SustainDC: Benchmarking for Sustainable Data Center Control

Avisek Naug[†], **Antonio Guillen**[†], **Ricardo Luna**[†], **Vineet Gundecha**[†], **Cullen Bash**, **Sahand Ghorbanpour**, **Sajad Mousavi**, **Ashwin Ramesh Babu**, **Dejan Markovikj**, **Lekhapriya D Kashyap**, **Cullen Bah**,**Desik Rengarajan**, **Soumyendu Sarkar**[†*]

Hewlett Packard Enterprise (Hewlett Packard Labs)

{avisek.naug, antonio.guillen, rluna, vineet.gundecha, cullen.bash, sahand.ghorbanpour, sajad.mousavi, ashwin.ramesh-babu, dejan.markovikj, lekhapriya.dheeraj-kashyap, desik.rengarajan, soumyendu.sarkar}@hpe.com

Abstract

Machine learning has driven an exponential increase in computational demand, leading to massive data centers that consume significant energy and contribute to climate change. This makes sustainable data center control a priority. In this paper, we introduce SustainDC, a set of Python environments for benchmarking multi-agent reinforcement learning (MARL) algorithms for data centers (DC). SustainDC supports custom DC configurations and tasks such as workload scheduling, cooling optimization, and auxiliary battery management, with multiple agents managing these operations while accounting for the effects of each other. We evaluate various MARL algorithms on SustainDC, showing their performance across diverse DC designs, locations, weather conditions, grid carbon intensity, and workload requirements. Our results highlight significant opportunities to improve data center operations using MARL algorithms. Given the increasing use of DC due to AI, SustainDC provides a crucial platform for developing and benchmarking advanced algorithms essential for achieving sustainable computing and addressing other heterogeneous real-world challenges.

1 Introduction

One of the growing areas of energy and carbon footprint (CFP) can be traced to cloud data centers (DCs). The increased use of cloud resources for batch workloads related to AI model training, multimodal data storage and processing, or interactive workloads like streaming services, hosting websites have prompted enterprise clients to construct numerous data centers. Governments and regulatory bodies are increasingly focusing on environmental sustainability and imposing stricter regulations to reduce carbon emissions. This has prompted industry-wide initiatives to adopt more intelligent DC control approaches. This paper presents SustainDC, a sustainable DC Multi-Agent Reinforcement Learning (MARL) set of environments. SustainDC helps promote and prioritize sustainability, and it serves as a platform that facilitates collaboration among AI researchers, enabling them to contribute to a more environmentally responsible DC.

The main contributions of this paper are the following:

- A highly customizable suite of environments focused on Data Center (DC) operations, designed to benchmark energy consumption and carbon footprint across various DC config-

[*]Corresponding author. †These authors contributed equally.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

urations. The framework supports the subclassing of models for different DC components ranging from workloads and individual server specifications to cooling systems, enabling users to test fine-grained design choices.

- The environments are implemented using the Gymnasium *Env* class, facilitating the benchmarking of various control strategies to optimize energy use, reduce carbon footprint, and evaluate related performance metrics.
- Supports both homogeneous and heterogeneous multi-agent reinforcement learning (MARL) controllers and traditional non-ML controllers. Extensive studies within these environments demonstrate the advantages and limitations of various multi-agent approaches.
- SustainDC enables reward shaping, allowing users to conduct ablation studies on specific DC components to optimize performance in targeted areas.
- SustainDC serves as a comprehensive benchmark environment for heterogeneous, multi-agent, and multi-objective reinforcement learning algorithms, featuring diverse agent interactions, customizable reward structures, high-dimensional observations, and reproducibility.

Code, licenses, and setup instructions for SustainDC are available at GitHub[2]. The documentation can be accessed at [3].

2 Related Work

Recent advancements in Reinforcement Learning (RL) have led to an increased focus on optimizing energy consumption in areas such as building and DC management. This has resulted in the development of several environments for RL applications. *CityLearn* (1) is an open-source platform that supports single and MARL strategies for energy coordination and demand response in urban environments. *Energym* (2), *RL-Testbed* (3) and *Sinergym* (4) were developed as RL wrappers that facilitate communication between Python and EnergyPlus, enabling RL evaluation on the collection of buildings modeled in EnergyPlus. *SustainGym* (5) is one of the latest suite of general purpose RL tasks for evaluation of sustainability, simulating electric vehicle charging scheduling and battery storage bid, which lends itself to benchmarking different control strategies for optimizing energy, carbon footprint, and related metrics in electricity markets.

