Phyllis: Physics-Informed Lifelong Reinforcement Learning for Data Center Cooling Control

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

Zhiwei Cao
zhiwei003@ntu.edu.sg
Nanyang Technological University
Singapore

Xin Zhou
1020161201@jxstnu.edu.cn
Jiangxi Science and Technology
Normal University, China

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

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

ABSTRACT

Deep reinforcement learning (DRL) has shown good performance in data center cooling control for improving energy efficiency. The main challenge in deploying the DRL agent to real-world data centers is how to quickly adapt the agent to the ever-changing system with thermal safety compliance. Existing approaches rely on DRL's native fine-tuning or a learned data-driven dynamics model to assist the adaptation. However, they require long-term unsafe exploration before the agent or the model can capture a new environment. This paper proposes *Phyllis*, a physics-informed reinforcement learning approach to assist the DRL agent's lifelong learning under evolving data center environment. Phyllis first identifies a transition model to capture the data hall thermodynamics in the offline stage. When the environment changes in the online stage, Phyllis assists the adaptation by i) supervising safe data collection with the identified transition model, ii) fitting power usage and residual thermal models, iii) pretraining the agent by interacting with these models, and iv) deploying the agent for further fine-tuning. Phyllis uses known physical laws to inform the transition and power models for improving the extrapolation ability to unseen states. Extensive evaluation for two simulated data centers with different system changes shows that Phyllis saves 5.7% to 13.8% energy usage compared with feedback cooling control and adapts to new environments 8x to 10x faster than fine-tuning with at most 0.74°C temperature overshoot.

CCS CONCEPTS

• **Computing methodologies** → **Online learning settings**; *Reinforcement learning*; • **Hardware** → **Enterprise level and data centers power issues**.

KEYWORDS

Data centers, cooling control optimization, lifelong reinforcement learning, safe exploration, domain adaptation

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
e-Energy '23, June 20–23, 2023, Orlando, FL, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0032-3/23/06.
https://doi.org/10.1145/3575813.3595189

ACM Reference Format:
Ruihang Wang, Zhiwei Cao, Xin Zhou, Yonggang Wen, and Rui Tan. 2023. Phyllis: Physics-Informed Lifelong Reinforcement Learning for Data Center Cooling Control. In *The 14th ACM International Conference on Future Energy Systems (e-Energy '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3575813.3595189

1 INTRODUCTION

The global data center (DC) industry has been growing rapidly to support the Internet ecosystem. Such growth would require more electricity to operate the information technology (IT) devices and associated cooling systems. The IT devices consume electricity to provide computing and storage services under the required thermal conditions maintained by the cooling systems. Several reports have estimated that DC electricity usage will triple or even quadruple by 2030 [2, 3]. To mitigate such growth and attain energy sustainability, it is crucial to improve DC energy efficiency by developing and deploying novel technologies and solutions.

Perpendicular to introducing energy-efficient facilities, we consider cooling control optimization to improve DC energy efficiency. The optimization aims to reduce the long-term average energy usage subject to specified thermal conditions by periodically adjusting the air temperatures and mass flow rates supplied from the computer room air conditioning (CRAC) units. This problem can be modeled as a constrained Markov decision process (MDP) and solved with the deep reinforcement learning (DRL) techniques [19, 22, 31, 34]. Compared with the traditional feedback controllers (e.g., the proportional–integral–derivative (PID) controller [39]) that only maintain the temperature at the setpoint, the DRL-based solutions can optimize the expectation of a reward that jointly captures the time-averaged DC energy usage and the temperature deviation from the setpoint. Prior studies have shown that DRL can save 11% to 15% DC cooling cost while satisfying thermal constraints [19]. However, these studies assume that the DC environment remains unchanged over time, i.e., the agents are trained in advance and then deployed to a test environment same or similar to that for training. Unfortunately, this assumption may not be true for real DC operations where the environment may have significant changes.

In real-world DC operations, the system dynamics are changing over time due to facility upgrades. As shown in Figure 1, high-performance IT devices may need to be installed to meet new service level agreements, which can lead to an unexpected increase

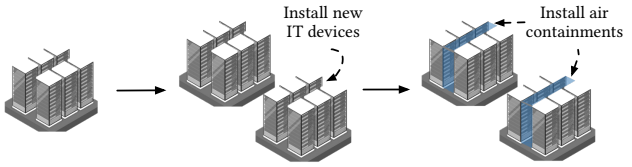

Fig. 1: DC upgrades over time with changing system dynamics. For example, new IT devices and air containments could be installed during the run time, which may affect the system power usage and indoor airflow distribution.

in IT power usage that is not covered in the DRL training. A DRL agent is observed with run-time performance degradation when the system configuration changes [26]. Such degradation is primarily caused by the shift of *state transition* probabilities and *reward* values given the same state-action pairs under different environment configurations. To address the degradation, transfer learning has been studied to apply the knowledge learned from the *source domains* to adapt the agent to a different but related *target domain* [46]. Under the reinforcement learning scheme, we refer to these domains as different MDPs and the transfer to different MDPs as a *lifelong learning* process. Prior studies have shown that policy transfer can be accelerated by relaxing the overly specialized source policies [40] or adding higher weights to the reward that captures important information of the target domain [11]. However, these approaches still require substantial exploratory data in the target domain to effectively implement the policy transfer. In the context of DC cooling control, collection of exploratory data from the environment poses thermal safety risks.

The implementation of lifelong reinforcement learning for DC operations faces two challenges. First, the adaptation to a new MDP should not lead to thermal unsafety. Second, the DRL agent should converge to the new optimal policy quickly for maintaining energy efficiency. Traditional model-free DRL agent tends to adapt slowly due to the need of extensive trial-and-error explorations [31]. Model-based reinforcement learning is an alternative method that offloads the learning process to a model of state transition and reward. Recent studies have shown that using a learned model to assist the training can improve learning efficiency and reduce the system's excursions to unsafe states [18]. The conventional model-based reinforcement learning methods build the model with black-box function approximators, such as Gaussian process [7] and neural network models [24]. However, these models are often data-demanding and poorly extrapolated to states unseen in the training process. As a result, the DRL agent still needs to experience (marginally) unsafe states before the model and agent can capture the new environment.

To overcome the limitations of the black-box function approximators, we propose to exploit the governing thermodynamics and cooling system laws as the prior knowledge to build the system models. Incorporation of prior knowledge into machine learning has been an interest of research due to its advantages in reducing data demand and improving extrapolation ability [16]. Physics-informed neural network (PINN) is a recent technique to embed the physical constraints into the training process of deep neural networks [30]. Several studies have attempted to extend the PINN to model-based optimal control [20, 25]. However, these studies assume both the state transition and reward values can be derived from the PINN given the current state-action pairs. For a DC, which is a complex multi-physics cyber-physical system, while the state transition can be characterized by the thermodynamics in the data hall, determining the reward requires the energy usage models of the cooling systems (e.g., the chiller plants). Unfortunately, such models are often not specified and can only be fitted using online exploratory data. To learn the system power usage model after the system changes, safe collection of exploratory data is necessary.

To address the above specific challenge, we propose *Phyllis* [1], a physics-informed lifelong reinforcement learning approach to assist the policy adaptation to the ever-changing DC environments with safety and speed considerations. Specifically, in the offline stage, Phyllis identifies a differentiable thermal transition model that adheres to the physical laws. In the online adaptation stage, Phyllis implements the following four steps to implement the adaptation. First, the identified transition model is used to supervise a short period of online exploration to safely collect data. Second, the collected online data is used to find the relationship between the control actions and cooling power usage. A residual model is also learned with the new online data to complement the previously identified transition model. Third, with the identified thermal state transition and reward functions, the policy is pre-trained by interacting with these models. Finally, we deploy the pre-trained policy to interact with the physical system for further fine-tuning. Phyllis draws the respective advantages of model-based reinforcement learning and physics-informed machine learning to address specific challenges faced by policy adaptation in DC. The incorporation of known physical laws reduces the data demand for system identification and helps better manage thermal safety when collecting online exploratory data. The main contributions of this paper are summarized as follows:

- We formulate the online adaptation as two associated problems that aim to search for the minimal action adjustment to enforce thermal constraints and maximize the forward transfer performance, respectively.
- We propose Phyllis that incorporates known physical laws to build system models and assist policy adaptation under changing DC environments. The incorporation of these laws improves the model extrapolation ability and better manages thermal safety.
- We conduct extensive evaluation on two simulated DCs with additions of IT devices and air containments, respectively. Evaluation results show that Phyllis saves 5.7% to 13.8% total power usage compared with traditional PID controller and accelerates the convergence speed by 8x to 10x compared with pure fine-tuning adaptation.

Paper roadmap: §2 reviews and categories the relevant studies. §3 presents the preliminary system models. §4 overviews the Phyllis approach design and associated problems. §5 illustrates the technical details of Phyllis. §6 evaluates Phyllis on two DCs. §7 discusses two issues and §8 concludes this paper.

[1] Phyllis means green leaf. We use this word to draw an analogy with green data centers and represent our physics-informed lifelong reinforcement learning approach.

Table 1: Categorization of relevant studies in cooling control optimization.

Environment	Approach	Study	Application	System modeling approach		Requirements for	
				Thermal transition	Power usage	System models	Safety
Stationary	model-free	[19, 22, 31]	DC cooling	Not required			Exploratory data
	model-based†	[18]		Linear model		Safe exploratory data	Action range
		[44]		Long short term memory networks (LSTM)		Exploratory data	
		[36]		Physics model	Not required	Thermodynamics & historical data	
Non-stationary*	model-free	[8, 43]	Building HVAC	Not required			Exploratory data
	model-based	[42]		Fully-connected neural network (FNN)		Historical & exploratory data	
		[27]		LSTM + FNN		Exploratory data	
		Phyllis	DC cooling	Hybrid models (PINN/POD + FNN)	Polynomial models	Thermodynamics & safe exploratory data	Thermodynamics

* "Non-stationary" means the state transition probabilities and reward values can be different given the same state-action pairs.
† The "model" refers to whether a model is introduced for direct DRL training interaction or assisting safe online exploration.

2 RELATED WORK

This section reviews the relevant studies in DRL-based cooling control. These approaches are categorized in Table 1 based on their types of environments and requirements for system models.

