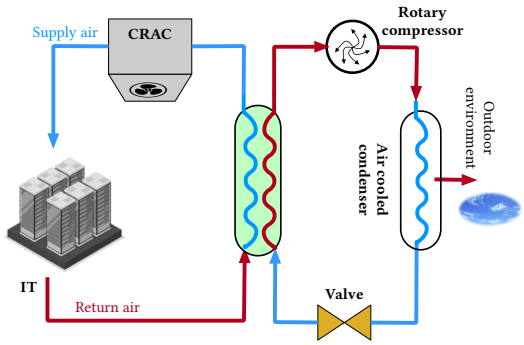

Fig. 1. DX-cooled DC.