Most of the above-mentioned works use *EnergyPlus* (6) or, *Modelica* (7), (8) which were primarily designed for modeling thermo-fluid interactions with traditional analytic control with little focus on Deep Learning applications. The APIs provided in these works only allow sampling actions in a model free manner, lacking a straightforward approach to customization or re-parameterization of system behavior. This is because most of the works have a set of pre-compiled binaries (e.g. FMUs in Modelica) or fine-tuned spline functions (in EnergyPlus) to simulate nominal behavior. Furthermore, there is a significant bottleneck in using these precompiled environments from Energyplus or Modelica for Python based RL applications due to latency associated with cross-platform interactions, versioning issues in traditional compilers for EnergyPlus and Modelica, unavailability of open source compilers and libraries for executing certain applications.

SustainDC allows users to simulate the electrical and thermo-fluid behavior of large DCs directly in Python. Unlike other environments that rely on precompiled binaries or external tools, SustainDC is easily end-user customizable and fast It enables the design, configuration, and control benchmarking of DCs with a focus on sustainability. This provides the ML community with a new benchmark environment specifically for Heterogeneous MARL in the context of DC operations, allowing for extensive goal-oriented customization of the MDP transition function, state space, actions space, and rewards.

3 Data Center Operational Model

Figure 1 illustrates the typical components of a DC operation as modeled in SustainDC. *Workloads* are uploaded to the DC from a proxy client. For non-interactive batch workloads, some of these jobs can be scheduled flexibly, allowing delays to different periods during the day for optimization. This

[2]GitHub repository: https://github.com/HewlettPackard/dc-rl.
[3]Documentation: https://hewlettpackard.github.io/dc-rl.

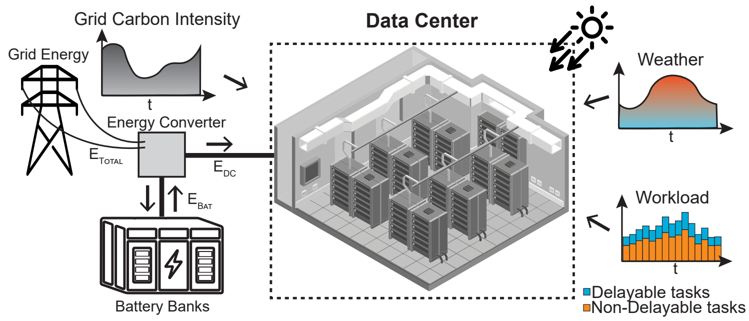


Figure 1: Operational Model of a SustainDC Data Center

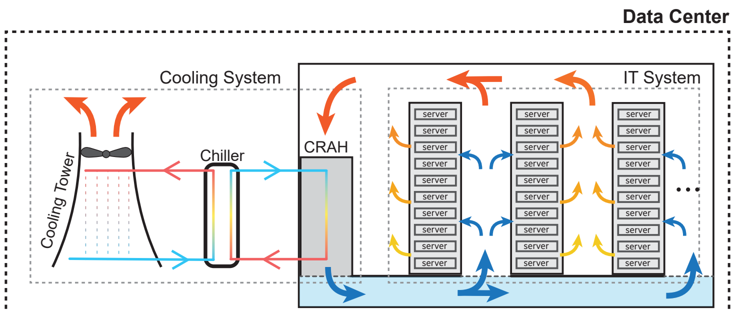


Figure 2: Model of the data center. The configuration allows customization of the number of cabinets per row, the number of rows, and the number of servers per cabinet. The cooling system, comprising the CRAH, chiller, and cooling tower, manages the heat generated by the IT system.

creates a scheduling challenge of postponing workloads to times when *Grid Carbon Intensity* (*CI*), energy consumption, or energy pricing is lower.