2.1 DRL-based DC Cooling Control

The DC cooling control can be viewed as an MDP and fitted into the DRL framework. Early studies adopt the *model-free* paradigm that allows the agent to learn by interacting with the system [19, 22, 31, 34]. Although these studies demonstrate substantial energy savings compared with conventional controllers, they suffer from high exploration risk and poor sample efficiency. For example, the model-free agent in [22] requires about 200,000 interaction steps to converge. The corresponding time for performing such many steps is 5.7 years, rendering the approach unrealistic. In addition, during the lengthy interaction, the thermal constraints in these studies are relaxed by following the reward shaping, which is only a palliative solution. When deploying DRL to DC operations, it is critical to address thermal safety and sample cost. Another group of studies adopts the *model-based* paradigm to improve the learning efficiency [18, 44]. However, the models used in these studies are either over-simplified [18] or data-intensive [44]. Recent studies propose to exploit the governing physics to assist the learning and rectify unsafe actions [36, 45]. The introduction of physical laws reduces the data demand for modeling. However, the models are expressed in the form of differential equations, and hence non-differentiable with respect to the control actions. As a result, the rectified actions can only be determined via heuristic search. The search process may fail to converge within a control period when the dimension of system variables is high. In contrast, we aim to develop a physics-informed differentiable model to efficiently solve the action search problem. Such a model shall ensure timely rectification and facilitate online usage. In summary, although existing studies have demonstrated remarkable performance for DRL-based DC cooling control, few of them focus on addressing the challenges when deploying the policy to non-stationary DC environments.

2.2 Transfer Learning in DRL

Transfer of a DRL agent to the changing MDPs is closely related to lifelong or continual reinforcement learning [17]. Previous studies on this topic aim to tackle the forgetting problem [32]. In this paper, we focus on optimizing the forward transfer performance with adaptation safety and speed considerations, which is more relevant to DC operations. To speed up transfer, previous studies adopt the pre-trained value or policy networks for fine-tuning [8, 43]. For example, [43] proposes to fine-tune the parameters of a sub-network in a new environment. While parameter transfer can reduce the convergence time compared with training from scratch, the re-learning process in [43] still requires weeks to converge and the system constraints are not explicitly addressed during the learning. Different from the parameter space transfer, we aim to learn system dynamics models and use them to assist the agent's transfer. A recent study [42] has shown that online learning can be accelerated by pre-training the agent offline with system models learned from historical data. However, the historical data collected from a DC running at a stable operating point are often non-exploratory and centered around a target setpoint under conventional feedback control. The model learned with such data may be overfitted and poorly extrapolated to unseen states. To capture the system changes, the dynamics model in [27] is continuously updated with incoming online data. However, the data-driven model needs to accumulate enough data to achieve satisfactory accuracy, which may require unsafe explorations. To address this issue, we develop an approach that captures both physical constraints and online data distribution to model the system dynamics.

2.3 Physics-Informed Learning and Control

Physical knowledge can be embedded into machine learning via observation data, model architecture and loss function [16]. To model the thermodynamics of the building environments, recent studies impose physical constraints on the neural networks' architecture [10] or loss function [13]. Another study [4] incorporates the energy balance principle to regulate the prediction of a DC thermal model based on the *proper orthogonal decomposition* (POD) and Gaussian process. While these studies have shown good prediction accuracy compared with black-box models, they do not evaluate the efficacy of the physics-informed models for optimal control and consider the system power usage. The latest study [25] extends the PINN to model the temperature and humidity in human-centric buildings. The PINN is then used to optimize the energy usage of the heating, ventilation, and air-conditioning (HVAC) systems using

Table 2: Summary of Notations

Symbol	Definition
V_s	volume of the data hall
C_p, ρ_{air}	specific heat capacity and density of air
n	number of temperature sensors
Q	heat load of the sensible and removed part
P_{IT}	IT power, which equals the sensible heat load
P_c	cooling power, $P_c = P_{crac} + P_{chp} + P_{ch} + P_{cp} + P_{ct}$
P_{ch}	chiller power usage
P_{crac}, P_{ct}	power usage of CRAC and cooling tower fans
P_{chp}, P_{cp}	power usage of chilled and condensed water pumps
U_{IT}	IT device utilization
T_s, T_z	air temperatures of supply and zone return
T_{in}	server inlet temperature, $T_{in} = (1-\alpha)\,T_s + \alpha T_z$
α	hot air recirculation ratio
$\hat{m}_s$	setpoint of supply air mass flow rate
$\hat{T}_s$, $\hat{T}_z$	setpoints of supply and zone air temperature
T_l, T_u	allowable temperature lower and upper limits
τ	time-slot length for a control period
$\mathbf{s}$	state, $\mathbf{s} = (T_s, T_z, P_{IT})$
$\boldsymbol{\mu}$	action, $\boldsymbol{\mu} = (\hat{T}_s, \hat{m}_s)$
R, C	reward and cost value for step transition
$\mathcal{S}$	safety set for temperature variations
$\mathcal{M}$	Markov decision process
$\mathcal{F}$, $\mathcal{F}_p$, $\mathcal{F}_d$	true dynamics, physics prior and residual models

model predictive control. While the study [25] shows the advantages of PINN for thermodynamics modeling, it is not designed to assist the DRL agent's transfer under changing environments where the models face more challenges in capturing system dynamics.

3 PRELIMINARY

This section presents the DC cooling control system models. We consider a typical enterprise single-hall DC equipped with a chilled water (CW) cooling system as shown in Figure 2. The cooling process consists of two stages. The first stage adopts the CRACs to supply cold air to the IT devices and cool the return hot air by the water-air heat exchangers. The second stage transfers the water-carried heat by the chiller and dissipates it to the ambient by the cooling tower. In this study, we focus on the data hall heat transfer and system power usage modeling. Table 2 summarizes the notations used in this paper.

3.1 Data Hall Thermodynamics

The heat transfer of the first stage can be characterized by the *computational fluid dynamics and heat transfer* (CFD/HT) technique. Let $\mathbf{T}_z \in \mathbb{R}^N$ denote the vector of zone air temperature containing N discrete points. The thermodynamics derived from the energy conservation as the partial derivative equation (PDE) form is [38]:

$$\rho_{air}\left(\frac{\partial \mathbf{T}_z}{\partial t} + \frac{\partial \mathbf{U}_i \mathbf{T}_z}{\partial x_i}\right) = \frac{\partial}{\partial x_i}\left(\Gamma_{eff}\frac{\partial \mathbf{T}_z}{\partial x_i}\right) + Q(t), \tag{1}$$

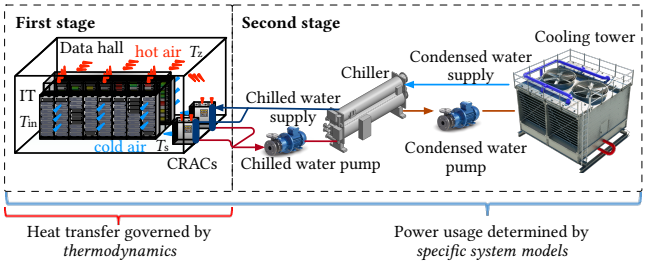

Fig. 2: A typical chilled water-cooled DC with two cooling stages. The first stage is governed by thermodynamics and the power usage is determined by specific system models.

where t is time, x_i is one of the three-dimensional spatial coordinates, $\mathbf{U}_i \in \mathbb{R}^N$ is the vector of air velocity in different directions with i equals 1, 2 or 3, respectively, ρ_{air} is the air density, Γ_{eff} is the diffusion coefficient and Q is the heat load that comprises the sensible part converted from device energy usage and the removed part by the CRACs. In this study, we assume the heat generated from the IT power usage (denoted by P_{IT}) dominates the sensible parts compared with those from lighting and human workers. To simplify the modeling, the EnergyPlus [6] simulation adopts the nodal model by considering that all CRACs take the same supply settings and the data hall has a uniform spatial temperature distribution. In practice, uniform spatial temperature distribution can be achieved with air containment and thermal-aware load balancing [21]. Thus, Eq. (1) is simplified to the ordinary differential equation (ODE) form:

$$\frac{dT_z(t)}{dt} = \frac{m_s(t)}{V_s \rho_{air}}\left(T_s(t) - T_z(t)\right) + \frac{1}{\alpha C_p V_s} P_{IT}(t), \tag{2}$$

where C_p is the air heat capacity, α is the air recirculation ratio, V_s is the data hall volume, m_s and T_s are the supply air mass flow rate and temperature, respectively. The ODE form omits the detailed spatial temperature distribution and focuses on the transient heat transfer process. In this study, we consider both uniform and non-uniform spatial temperature distributions. To simplify presentation, we use the scalar form notation in the following analysis.

The differential equations describe the temperature transition in the data hall as a continuous-time stochastic process. The stochasticity comes from the uncertain evolution of P_{IT} over time. To analyze the control process, we follow the time-slotted treatment [29] to discretize the time into K control periods of τ and assume P_{IT} only changes at the start of each period, i.e., $P_{IT}(t)|_{t\in(k\tau,(k+1)\tau]} = P_{IT}[k]$, $k \le K$. Let $\boldsymbol{\mu}$ denote the control action that consists of the supply air temperature and mass flow rate as $\boldsymbol{\mu} = (\hat{T}_s, \hat{m}_s)$. At the start of the k-th period, the cooling system also implements the control action via the actuator. Formally, $T_s(t)|_{t\in(k\tau,(k+1)\tau]} = \hat{T}_s[k]$, $m_s(t)|_{t\in(k\tau,(k+1)\tau]} = \hat{m}_s[k]$. Thus, the discrete thermal transition function is derived by substituting the above variables to Eq. (2):

$$T_z[k+1] = \mathcal{F}(\mathbf{s}[k], \boldsymbol{\mu}[k]), \tag{3}$$

where $\mathcal{F}$ is the transition function and $\mathbf{s}$ is a vector of the state that consists of the supply air temperature, zone return air temperature and the IT power usage, i.e., $\mathbf{s} = (T_s, T_z, P_{IT})$. In practice, the measurements of temperature and power usage during a control period can be averaged over τ for discrete analysis.

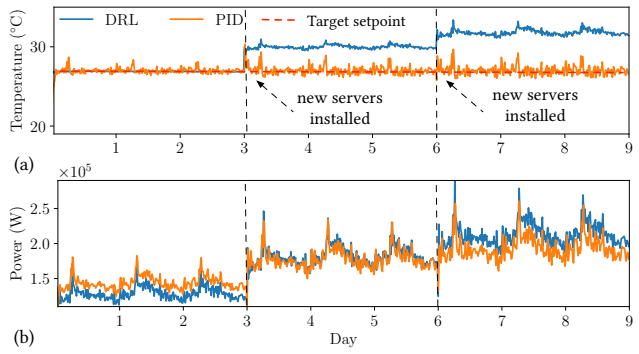

Fig. 3: Example of run-time performance degradation of the DRL agent without online adaptation. When new IT devices are installed on the 3rd and 6th days, (a) the temperature gradually deviates from the target setpoint, i.e., 27°C, and (b) the total power usage is getting higher than that under the PID control.

3.2 System Power Usage

The total DC power usage comprises electricity consumed by the IT devices and the associated cooling systems. The IT power is used by the CPU for computing and the internal fans for dissipating heat, which depend on the new CPU utilization (denoted by U_{IT}) and the previous inlet air temperature (denoted by T_{in}). Thus, the IT power usage of the k-th period is modeled by $P_{\text{IT}}[k] = f_{\text{IT}}(U_{\text{IT}}[k], T_{\text{in}}[k-1])$. The inlet temperature depends on the temperatures of the supply air and recirculated hot air, i.e., $T_{\text{in}}[k] = (1-\alpha)\, T_{\text{s}}[k] + \alpha T_{\text{z}}[k]$. In practice, α can be determined using historical data [38]. For a CW-cooled DC, the cooling power (denoted by P_{c}) is defined by $P_{\text{c}} = P_{\text{crac}} + P_{\text{chp}} + P_{\text{ch}} + P_{\text{cp}} + P_{\text{ct}}$, where $P_{\text{crac}}, P_{\text{chp}}, P_{\text{ch}}, P_{\text{cp}}, P_{\text{ct}}$ are the power usages of the CRAC fans, chilled water pump, chiller, condensed water pump and cooling tower fan, respectively. The power usage of each component at the k-th period can be modeled by $P_{\text{crac}}[k] = f_1(\hat{m}_{\text{s}}[k])$, $P_{\text{chp}}[k] = f_2(\hat{m}_{\text{chw}}[k])$, $P_{\text{ch}}[k] = f_3(T_{\text{chws}}[k], T_{\text{cws}}[k], Q_{\text{ch}}[k])$, $P_{\text{chp}}[k] = f_4(\hat{m}_{\text{cw}}[k])$, and $P_{\text{ct}}[k] = f_5(T_{\text{cws}}[k], T_{\text{cwr}}[k], T_{\text{o}}[k], Q_{\text{ct}}[k])$, respectively, where $\hat{m}_{\text{chw}}$, $\hat{m}_{\text{cw}}$ are mass flow rates of the chilled water and condensed water, T_{chws}, T_{cws}, T_{cwr}, T_{o} are temperatures of the chilled water supply, condensed water supply, condensed water return and outdoor air, Q_{ch} and Q_{ct} are heat loads removed by the chiller and cooling tower, respectively. Typically, T_{chws}, T_{cws} and T_{cwr} are fixed. These models are non-linear in general [35]. However, the detailed forms are often not specified and can only be identified from operational data. In this study, we need to understand the impact of data hall environment changes (e.g., adjusting $\hat{T}_{\text{s}}$ and $\hat{m}_{\text{s}}$) on the power usage of these components. To characterize the impact, in §5, we propose a safety-aware strategy to collect online exploratory data and exploit proper system laws to model the impact.

4 MOTIVATION & THE PHYLLIS APPROACH

This section first uses an example to illustrate the motivation of adapting the DRL agent to system changes. Then, we overview the proposed Phyllis approach and the associated technical problems.

4.1 A Motivating Example

Figure 3 illustrates an example of run-time performance degradation when the number of servers changes over time for a CW-cooled DC simulated by EnergyPlus. We also use this DC for evaluation in §6.2. In this example, the DRL agent is first trained with a reward design (c.f. §6.2) that aims to maintain T_{z} at 27°C and reduce power usage in a training environment of a simulated DC with 100 IT devices. Then, the agent is deployed to a system with the same configuration. From the figures, we observe that the agent can maintain the temperature at the target setpoint and power usage lower than that under the PID control during the first three days when the environment keeps stationary. However, when additional 50 servers are installed on the 3rd and 6th days, respectively, the temperature maintained by the DRL gradually deviates from the setpoint. The higher temperature leads to faster rotation of the internal servers' fan and thus more IT electricity usage. Such degradation is caused by the increases in IT power usage that are not seen during the DRL training. Therefore, the DRL agent needs to adapt to the evolving DC environments.

4.2 Approach Overview & Associated Problems

To model the environment changes, we consider an infinite sequence of MDPs, denoted by $\mathcal{M}_j$, $j = 1, 2, \ldots, \infty$. The state transition probabilities and reward functions are different given the same state-action pairs under different MDPs. Thus, the DRL-based lifelong learning aims to find a parameterized policy that maximizes the discounted expected return with incoming $\mathcal{M}$ sequentially. We denote the policy by π_{θ_j}, where θ_j is the set of parameters learned in $\mathcal{M}_j$. Figure 4 illustrates the proposed Phyllis approach to assist the DRL-based policy adaptation from $\mathcal{M}_{j-1}$ to $\mathcal{M}_j$. Specifically, in the offline stage, Phyllis first identifies a thermal transition model based on known thermodynamics. This step can be completed without interactions with the environment. The resulting model is generally applicable throughout all MDPs, where its minor MDP-dependent error will be rectified by a residual model learned for each MDP in the online stage. When the MDP changes in the online stage, Phyllis implements the following four steps to assist the adaptation. In Step 1 marked by ❶, the identified transition model is used to supervise l periods of exploratory data collection in the new $\mathcal{M}_j$. The actions are still determined by the previous policy $\pi_{\theta_{j-1}}$. In Step 2 marked by ❷, the collected exploratory data is used to fit the power usage models and a residual model to complement the prediction of the previous transition model. Note that as the power usage models change with the environment, it is necessary to fit new power usage models for the new environment that will be used to determine the DRL's reward. This is also the reason for Step 1 to collect exploratory data. In Step 3 marked by ❸, the policy π_{θ_j} is initialized to $\pi_{\theta_{j-1}}$ and pre-trained by interacting with the models obtained in the second step. In step 4 marked by ❹, the adequately pre-trained π_{θ_j} is deployed to the system for further fine-tuning through the interactions with $\mathcal{M}_j$. The time needed for online implementing of Steps 2 & 3 will be given in Table 3 of §6.4. In the following, we formulate the problems in Steps 1, 3 and 4.

The effectiveness of Phyllis relies on the accuracy of the thermodynamics and power usage models. While the thermodynamics $\mathcal{F}$ is governed by the differential equations in §3, modeling the

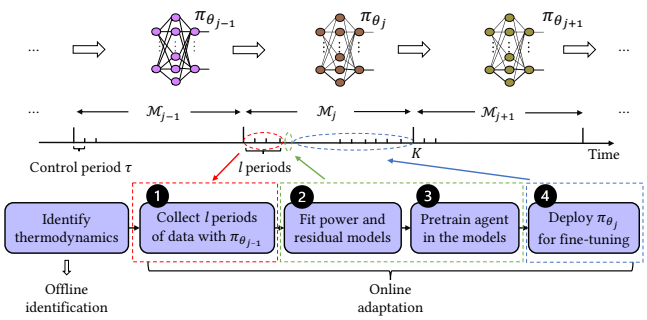

Fig. 4: Overview of the Phyllis approach to achieve lifelong DRL. Phyllis first identifies a transition model to capture the thermodynamics. When the environment changes, Phyllis assists the adaptation by i) collecting l periods of exploratory data, ii) fitting power and residual models with the collected data, iii) pre-training the DRL agent in the fitted models, and iv) deploying the agent for further fine-tuning.

power usage requires online exploratory data in the new operating environment. In this regard, the data collection, directed by the old policy, shall not violate the thermal constraints. With $\mathcal{F}$ identified in the offline stage, Phyllis can predict the temperature at the end of each control period. If the predicted temperature falls out of the allowed range, Phyllis searches for the minimum rectification to the action recommended by $\pi_{\theta_{j-1}}$ to avoid the breach of the thermal constraints. This is formulated as:

$$\begin{aligned} &\tilde{\boldsymbol{\mu}}^*[k] \triangleq \arg\min_{\tilde{\boldsymbol{\mu}}[k]} \frac{1}{2} \|\tilde{\boldsymbol{\mu}}[k] - \boldsymbol{\mu}[k]\|_2^2, \\ &\text{s.t. } \mathcal{F}(\mathbf{s}[k], \tilde{\boldsymbol{\mu}}[k]) \in \mathcal{S}, \end{aligned} \tag{4}$$

where $\mathcal{S}$ is the allowable temperature region to ensure thermal safety, $\tilde{\boldsymbol{\mu}}$ and $\boldsymbol{\mu}$ are the rectified and original actions, respectively. The complexity of solving Eq. (4) depends on the property of $\mathcal{F}$ and the definition of $\mathcal{S}$. Typically, $\mathcal{S}$ is specified with an upper and lower limit based on the service level agreement. In §5, we will present the modeling approach of $\mathcal{F}$ to facilitate fast solving of Eq. (4) for online usage.

With the safely collected exploratory data, we use them to fit the power usage models in Step 2. Once these models for $\mathcal{M}_j$ are identified, we pre-train the policy π_{θ_j} by interacting with the models and deploy for further fine-tuning. This pre-training and fine-tuning addresses a problem of maximizing the expected reward under $\mathcal{M}_j$, which is formally expressed as:

$$\theta_j^* \triangleq \arg\max_{\theta_j} \mathbb{E}_{\pi_{\theta_j}, P_{\mathrm{IT}}} \left[\sum_{k=1}^{K} \gamma^k R[k] \right], \tag{5}$$

where R is the reward and γ is a discount factor. The adaptation for every informed upgrade leads to $\{\theta_1^*, \theta_2^*, \ldots, \theta_j^*, \ldots\}$ and thus achieves lifelong DRL. Note that for $\mathcal{M}_1$, the θ_1 can be randomly initialized or learned by imitating an existing PID controller [5, 38]. In the next section, we present the technical details in modeling the system and solving the above optimization problems in Eqs. (4) and (5), respectively.

5 DETAILED DESIGN OF PHYLLIS

In this section, we present the solutions and technical details involved in the Phyllis approach.