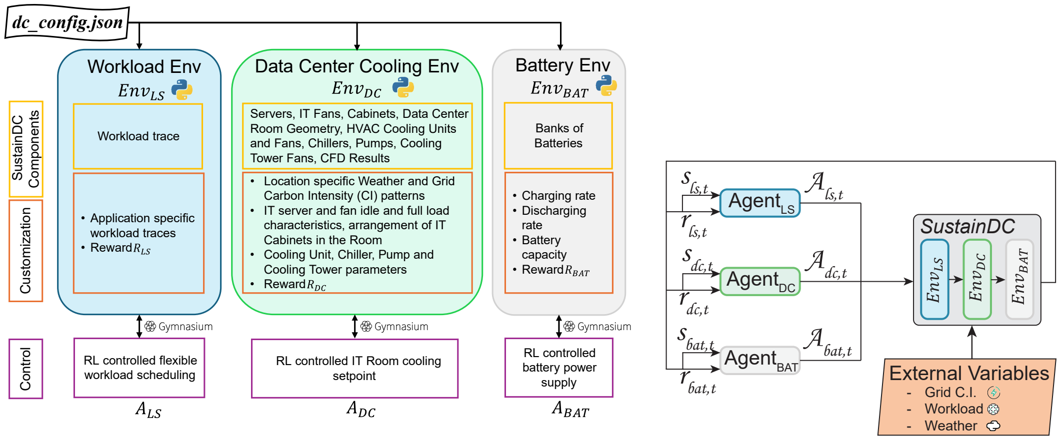
As the servers (IT systems) in the DC process these workloads, they generate heat that must be removed. A complex HVAC system with multiple components is used to cool the IT system. As shown in Figure 2, warm air rises from the servers via convection. Driven by the HVAC fan's forced draft, this warm air enters the *Computer Room Air Handler* (CRAH) (depicted by red arrows), where it is cooled to an optimal setpoint by a heat exchange process using a "primary" chilled water loop. The chilled air is then returned to the IT room through a plenum located beneath the DC (shown by blue arrows). The warmed water from this loop returns to the *Chiller*, where another heat exchange process transfers heat to a "secondary" chilled water loop, which carries the heat to a *Cooling Tower*. The cooling tower fan, operating at variable speeds, rejects this heat to the external environment, with fan speed and energy consumption determined by factors such as the secondary loop's inlet temperature at the cooling tower, the desired outlet temperature setpoint, and external air temperature and humidity. Depending on the external *Weather* and processed *Workload*, the IT and cooling systems consume *Grid Energy*. Selecting the optimal cooling setpoint for the CRAH can reduce the DC's carbon footprint and also impacts the servers' energy efficiency (9).

Larger DCs may include onsite *Battery Banks* that charge from the grid during low *CI* periods and may optionally provide auxiliary energy during high *CI* periods. This introduces a decision-making sustainability challenge to determine the optimal charging and discharging intervals for the batteries.

These three control problems are interrelated, motivating the development of testbeds and environments for evaluating multi-agent control approaches that collectively aim to minimize carbon footprint, energy and water usage, energy cost, and other sustainability metrics of interest.

4 SustainDC environment overview

A high-level overview of SustainDC is provided in Figure 3, outlining the three main environments developed in *Python* along with their individual components, customization options, and associated control challenges.



(a) High-level overview of SustainDC, showing the three main environments (*Workload Env*, *Data Center Cooling Env*, and *Battery Env*) along with their customizable components and control actions.

(b) RL loop in SustainDC, depicting how states and actions are formed from individual agents.

Figure 3: SustainDC overview and RL loop

The *Workload Environment* models and controls the execution and scheduling of delay-tolerant workloads within the DC.

In the *Data Center Environment*, servers housed in IT room cabinets process these workloads. This environment simulates both electrical and thermo-fluid dynamics, modeling heat generated by the workload processing and its transfer to the external environment through HVAC cooling components.

The *Battery Environment* simulates grid charging during off-peak hours and supplies auxiliary energy to the DC during periods of high grid carbon intensity, offering a solution to manage energy demand sustainably.

Detailed physics-based implementations for each environment are available in the supplementary document. Customization parameters for all aspects of the DC environment design in SustainDC can be fully specified through *dc_config.json*, a universal configuration file.

Figure 3a further illustrates SustainDC, showing the *Workload Environment*, *Data Center Environment*, and *Battery Environment* along with their customizable parameters. Figure 3b depicts the RL loop in SustainDC, illustrating how agents' actions and states optimize DC operations, considering external variables like grid CI, workload, and weather.

4.1 Workload Environment

The *Workload Environment* (Env_{LS}) manages the execution and scheduling of delay tolerant workloads within the DC by streaming workload traces (measured in FLOPs) over a specified time period. SustainDC includes a set of open-source workload traces from *Alibaba* (10) and *Google* (11) data centers. Users can customize this component by adding new workload traces to the *data/Workload* folder or by specifying a path to existing traces in the *dc_config.json* file.

Some workloads are flexible, meaning they can be rescheduled within an allowable time horizon. Tasks such as updates or backups do not need immediate execution and can be delayed based on urgency or Service-Level Agreements (SLA). This flexibility allows workloads to be shifted to periods of lower grid carbon intensity (CI), thereby reducing the DC's overall carbon footprint (CFP).