5.1 Offline Thermodynamics Modeling

The thermodynamics model aims to predict the zone air temperature (denoted by $\widetilde{T}_{\mathrm{z}}$) of the next control step given current state-action pairs. This ability is used in determining the action rectification for ensuring safety during exploratory data collection (i.e., Eq. (4). To predict $\widetilde{T}_{\mathrm{z}}[k+1]$, the straight solution is to numerically solve the differential equations presented in §3 with the initial values specified at the k-th control period. However, the numerical solutions are often computationally expensive and non-differentiable in terms of control actions. As a result, grid search is the only viable approach to solving the quadratic problem in Eq. (4), which is undesirable for online usage, because fine-grained grid search incurs high computation overhead. Thus, we aim to develop a differentiable surrogate model which can lead to an efficient solution to Eq. (4). We separately consider the scenarios of uniform and non-uniform data hall spatial temperature distribution.

5.1.1 Uniform spatial temperature distribution. The governing ODE is given by Eq. (2). Assuming the control system has zero steady-state errors for the k-th control period, the supply air temperature and mass flow rate of the period is equal to the applied setpoints $\boldsymbol{\mu} = (\hat{T}_{\mathrm{s}}, \hat{m}_{\mathrm{s}})$. Formally, $m_{\mathrm{s}}(t)|_{t\in(k\tau,(k+1)\tau]} = \hat{m}_{\mathrm{s}}[k]$, $T_{\mathrm{s}}(t)|_{t\in(k\tau,(k+1)\tau]} = \hat{T}_{\mathrm{s}}[k]$. If the IT power remains constant for the k-th control period, the temperature evolution can be modeled by $\widetilde{T}_{\mathrm{z}}(t) = \mathcal{F}_{\mathrm{p}}(T_{\mathrm{z}}[k], P_{\mathrm{IT}}[k], \hat{m}_{\mathrm{s}}[k], \hat{T}_{\mathrm{s}}[k], t)$ where $t \in [0, \tau]$, $\widetilde{T}_{\mathrm{z}}(t)|_{t=0} = T_{\mathrm{z}}[k]$ and $\mathcal{F}_{\mathrm{p}}$ is a FNN-based surrogate model. To capture the thermodynamics, we embed Eq. (2) to the loss function of $\mathcal{F}_{\mathrm{p}}$ to train the model with specified initial values. With these initial values, the physical loss is defined as the averaged residuals of the governing equations in the discrete form:

$$\mathcal{L}_{\mathrm{p}} = \frac{1}{N_{\mathrm{b}}} \frac{1}{N_\tau} \sum_{i=1}^{N_{\mathrm{b}}} \sum_{t=1}^{N_\tau} \left\| \frac{d\widetilde{T}_{\mathrm{z}}(t)}{dt} - \mathcal{H}(T_{\mathrm{z}}[i], P_{\mathrm{IT}}[i], \hat{m}_{\mathrm{s}}[i], \hat{T}_{\mathrm{s}}[i]) \right\|_2^2, \tag{6}$$

where $\mathcal{H}$ is the right-hand side of Eq. (2), N_{b} and N_τ are the batch size of the specified initial values and the intermediate points collected within a control period τ. For example, if τ is 15 minutes and the data is collected every 1 minute, N_τ is 15. We also consider the loss corresponds to data at the initial values by:

$$\mathcal{L}_{\mathrm{b}} = \frac{1}{N_{\mathrm{b}}} \sum_{i=1}^{N_{\mathrm{b}}} \left\| \widetilde{T}_{\mathrm{z}}(t)|_{t=0} - T_{\mathrm{z}}[i] \right\|_2^2. \tag{7}$$

In practice, the initial values of $\hat{m}_{\mathrm{s}}$ and $\hat{T}_{\mathrm{s}}$ are set within the allowable control action ranges. T_{z} can be set to cover a wide range of states for better generalization. P_{IT} can be set based on the designed IT power usage. We will illustrate the ranges of these initial values in Table 6 of the Appendix. The physics-informed modeling captures the prior DC thermodynamics and doesn't require any online data to optimize the above loss functions.

5.1.2 Non-uniform spatial temperature distribution. The governing PDE is given by Eq. (1), which can be also incorporated into the

loss function for training. However, when the spatial domain is discretized with fine-grained mesh grids, the number of temperature points N can be up to millions [37]. Thus, directly approximating such high dimensional output can cause statistical and computational issues. To reduce the modeling complexity, we adopt the POD technique [4] to decompose the high-dimensional temperature field with a linear combination of $J(J \ll N)$ orthogonal basis functions (i.e., POD modes) and the corresponding coefficients as $\mathbf{T}_z[k] = \sum_{i=1}^{J} \beta_i[k]\boldsymbol{\Phi}_i$, where $\boldsymbol{\Phi}_i$ and $\beta_i, i = 1, 2, \ldots, J$ are the vector of the POD modes and coefficient value, respectively. In practice, the POD modes can be derived by the snapshot method based on the results of solving Eq. (1). Once the POD modes are determined, the thermal transition modeling is shifted to predict the low-dimensional POD coefficients given the boundary conditions of the hosted facilities of the k-th period as $\boldsymbol{\beta}[k+1]=\mathcal{F}_{\mathrm{p}}(\hat{\mathbf{T}}_{\mathrm{s}}[k], \hat{\mathbf{m}}_{\mathrm{s}}[k], \mathbf{P}_{\mathrm{IT}}[k], \mathbf{m}_{\mathrm{IT}}[k])$, where $\mathbf{m}_{\mathrm{IT}}$ is the vector of the IT mass flow rates. In practice, $\mathbf{m}_{\mathrm{IT}}$ can be identified offline using historical data [37]. Different from the uniform temperature modeling that directly embeds the physical equation into the loss function for training, the physics in POD modeling is informed via the observational data generated from Eq. (1) to extract the orthogonal basis functions and coefficients.

5.2 Step 1: Safety-Aware Online Exploration

This section presents how to use the thermodynamics model identified in §5.1 to safely guide online exploration after the DC upgrade is implemented. The exploration aims to collect a short period of online data to fit the power usage models and an augmented residual model to further complement the temperature prediction. Since the previously learned policy has converged to address the last MDP $\mathcal{M}_{j-1}$, the collected data may concentrate on a certain operating point and negatively affect the model fitting. To encourage exploration, Phyllis first relaxes the learned policy $\pi_{\theta_{j-1}}$ to randomly select actions from a uniform distribution as:

$$\boldsymbol{\mu}[k] = \begin{cases} \mathrm{Uniform}(\mathcal{A}), & \text{if } k \le \epsilon, \\ \pi_{\theta_{j-1}}(\boldsymbol{\mu} \mid \mathbf{s}), & \text{if } \epsilon < k \le l, \end{cases} \tag{8}$$

where $\mathcal{A}$ is a set of available actions and ϵ is the number of random exploration periods. The relaxed policy may lead to thermal unsafety during exploration. To address this issue, Phyllis adopts the thermal transition model identified in §5.1 to solve the constrained optimization problem in Eq. (4). According to the guideline from the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) [33], the DC temperature should be maintained within a range of certain standards with a lower and upper limit, denoted by T_{l} and T_{u}, respectively. Thus, a proper form of $\mathcal{S}$ is defined as $\mathcal{S} = \left\{T_{\mathrm{z}}^{(i)}[k] \mid T_{\mathrm{l}} \le T_{\mathrm{z}}^{(i)}[k] \le T_{\mathrm{u}}, \forall k, i = 1, 2, \ldots, n\right\}$, where n is the number of deployed temperature sensors. We refer to [28] by considering return zone air T_{z} and IT inlet temperature T_{in} for safety evaluation in §6, respectively. To incorporate the constraints into the problem, we define a cost function (denoted by C) to represent the temperature violation magnitude at the k-th control period by:

$$C[k] = \sum_{i=1}^{n} \mathrm{ReLU}\left(T_{\mathrm{z}}^{(i)}[k] - T_{\mathrm{u}}\right) + \mathrm{ReLU}\left(T_{\mathrm{l}} - T_{\mathrm{z}}^{(i)}[k]\right), \tag{9}$$

where ReLU is the rectified linear activation function defined as $\mathrm{ReLU}(x) = \max\{x, 0\}$. With this definition, the enforcement of the constraints in Eq. (4) is then equivalent to ensure $C[k+1]$ is less equal than 0 after the rectified control action is applied at the k-th period. Formally, $C(\tilde{\boldsymbol{\mu}}[k]) \le 0$. We now present a two-step method to solve this problem.

5.2.1 Convex set projection. In the first step, Phyllis adopts the first-order Taylor expansion to locally approximate the cost with the rectified action at the start of the k-th period as $C(\tilde{\boldsymbol{\mu}}[k]) = C(\boldsymbol{\mu}[k]) + C'(\boldsymbol{\mu}[k])(\tilde{\boldsymbol{\mu}}[k] - \boldsymbol{\mu}[k])$. To ensure $C(\tilde{\boldsymbol{\mu}}[k]) \le 0$, Eq. (4) is converted to a convex quadratic program as:

$$\begin{aligned} \tilde{\boldsymbol{\mu}}^*[k] \triangleq \arg\min_{\tilde{\boldsymbol{\mu}}[k]} \frac{1}{2}\tilde{\boldsymbol{\mu}}^\intercal[k]\mathbf{I}\tilde{\boldsymbol{\mu}}[k] - \boldsymbol{\mu}^\intercal[k]\tilde{\boldsymbol{\mu}}[k], \\ \text{s.t. } C'(\boldsymbol{\mu}[k])^\intercal\tilde{\boldsymbol{\mu}}[k] \le C'(\boldsymbol{\mu}[k])^\intercal\boldsymbol{\mu}[k] - C(\boldsymbol{\mu}[k]), \end{aligned} \tag{10}$$

where $\mathbf{I}$ is an identity matrix and $C'(\boldsymbol{\mu}[k])$ is the first-order derivative of the violation cost to the original control action. With the first-order approximation, the safe action can be efficiently derived by solving Eq. (10) in polynomial time with CVXPY [9]. The first-order approximation is similar to [5] that assumes the state transition function is linear for the investigated system. However, since the DC thermodynamics is nonlinear, the approximation error could degrade thermal safety compliance.

5.2.2 Local search. To mitigate the approximation error, Phyllis uses the original form of $\mathcal{F}_{\mathrm{p}}$ to predict the violation cost and derive the safe action via a local search. In this case, the constraint functions are nonlinear and more accurate to capture the true state transition. While the second step is more rigorous to enforce thermal safety without transition approximation, it can be time-consuming when the dimension of action space is high or the original action deviates far from the safety region. Therefore, Phyllis only conducts it locally when an approximated solution is found by solving Eq. (10). Note that since $\mathcal{F}_{\mathrm{p}}$ may not well approximate the online data before the residual model is identified. To mitigate this effect, we could slightly tighten the temperature bound to offset the approximation error during the first exploration epoch. Figure 5 shows an example of the constraint compliance and rectification overhead by adopting the two-step method for one CRAC with two-dimensional action space. From the figure, we can see the linear projection generates a higher violation cost than the grid search and two-step methods, indicating the linear approximation is insufficient to address the thermal safety compliance. In contrast, the proposed two-step method maintains a lower cost while significantly reducing the search overhead compared with the grid search method.

5.3 Step 2: Power Usage & Residual Modeling

With the collected online data, Phyllis aims to model the impact of data hall environment changes on the system power usage. As discussed in §3, the system power usage mainly comes from the IT devices and the associated cooling systems. For fans and pumps involved in the cooling process, the *affinity laws* [35] describe that the power usage is proportional to the cubic shaft speed. Thus, we can adapt the cubic polynomial regression to model f_1, f_2 and f_5, which reduces the complexity of introducing high-order models. For IT equipment, the power usage is jointly affected by the CPU

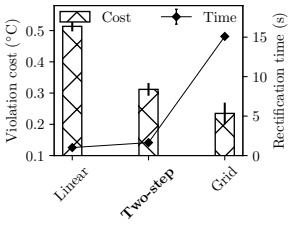

Fig. 5: Rectification effectiveness versus the overhead during the first exploration epoch. The violation costs are averaged over the epoch.

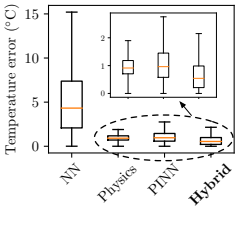

Fig. 6: Evaluation of different dynamics models for temperature prediction errors. The test data are randomly sampled.

utilization and inlet air temperature. Thus, Phyllis adopts the bi-quadratic regression to model the IT power usage of the k-th period as $P_{\text{IT}}[k] = P_{\text{rated}}(a_0 + a_1 U_{\text{IT}}[k] + a_2 U_{\text{IT}}^2[k] + a_3 T_{\text{in}}[k-1] + a_4 T_{\text{in}}^2[k-1] + a_5 U_{\text{IT}}[k] T_{\text{in}}[k-1])$, where P_{rated} is the rated IT power which is specified by the manufacture and a_i, $i = 0, 1, \ldots, 5$ are unknown coefficients. The chiller is the most complex component in the cooling system, whose power usage is affected by multiple factors. Phyllis quantifies its power usage by the coefficient of performance (COP) defined as the ratio of the heat removed to the chiller power usage by $\text{COP} = Q_{\text{ch}}/P_{\text{ch}}$. To satisfy the energy balance, the cooling load Q_{ch} should approximately match the generated heat load in the data hall, i.e., $Q_{\text{ch}} \approx P_{\text{IT}} + P_{\text{crac}}$. Therefore, if the COP value is determined, we can derive the chiller power usage by $P_{\text{ch}} = (P_{\text{IT}} + P_{\text{crac}})\,/\text{COP}$. Its value has been observed to change quadratically with the supply air temperature in [21]. Similarly, we use a bi-quadratic polynomial regression to model the COP as a function of the supply air temperature and mass flow rate at the k-th period by $\text{COP}[k] = b_0 + b_1 \hat{T}_{\text{s}}[k] + b_2 \hat{T}_{\text{s}}^2[k] + b_3 \hat{m}_{\text{s}}[k] + b_4 \hat{m}_{\text{s}}^2[k] + b_5 \hat{T}_{\text{s}}[k] \hat{m}_{\text{s}}[k]$, where b_i, $i = 0, 1, \ldots, 5$ are unknown coefficients. Ideally, the minimum number of data pairs needed to fit these models is equal to the coefficients numbers.

In addition to fitting the power usage models, Phyllis also uses the collected data to learn a residual model to complement the prediction of the physics model. As discussed in §3, the dynamics captured by the physical prior can be inaccurate due to incomplete consideration of unknown factors, such as heat transfer from interzone air mixing or infiltration of outside air. To mitigate this effect, we augment the physics model with a data-driven component (denoted by $\mathcal{F}_{\text{d}}$) to predict the unmodeled residuals. Thus, the dynamics is approximated by $\mathcal{F} = \mathcal{F}_{\text{p}} + \mathcal{F}_{\text{d}}$. With this decomposition, we aim to fine-tune $\mathcal{F}_{\text{p}}$ to be close to the true dynamics while leaving $\mathcal{F}_{\text{d}}$ as a complementary term. Thus, the objective can be achieved by minimizing the third loss function:

$$\mathcal{L}_{\text{d}} = \frac{1}{N_{\text{d}}} \sum_{i=1}^{N_{\text{d}}} \left\| \widetilde{T}_{\text{z}i} - T_{\text{z}i} \right\|_2^2 + \lambda \left\| \mathcal{F}_{\text{d}} \right\|_2^2, \tag{11}$$

where N_{d} is the number of collected data samples and λ is a hyperparameter to balance $\mathcal{F}_{\text{d}}$ as small as possible. In this study, the residual model is continuously updated with more collected online data to improve prediction accuracy. Figure 6 shows the box plot of the absolute temperature prediction error of different dynamics models. In this example, the black-box model is an ensemble of black-box MLPs used in [15] trained with historical data. The test data are randomly sampled. From this figure, we can see the black-box model extrapolates poorly to the test data since the historical data are insufficient to cover various states. In contrast, the physics-informed methods all achieve good prediction performance. With the hybrid modeling, the average prediction error is only 0.7°C.

Algorithm 1 Physics-informed policy adaptation

Input: Initialize policy parameters $\pi_{\theta_{j-1}}$, temperature transition model $\mathcal{F}_{\text{p}}$, residual model $\mathcal{F}_{\text{d}}$, IT power models f_{IT}, cooling power models f_i, $i = 1, 2, \ldots, 5$, initial state estimations, environment dataset $\mathcal{D}_{\text{env}}$ and synthetic dataset $\mathcal{D}_{\text{model}}$.
Output: Optimal policy $\pi^*_{\theta_j}$ for $\mathcal{M}_j$.

1: Train $\mathcal{F}_{\text{p}}$ in the offline stage with estimated initial values and thermodynamics based on Eq. (6) and (7);
2: **while** $\mathcal{M}$ changes **do**
3: // *Step 1: Safe exploratory data collection*
4: **for** l exploration steps **do**
5: Generate actions $\boldsymbol{\mu}$ by Eq. (8);
6: Generate data with safe actions rectified by solving Eq. (4) using the two-step method and add data to $\mathcal{D}_{\text{env}}$;
7: **end for**
8: // *Step 2: Fit power usage and residual models*
9: Sample data from $\mathcal{D}_{\text{env}}$;
10: Fit models of f_{IT} and f_i, $i = 1, 2, \ldots, 5$;
11: Train $\mathcal{F}_{\text{d}}$ based on Eq. (11);
12: // *Step 3: Pre-train the policy to optimize Eq* (5)
13: **for** G gradient steps **do**
14: Collect synthetic data and add to $\mathcal{D}_{\text{model}}$;
15: Sample data from $\mathcal{D}_{\text{model}}$ and update policy π_{θ_j};
16: **end for**
17: // *Step 4: Deploy policy for further fine-tuning*
18: Fine-tune π_{θ_j} online by repeating line 13 to 16 with $\mathcal{D}_{\text{env}}$
19: **end while**

5.4 Steps 3 & 4: Pre-training and Fine-tuning

With the identified transition and power models, Phyllis adopts the model-based reinforcement learning paradigm to transfer the policy to the target MDP. Specifically, the policy is first trained by the synthetic data generated from these models. Compared with collecting data from the physical DC, the synthetic data generation is much more sample efficient as the computation overhead for executing these models is low. Moreover, as our transition model captures physical laws, it is good at extrapolating states that are hard to obtain from a stably running DC. Thus, the agent can extensively explore a better initial policy without risking the physical system before online deployment. Once the policy is adequately trained, Phyllis deploys it to interact with the target system to further improve its performance. With model-assisted training, the policy is expected to adapt faster when deployed to a new MDP.

Putting together all the steps presented in §5.1-§5.4, we have the physics-informed policy adaptation process. Algorithm 1 shows the pseudocode of this process.

6 EVALUATION

This section evaluates the performance of Phyllis in two cases with uniform and non-uniform temperature distributions and compares it with five baseline approaches.

6.1 Evaluation Methodology and Settings

We present the implementation and configurations of Algorithm 1. The configurations include 1) the DRL policy used to optimize the energy efficiency, 2) the parametrization of $\mathcal{F}_\mathrm{p}$ and $\mathcal{F}_\mathrm{d}$, 3) and the hyperparameters. The details are as follows.

6.1.1 Implementation & Evaluation settings. To optimize DC energy efficiency, we adopt soft actor-critic (SAC) [14] as the optimization algorithm due to three considerations. First, the entropy regularization in SAC encourages exploration, making it suitable when transferring to a new environment. Second, it learns in an off-policy manner and can effectively use the data sampled by the rectified actions. Third, it can be used for DC cooling control with continuous action space. In this study, we use the SAC in Tianshou library [41]. To capture the temperature transition and prediction residual, we model $\mathcal{F}_\mathrm{p}$ and $\mathcal{F}_\mathrm{d}$ with two MLPs, respectively. $\mathcal{F}_\mathrm{p}$ is trained in a physics-informed manner while $\mathcal{F}_\mathrm{d}$ is trained with online streaming data enters. To drive the simulation, we use the weather data that contains 1-year outdoor air temperature collected from a tropical area [6]. The IT utilization trace is aggregated from a real Internet DC hosting 4,000 servers [1]. Table 4 in the Appendix summarizes other default settings used in our evaluation.

6.1.2 Comparison baselines. To evaluate the performance of Phyllis, we implement the following baseline approaches:

■ **BL1** (*PID*) uses the feedback of temperature deviation to adjust the supply air temperature and focuses on maintaining the temperature at a specified setpoint [39]. The coefficients for the proportional, integral and derivative terms are 0.4, 0.6 and 0, respectively.