Users can also customize the CI data. By default, we provide a one-year CI dataset for the following states: Arizona, California, Georgia, Illinois, New York, Texas, Virginia, and Washington, locations selected due to their high data center density. The carbon intensity data files, sourced from eia.gov (`https://api.eia.gov/bulk/EBA.zip`), are located in the *data/CarbonIntensity* folder.

Let B_t be the instantaneous DC workload trace at time t, with $X\%$ of the load reschedulable up to N simulation steps into the future. The objective of an RL agent ($Agent_{LS}$) is to observe the current time of day (SC_t), the current and forecast grid CI data ($CI_{t...t+L}$), and the remaining amount of reschedulable workload (D_t). Based on these observations, the agent chooses an action $A_{ls,t}$ (as shown in Table 1) to reschedule the flexible portion of B_t, to minimize the net CFP over N steps.

4.2 Data Center Environment

The *Data Center* environment (Env_{DC}) provides a comprehensive set of configurable models and specifications. For IT-level design, SustainDC enables users to define IT Room dimensions, server cabinet arrangements (including the number of *rows* and *cabinets* per row), and both *approach* and *return* temperatures. Additionally, users can specify server and fan power characteristics, such as *idle power*, *rated full load power*, and *rated full load frequency*.

On the cooling side, SustainDC allows customization of the *chiller reference power*, *cooling fan reference power*, and the supply air *setpoint* temperature for IT Room cooling. It also includes specifications for the pump and cooling tower, such as *rated full load power* and *rated full load frequency*. All these parameters can be configured in the *dc_config.json* file.

One of SustainDC's key features is its ability to automatically adjust HVAC cooling capacities based on workload demands and IT room configurations, a process known as "sizing." This ensures that the data center remains adequately cooled without unnecessary energy expenditure. In contrast, previous environments often neglected this capability, resulting in inaccurate outcomes. For example, changing IT room configurations in other environments typically impacted only IT energy consumption without considering the overall cooling requirements, leading to inconsistent RL-based control results, as seen in *RL-Testbed* in (3). SustainDC addresses this by integrating custom supply and approach temperatures derived from Computational Fluid Dynamics (CFD) simulations, simplifying the complex calculations of temperature changes between the IT Room HVAC and the IT Cabinets (9).

In addition, SustainDC includes weather data (in *data/Weather*) in the .epw format for the same locations as the CI data. This data, sourced from `https://energyplus.net/weather`, represents typical weather conditions for these regions. Users can also specify their own weather files if needed.

Given $\hat{B}_t$ as the adjusted workload from the *Workload Environment*, the goal of the RL agent ($Agent_{DC}$) is to select an optimal cooling setpoint $A_{dc,t}$ (Table 1) to minimize the net carbon footprint CFP from combined cooling (E_{hvac}) and IT (E_{it}) energy consumption over an N-step horizon. In SustainDC, the agent's default state space includes the time of day and year (SC_t), ambient weather (t_{db}), IT Room temperature (t_{room}), previous step cooling (E_{hvac}) and IT (E_{it}) energy usage, and forecasted grid CI data ($CI_{t...t+L}$).

4.3 Battery Environment

The *Battery Environment* (Env_{BAT}) is based on battery charging and discharging models, such as $f_{charging}(BatSoc, \delta\tau)$ from (12). Parameters for these components, including battery capacity, can be configured in the *dc_config.json* file.

The objective of the RL agent ($Agent_{BAT}$) is to optimally manage the battery's state of charge ($BatSoc_t$). Using inputs such as the net energy consumption ($E_{hvac} + E_{it}$) from the *Data Center* environment, the time of day (SC_t), the current battery state of charge ($BatSoc_t$), and forecasted grid CI data ($CI_{t...t+L}$), the agent determines an action $A_{bat,t}$ (as outlined in Table 1). Actions include charging the battery from the grid, taking no action, or discharging to provide auxiliary energy to the data center, all aimed at minimizing the overall carbon footprint, energy consumption, etc.