■ **BL2** (*Model-free fine-tuning*) sequentially fine-tunes one single agent when transferring to new environment [43]. Without using models to rectify actions, the constraint is addressed in a relaxed manner using a reward shaping method from [22].

■ **BL3** (*Model-free progressive learning*) extends BL2 with an expansion-based progressive neural network (PNN) [32]. When adapting to a new environment, it freezes previously learned parameters and allocates new sub-networks for re-learning.

■ **BL4** (*Model-based policy optimization*) refers to a state-of-the-art model-based method that uses short rollouts from the model to update the agent [15]. The used model is an ensemble of 10 black-box neural networks as [15] that jointly predicts the temperature transition and reward with a rollout length of 1.

■ **BL5** (*Physics-informed only*) uses the physics-informed method to learn the transition model and adopts it to guide the safe exploration. Similar to [36, 45], this baseline only focuses on addressing the safety constraints without additional power consumption modeling and model-based pre-training. It stands as an ablation of the Phyllis approach.

6.1.3 Evaluation metrics. We consider the following four metrics to evaluate the performance when the agent adapts to a new DC environment, i.e., 1) the *jumpstart performance* that measures the initial violation when encountering system upgrades; 2) the *convergence speed* that measures how many days are needed for the temperature to converge to the setpoint; 3) the *average cost* that measures the temperature violation magnitude, and 4) the *average total power usage* during the online learning period.

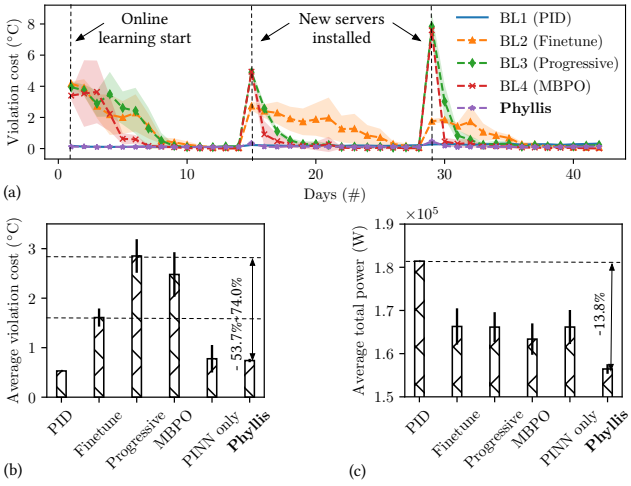

Fig. 7: Adaptation performance of different methods on the CW-cooled DC. (a) Per-day violation cost; (b) average violation cost and (c) average DC total power usage during the online learning period. The shaded areas and error bars show the standard deviation over five independent experiments.

6.2 Case 1: Install Servers in a CW-Cooled DC

6.2.1 DC testbed and configurations. We first evaluate the adaptation performance when new servers are installed in a CW-cooled DC. The DC model is configured by [22] and implemented using EnegyPlus 9.5.0 simulator. The control actions applied at the k-th period consist of the CRAC's supply air temperature and mass flow rate defined as $\mathbf{a}[k] = (\hat{T}_\mathrm{s}[k], \hat{m}_\mathrm{s}[k])$. The states consist of the supply air temperature, return zone air temperature, and power usage of the IT and cooling systems, respectively. Formally, $\mathbf{s}[k] = (T_\mathrm{s}[k], T_\mathrm{z}[k], P_\mathrm{IT}[k], P_\mathrm{c}[k])$. The control objective is to maintain the air temperature at a certain setpoint, denoted by $\hat{T}_\mathrm{z}$, and reduce total power usage. Thus, we define the reward function of the k-th time step as $R[k] = -\lambda_\mathrm{P} P_\mathrm{DC}[k] + \lambda_\mathrm{T} \mathcal{G}(T_\mathrm{z}[k], \hat{T}_\mathrm{z})$ where $\mathcal{G}$ is a Gaussian function defined as $\mathcal{G} = \exp(-\lambda_1 (T_\mathrm{z}[k] - \hat{T}_\mathrm{z})^2)$, P_DC is the sum of cooling and IT power usage, λ_1, λ_T and λ_P are hyperparameters set to 0.5, 1.5 and 10^{-5}, respectively. In this study, the setpoint is 27°C, which is a typical return air setting in DC [28]. The allowable temperature variation range is 1°C. Thus, the lower and upper bound in Eq. (9) are 26°C and 28°C, respectively. In this case, the original DC is equipped with 100 servers. Each server is specified with a rated power of 1,000W. To evaluate the Phyllis approach, we install additional 60 servers at the start of the 3rd and 6th week, respectively.

6.2.2 Jumpstart and convergence performance. Figure 7(a) shows the per-day average violation cost of various approaches during the

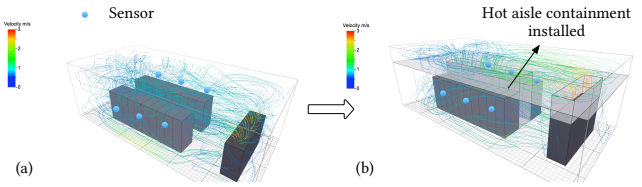

Fig. 8: DC layout of the second case. (a) Air mixing; (b) air separation before and after the containment is installed. The sensor is placed at the positions marked with blue dots.

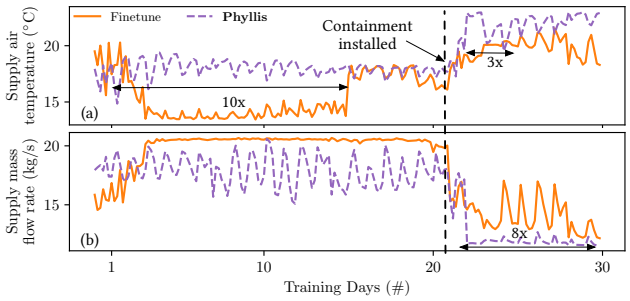

Fig. 9: CRAC's actions over time. (a) supply air temperature; (b) supply air mass flow rate. Phyllis quickly learns to increase the air temperature and decrease the mass flow rate to save cooling energy after the containment is installed.

six-week operations. The dashed lines with symbols are mean values and the shaded areas represent the standard deviation over five independent experiments. From this figure, we can see that model-free fine-tuning and progressive learning generate high violation costs on the initial day when new servers are installed. Progressive learning has a higher cost since it allocates new sub-networks with randomly initialized parameters for re-learning. MBPO also generates a high cost when first deployed to interact with the system, but it converges faster than the model-free approaches, indicating the model-based method has higher sample efficiency. However, since the black-box model requires more data and does not extrapolate well to the evolving dynamics, it still suffers from high violation costs when new servers are installed in the DC. In contrast, the PID control and Phyllis can maintain the temperature violation at a lower cost during the entire adaptation process. Note the baseline BL5 is excluded in Figure 7(a) for better visualization since it also achieves similar performance in addressing safety compliance as Phyllis. In summary, Phyllis converges 8x to 10x faster than the model-free fine-tuning approach.

6.2.3 Average performance. Figure 7(b) and (c) show the average violation cost and total power usage over the six-week operational period. From Figure 7(b), we observe that Phyllis reduces the violation cost by 53.7% to 74.0% compared with other DRL-based methods. This suggests the proposed two-step method is effective in preventing safety violations. The PID uses temperature deviation feedback to control and also generates a low cost. However, it has high power usage since it only focuses on maintaining the temperature at the target setpoint. In contrast, the DRL-based methods additionally admit the DC power usage for optimization and thus exhibit savings in energy usage. From Figure 7(c), we can find that MBPO slightly saves more energy compared with the model-free approaches, suggesting that the model-based method performs well in the evolving dynamics. Phyllis achieves 13.8% power saving compared with the PID controller. We also conduct the ablation study that only adopts the trained PINN to guide safe exploration. While it also performs well in addressing thermal safety as Phyllis, the power usage under this method is higher than that under Phyllis. This is due to the lack of the model-based pre-training in Step 3.

6.3 Case 2: Install Containment to Hot Aisle

6.3.1 DC testbed and configurations. We next evaluate the adaptation performance when a hot aisle containment is installed. Installing air containments is a typical method to improve DC energy efficiency [28]. Figure 8(a) and (b) show the data hall layout and air flow streamlines before and after the containment is installed, respectively. The evaluated DC is equipped with 20 racks placed in two rows that host 299 servers in total [4]. Each server has a rated power of 1,000W. A CRAC unit is installed to supply cold air and draw hot air on its top. Six sensors are deployed at the inlet positions of the rack to monitor the IT intake air temperature. Since the air mixing causes non-uniform temperature distribution before the containment is installed, we adopt the fine-grained POD method to model the temperature transition in the data hall. In this case, $\mathcal{F}_{\mathrm{p}}$ is modeled using a Gaussian process in GPytorch [12] to predict the POD coefficients. Similar to the first case, the control actions include the CRAC's supply air temperature and mass flow rate applied at the k-th period defined as $\mathbf{a}[k] = (\hat{T}_{\mathrm{s}}[k], \hat{m}_{\mathrm{s}}[k])$. The states consist of the temperature measured by the six sensors and the power usage of the IT and cooling systems, respectively. Formally, $\mathbf{s}[k] = (T_{\mathrm{in}1}[k], \ldots, T_{\mathrm{in}6}[k], P_{\mathrm{IT}}[k], P_{\mathrm{c}}[k])$. In this case, the control objective is to maintain the server inlet temperature within the range of 18-27°C according to the ASHARE guideline [33]. Thus, the reward function of the k-th time step is defined as $R[k] = -\lambda_{\mathrm{P}} P_{\mathrm{DC}}[k] - \lambda_{\mathrm{T}} C[k]$, where C is the temperature violation cost with the lower and upper bound in Eq. (9) defined as 18°C and 27°C, respectively. λ_{P} and λ_{T} are two hyperparameters set to 10^{-5} and 1, respectively. In this case, the hot aisle containment is installed on the 21st day during the operations.

6.3.2 Convergence analysis. Figure 9 (a) and (b) show the CRAC supply air temperature and mass flow rate varying over the training days. The dashed orange line and purple line represent the results of the model-free fine-tuning and Phyllis, respectively. From the figures, we observe that Phyllis quickly learns to adjust the supply air temperature and mass flow rate periodically with respect to the IT load variation after one day of exploration. Such behavior is attributed to the air recirculation effect before the hot aisle containment is installed. To maintain the inlet temperature within the allowable range, the CRAC needs to decrease the supply air temperature and increase the mass flow rate when the IT workload is high. In contrast, the model-free agent learns a conservative policy to supply low temperature and high mass flow rate, which consume more cooling energy. It takes about 15 days for the model-free agent to learn to periodically adjust the supply air temperature. Phyllis

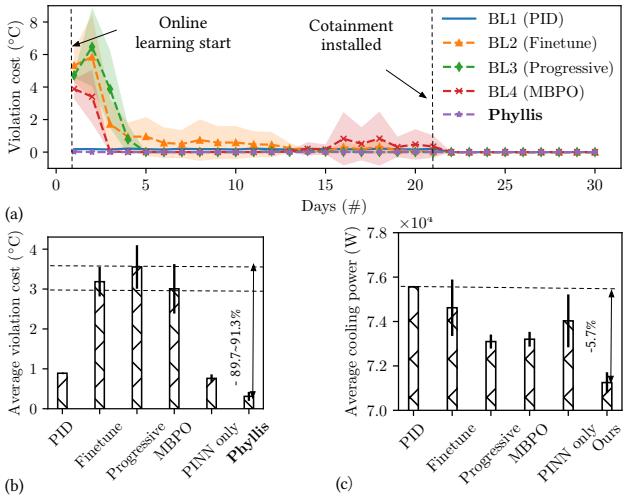

Fig. 10: Performance of different methods on the second case. (a) Per-day violation cost; (b) average violation cost and (c) DC cooling power usage during the online learning period.

brings about 8x acceleration compared with the model-free agent. After the containment is installed on the 21st day, the hot air is separated from the cold supply air. Phyllis then quickly learns to increase the supply air temperature and reduce the mass flow rate to save cooling energy. In contrast, the model-free agent supplies a lower supply temperature and higher mass flow rate. It takes another 8 days to learn an energy-efficient policy.

6.3.3 Jumpstart and average performance. Figure 10 (a) shows the per-day violation cost. From the figure, we observe that Phyllis and PID both maintain a low-cost value over the period. Other DRL-based methods suffer from high jumpstart violation costs before the containment is installed and take about 3 to 7 days to converge. After the containment is installed, all the methods can eliminate the safety violation since the hot air is separated from the cold supply air, indicating the hot aisle containment is useful in preventing thermal unsafety. In this case, since the number of IT devices keeps the same, we focus on analyzing the cooling power usage. Figure 10(b) and (c) show the average violation cost and cooling power usage over the online learning period. From the figures, we can see Phyllis reduces violation cost by 89.7% to 91.3% compared with other DRL-based methods and saves 5.7% cooling power usage compared with the PID controller.

6.4 Online Adaptation Overhead

We next present the time needed for Steps 2 & 3 during the online adaptation in Table 3. From the results, we can see that the two steps consume at most 1120.26 seconds (i.e., 18.67 minutes) for data collection and thermal modeling. Thus, these two steps can be finished in about 1 to 2 control periods (i.e., 15 to 30 minutes). The DRL training takes more time with non-uniform thermal modeling. This is because the forwarding process of POD takes more time since it needs to generate a high dimensional temperature vector for each control step.

Table 3: Overhead for online model fitting & pre-training.

Step	Uniform modeling (PINN) (s)	Non-uniform modeling POD (s)
Step 2	4.23 ± 1.37	3.90 ± 1.48
Step 3	40.89s ± 3.0	1116.36 ± 0.64

*Intel(R) Xeon(R) CPU E7-8880 v4 @ 2.20GHz

7 DISCUSSION

We now discuss two issues that are not addressed in this paper and leave them for future work.

■ **Heterogeneity in state-action space:** In this study, we focus on investigating policy transfer to environments where only the reward and transition functions change over time. We assume the state-action space keeps the same. In future studies, heterogeneous transfer learning can be incorporated into our framework to address the changes in the state-action space. Specifically, when new sensors or CRACs are installed, the input or output of the networks used in this framework can be correspondingly expanded.

■ **Clustering of similar environments:** From the evaluation results, our approach can adapt well to abrupt changes in DC. For small changes (e.g., the 28th day in case 1 with fewer servers being installed), the original policy also adapts faster with a few days of fine-tuning. Therefore, environments with similar configurations can be clustered. The online re-learning will be triggered only when new clusters are instantiated. This consideration will further reduce the re-learning overhead.

8 CONCLUSION

This paper proposes Phyllis, a physics-informed lifelong reinforcement learning approach for DC cooling control. By leveraging the thermodynamics and system laws, Phyllis can safely collect online data and efficiently model the DC thermal transition and power usage. With the identified state transition and reward models, Phyllis adopts the model-based paradigm to accelerate online adaptation. Our extensive evaluation shows that, with new IT devices and containment installed, Phyllis converges 8x to 10x faster than model-free fine-tuning methods with at most 0.74°C temperature overshoot. With a faster convergence speed, Phyllis saves more energy usage for DC operations. The proposed Phyllis approach sheds light on deploying DRL-based policies to non-stationary DC with better safety and convergence speed management.

ACKNOWLEDGMENTS

This research is supported in part by the National Research Foundation, Singapore, funded under its Energy Research Testbed and Industry Partnership Funding Initiative, part of the Energy Grid (EG) 2.0 programme and its Central Gap Fund (Award No. NRF2020NRF-CG001-027), in part by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 of RT14/22 and RG96/20, in part by the Nanyang Technological University, Singapore, under its 8962-Accelerating Creativity and Excellence grant call (No. NTU–ACE2020-01), in part by the National Natural Science Foundation, China, funded under the project (No. 62262026) and the project of Jiangxi Education Department (No. GJJ211111), respectively.

REFERENCES

[1] 2021. Alibaba Cluster Trace Program. https://github.com/alibaba/clusterdata.

[2] Anders SG Andrae and Tomas Edler. 2015. On Global Electricity Usage of Communication Technology: Trends to 2030. *Challenges* 6, 1 (2015), 117–157.

[3] Tom Bawden et al. 2016. Global Warming: Data Centres to Consume Three Times as Much Energy in Next Decade, Experts Warn. *The Independent* 23 (2016), 276.

[4] Zhiwei Cao, Ruihang Wang, Xin Zhou, and Yonggang Wen. 2022. Reducio: Model Reduction for Data Center Predictive Digital Twins via Physics-Guided Machine Learning. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 1–10.

[5] Bingqing Chen, Priya L Donti, Kyri Baker, J Zico Kolter, and Mario Bergés. 2021. Enforcing Policy Feasibility Constraints through Differentiable Projection for Energy Optimization. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*. 199–210.

[6] Drury B Crawley, Linda K Lawrie, Curtis O Pedersen, and Frederick C Winkelmann. 2000. EnergyPlus: Energy Simulation Program. *ASHRAE journal* 42, 4 (2000), 49–56.

[7] Marc Deisenroth and Carl E Rasmussen. 2011. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *Proceedings of the 28th International Conference on machine learning (ICML-11)*. 465–472.

[8] Xiangtian Deng, Yi Zhang, and He Qi. 2022. Towards Optimal HVAC Control in Non-Stationary Building Environments Combining Active Change Detection and Deep Reinforcement Learning. *Building and Environment* 211 (2022), 108680.

[9] Steven Diamond and Stephen Boyd. 2016. CVXPY: A Python-Embedded Modeling Language for Convex Optimization. *Journal of Machine Learning Research* 17, 83 (2016), 1–5.

[10] Ján Drgoňa, Aaron R Tuor, Vikas Chandan, and Draguna L Vrabie. 2021. Physics-Constrained Deep Learning of Multi-Zone Building Thermal Dynamics. *Energy and Buildings* 243 (2021), 110992.

[11] Benjamin Eysenbach, Swapnil Asawa, Shreyas Chaudhari, Sergey Levine, and Ruslan Salakhutdinov. 2020. Off-dynamics Reinforcement Learning: Training for Transfer with Domain Classifiers. *arXiv preprint arXiv:2006.13916* (2020).

[12] Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. 2018. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In *Advances in Neural Information Processing Systems*.

[13] Gargya Gokhale, Bert Claessens, and Chris Develder. 2022. Physics informed neural networks for Control Oriented Thermal Modeling of Buildings. *Applied Energy* 314 (2022), 118852.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In *International conference on machine learning*. PMLR, 1861–1870.

[15] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. 2019. When to Trust Your Model: Model-Based Policy Optimization. *Advances in Neural Information Processing Systems* 32 (2019).

[16] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. 2021. Physics-Informed Machine Learning. *Nature Reviews Physics* 3, 6 (2021), 422–440.

[17] Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. 2022. Towards Continual Reinforcement Learning: A Review and Perspectives. *Journal of Artificial Intelligence Research* 75 (2022), 1401–1476.

[18] Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, MK Ryu, and Greg Imwalle. 2018. Data Center Cooling Using Model-Predictive Control. *Advances in Neural Information Processing Systems* 31 (2018).

[19] Yuanlong Li, Yonggang Wen, Dacheng Tao, and Kyle Guan. 2019. Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning. *IEEE transactions on cybernetics* 50, 5 (2019), 2002–2013.

[20] Xin-Yang Liu and Jian-Xun Wang. 2021. Physics-Informed Dyna-Style Model-Based Deep Reinforcement Learning for Dynamic Control. *Proceedings of the Royal Society A* 477, 2255 (2021), 20210618.

[21] Justin D Moore, Jeffrey S Chase, Parthasarathy Ranganathan, and Ratnesh K Sharma. 2005. Making Scheduling" Cool": Temperature-Aware Workload Placement in Data Centers.. In *USENIX ATC, General Track*. 61–75.

[22] Takao Moriyama, Giovanni De Magistris, Michiaki Tatsubori, Tu-Hoa Pham, Asim Munawar, and Ryuki Tachibana. 2018. Reinforcement Learning Testbed for Power-Consumption Optimization. In *Methods and Applications for Modeling and Simulation of Complex Systems: 18th Asia Simulation Conference, AsiaSim 2018*. Springer, 45–59.

[23] David Moss and John H. Bean. 2012. Energy Impact of Increased Server Inlet Temperature. (2012). https://www.se.com/us/en/download/document/SPD_JBEN-7KTR88_EN/

[24] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. 2018. Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 7559–7566.

[25] Srinarayana Nagarathinam, Yashovardhan S Chati, Malini Pooni Venkat, and Arunchandar Vasan. 2022. PACMAN: Physics-Aware Control MANager for HVAC. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 11–20.

[26] Avisek Naug, Marcos Q'uiñones-Grueiro, and Gautam Biswas. 2020. A Relearning Approach to Reinforcement Learning for Control of Smart Buildings. In *Annual Conference of the PHM Society*, Vol. 12. 14–14.

[27] Avisek Naug, Marcos Quinones-Grueiro, and Gautam Biswas. 2022. Deep Reinforcement Learning Control for Non-Stationary Building energy management. *Energy and Buildings* 277 (2022), 112584.

[28] John Niemann, Kevin Brown, and Victor Avelar. 2011. Impact of Hot And Cold Aisle Containment on Data Center Temperature and Efficiency. *Schneider Electric Data Center Science Center, White Paper* 135 (2011), 1–14.

[29] K. Ogata. 1995. *Discrete-Time Control Systems*. Prentice-Hall, Inc.

[30] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 2019. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *Journal of Computational physics* 378 (2019), 686–707.

[31] Yongyi Ran, Han Hu, Xin Zhou, and Yonggang Wen. 2019. DeepEE: Joint Optimization of Job Scheduling and Cooling Control for Data Center Energy Efficiency Using Deep Reinforcement Learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 645–655.

[32] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. 2016. Progressive Neural Networks. *arXiv preprint arXiv:1606.04671* (2016).

[33] ASHRAE TC et al. 2016. Data Center Power Equipment Thermal Guidelines and Best Practices. *ASHRAE TC 9.9, ASHRAE, USA* (2016).

[34] Duc Van Le, Yingbo Liu, Rongrong Wang, Rui Tan, Yew-Wah Wong, and Yonggang Wen. 2019. Control of Air Free-Cooled Data Centers in Tropics via Deep Reinforcement Learning. In *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 306–315.

[35] Hoang Dung Vu, Kok Soon Chai, Bryan Keating, Nurislam Tursynbek, Boyan Xu, Kaige Yang, Xiaoyan Yang, and Zhenjie Zhang. 2017. Data Driven Chiller Plant Energy Optimization with Domain Knowledge. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1309–1317.

[36] Ruihang Wang, Xinyi Zhang, Xin Zhou, Yonggang Wen, and Rui Tan. 2022. Toward Physics-Guided Safe Deep Reinforcement Learning for Green Data Center Cooling Control. In *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 159–169.

[37] Ruihang Wang, Xin Zhou, Linsen Dong, Yonggang Wen, Rui Tan, Li Chen, Guan Wang, and Feng Zeng. 2020. Kalibre: Knowledge-based Neural Surrogate Model Calibration for Data Center Digital Twins. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 200–209.

[38] Xiaodong Wang, Xiaorui Wang, Guoliang Xing, Jinzhu Chen, Cheng-Xian Lin, and Yixin Chen. 2011. Towards Optimal Sensor Placement for Hot Server Detection in Data Centers. In *2011 31st International Conference on Distributed Computing Systems*. IEEE, 899–908.

[39] Ya-Gang Wang, Zhi-Gang Shi, and Wen-Jian Cai. 2001. PID Autotuner and Its Application in HVAC Systems. In *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, Vol. 3. IEEE, 2192–2196.

[40] Zhi Wang, Han-Xiong Li, and Chunlin Chen. 2019. Incremental Reinforcement Learning in Continuous Spaces via Policy Relaxation and Importance Weighting. *IEEE transactions on neural networks and learning systems* 31, 6 (2019), 1870–1883.

[41] Jiayi Weng, Huayu Chen, Dong Yan, Kaichao You, Alexis Duburcq, Minghao Zhang, Yi Su, Hang Su, and Jun Zhu. 2022. Tianshou: A Highly Modularized Deep Reinforcement Learning Library. *Journal of Machine Learning Research* 23, 267 (2022), 1–6. http://jmlr.org/papers/v23/21-1127.html

[42] Shichao Xu, Yangyang Fu, Yixuan Wang, Zhuoran Yang, Zheng O'Neill, Zhaoran Wang, and Qi Zhu. 2022. Accelerate Online Reinforcement Learning for Building HVAC Control with Heterogeneous Expert Guidances. In *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 89–98.

[43] Shichao Xu, Yixuan Wang, Yanzhi Wang, Zheng O'Neill, and Qi Zhu. 2020. One for Many: Transfer Learning for Building HVAC Control. In *Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 230–239.

[44] Chi Zhang, Sanmukh R Kuppannagari, Rajgopal Kannan, and Viktor K Prasanna. 2019. Building HVAC Scheduling Using Reinforcement Learning via Neural Network Based Model Approximation. In *Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation*. 287–296.

[45] Qingang Zhang, Muhammad Haiqal Bin Mahbod, Chin-Boon Chng, Poh-Seng Lee, and Chee-Kong Chui. 2022. Residual Physics and Post-Posed Shielding for Safe Deep Reinforcement Learning Method. *IEEE Transactions on Cybernetics* (2022), 1–12. https://doi.org/10.1109/TCYB.2022.3178084

[46] Zhuangdi Zhu, Kaixiang Lin, and Jiayu Zhou. 2020. Transfer Learning in Deep Reinforcement Learning: A Survey. *arXiv preprint arXiv:2009.07888* (2020).

Table 4: Experimental settings.

Hyperparameter	Setting	Hyperparameter	Setting
Control period τ (minutes)	15	PINN hidden units	[32, 32]
Update per step	96	$\mathcal{F}_\text{d}$ hidden units	[32, 32]
Actor/Critic learning rate	1e-3	Used POD modes	5
Actor/Critic hidden units	[32, 32]	Entropy regularization	0.02
Training batch size	256	Discounted factor (γ)	0.99
Pre-training steps	2880	Exploration steps	96

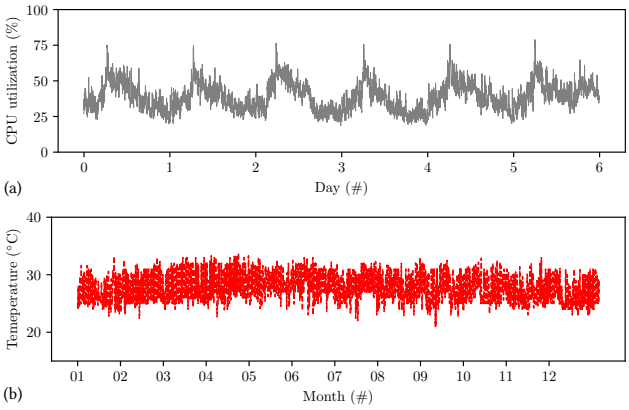

Fig. 11: (a) Aggregated CPU utilization trace; (b) one-year historical weather data collected from the tropical area.

A EVALUATION SETTINGS AND DATASET

The hyperparameter settings used in the experimental evaluation are summarized in Table 4. To drive the simulation, we also need to specify the CPU utilization and outdoor air temperature. Fig. 11(a) shows the aggregated CPU utilization trace collected from a real-world DC hosting 4,000 servers, which is provided by Alibaba Inc. [1]. Figs. 11(b) shows the outdoor air temperature collected from the tropical zone, which is provided by EnergyPlus.

B SYSTEM POWER USAGE MODELS

In the evaluation of this paper, we adopt the instantiated curves from `2ZoneDataCenterHVACwEconomizer.idf` provided by EnergyPlus [6] to model the DC cooling system power usage. This model is also used in [19, 22, 36, 44] for evaluation. For the IT devices, we model the power usage as a biquadratic function of the CPU utilization (denoted by U_IT) and the inlet air temperature (denoted by T_in), i.e., $P_\text{IT} = P_\text{rated}(c_0 + c_1 U_\text{IT} + c_2 U_\text{IT}^2 + c_3 T_\text{in} + c_4 T_\text{in}^2 + c_5 U_\text{IT} T_\text{in})$, where P_rated is the rated IT power usage which is specified by the manufacture and $c_i, i = 0, 1, \ldots, 5$ are coefficients. In this paper, these coefficients are obtained from EnergyPlus [6] for Case 1 and the on-site measurement from Schneider Electric [23] for Case 2. Table 5 shows the fitted coefficient values for the two cases. For Case 1, the IT power usage is linear in terms of the inlet air temperature. While for Case 2, the relationship becomes quadratic. Note that the design inlet temperature for the two cases are set to 18°C and 22°C, respectively.

Table 5: Coefficient values for IT power model.

Case	Coefficient					
	c_0	c_1	c_2	c_3	c_4	c_5
Case 1	-1	1	0	0.0556	0	0
Case 2	0.32	1	0	-0.032	0.0008	0

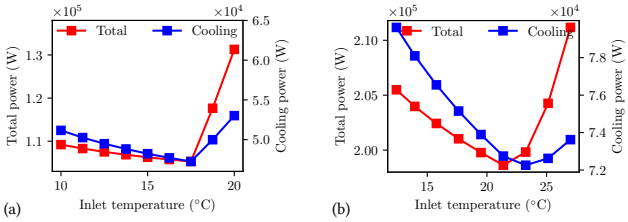

Fig. 12: Total power usage versus inlet air temperature (a) Case 1; (b) Case 2.

To help understand the optimization results, we plot the average total and cooling power usage under various inlet air temperatures in Figure 12. The input IT utilization is from Figure 11(a). From the figures, the total DC power usage exhibits an interesting trend where it initially declines and then rises as the inlet air temperature increases. This can be attributed to the fact that the internal IT fans need to rotate faster when the intake air temperature rises. When the temperature exceeds a certain point, the rise in IT power usage may offset the savings from the cooling side. As a result, simply increasing the inlet air temperature may not be beneficial for energy saving. The energy efficient cooling control is expected to find a policy that can maintain the IT inlet temperature at an optimal setpoint.

C INITIAL VALUE SETTINGS

To identify the thermal transition model in the offline stage, we need to specify the initial value ranges to create the training dataset. Normally, the initial value ranges should be set to cover a wide range of system states for better generalization. Table 6 shows the initial value ranges for training $\mathcal{F}_\text{p}$. In this paper, ten values are uniformly sampled within each specified range.

Table 6: Initial value ranges for training $\mathcal{F}_\text{p}$.

Variable	Range	Variable	Range
$\hat{T}_\text{s}$ (°C)	[10, 25]	T_z (°C)	[20, 35]
$\hat{m}_\text{s}$ (kg/s)	[5, 15]	P_IT (kW)	[60, 180]