4.4 Heterogeneous Multi Agent Control Problem

While SustainDC enables users to tackle the individual control problems for each of the three environments, the primary goal of this paper is to establish a multi-agent control benchmark that facilitates joint optimization of the CFP by considering the coordinated actions of all three agents ($Agent_{LS}$, $Agent_{DC}$, and $Agent_{BAT}$). The sequence of operations for the joint multi-agent and multi-environment functions can be represented as follows:

$$
\begin{aligned}
Agent_{LS} :& (SC_t \times CI_t \times D_t \times B_t) \rightarrow A_{ls,t} && (1)\\
Agent_{DC} :& (SC_t \times t_{db} \times t_{room} \times E_{hvac} \times E_{it} \times CI_t) \rightarrow A_{dc,t} && (2)\\
Agent_{BAT} :& (SC_t \times Bat\_SoC \times CI_t) \rightarrow A_{bat,t} && (3)\\
Env_{LS} :& (B_t \times A_{ls,t}) \rightarrow \hat{B}_t && (4)\\
Env_{DC} :& (\hat{B}_t \times t_{db} \times t_{room} \times A_{dc,t}) \rightarrow (E_{hvac}, E_{it}) && (5)\\
Env_{BAT} :& (Bat\_SoC \times A_{bat,t}) \rightarrow (Bat\_SoC, E_{bat}) && (6)\\
CFP_t =& (E_{hvac} + E_{it} + E_{bat}) \times CI_t && (7)
\end{aligned}
$$

where E_{bat} represents the net discharge from the battery based on the change in battery state of charge (Bat_SoC), which can be positive or negative depending on the action $A_{bat,t}$. If the battery provides auxiliary energy, E_{bat} is negative; if it charges from the grid, E_{bat} is positive.

The objective of the multi-agent problem is to determine θ_{LS}, θ_{DC}, and θ_{BAT}, which parameterize the policies for $Agent_{LS}$, $Agent_{DC}$, and $Agent_{BAT}$, respectively, such that the total CFP is minimized over a specified horizon N. For this study, we set $N = 31 \times 24 \times 4$, representing a 31-day horizon with a step duration of 15 minutes.

$$\Big(\theta_{LS}, \theta_{DC}, \theta_{BAT}\Big) = argmin\Big(\sum_{t=0}^{t=N} CFP_t\Big) \tag{8}$$

4.5 Rewards

While CFP reduction is the default objective in SustainDC, the reward formulation is highly customizable, allowing users to incorporate alternative objectives such as total energy consumption, operating costs across all DC components, and water usage.

We primarily consider the following default rewards for the three environments (Env_{LS}, Env_{DC}, Env_{BAT}):

$$(r_{LS}, r_{DC}, r_{BAT}) = \Big(-(CFP_t + LS_{Penalty}), -(E_{hvac,t} + E_{it,t}), -(CFP_t)\Big)$$

Here, $LS_{Penalty}$ is a penalty applied to the Load Shifting Agent ($Agent_{LS}$) in the Workload Environment (Env_{LS}) if it fails to reschedule flexible workloads within the designated time horizon N. Specifically, if D_t is positive at the end of a horizon N, $LS_{Penalty}$ is assigned. Details on calculating

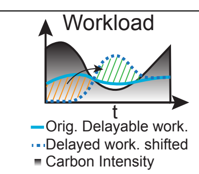
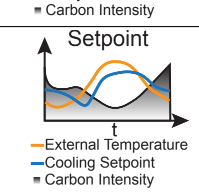
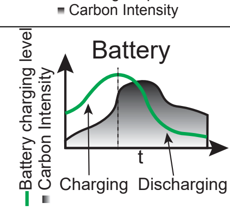
Agent	Control Knob	Actions	Optimization Strategy	Figure
$\mathbf{Agent_{LS}}$	Delay-tolerant workload scheduling	$\begin{cases} 0 & \text{Store Delayable Tasks} \\ 1 & \text{Compute All Immediate Tasks} \\ 2 & \text{Maximize Throughput} \end{cases}$	Shift tasks to periods of lower CI/lower external temperature/other variables to reduce the CFP.	
$\mathbf{Agent_{DC}}$	Cooling Setpoint	$\begin{cases} 0 & \text{Decrease Setpoint} \\ 1 & \text{Maintain Setpoint} \\ 2 & \text{Increase Setpoint} \end{cases}$	Optimize cooling by adjusting cooling setpoints based on workload, external temperature, and CI.	
$\mathbf{Agent_{BAT}}$	Battery energy supply/store	$\begin{cases} 0 & \text{Charge Battery} \\ 1 & \text{Hold Energy} \\ 2 & \text{Discharge Battery} \end{cases}$	Store energy when CI/temperature/workload/other is low and use stored energy when is high to reduce CFP.	

Table 1: Overview of control choices in SustainDC: the tunable knobs, the respective action choices, optimization strategies, and visual representations.

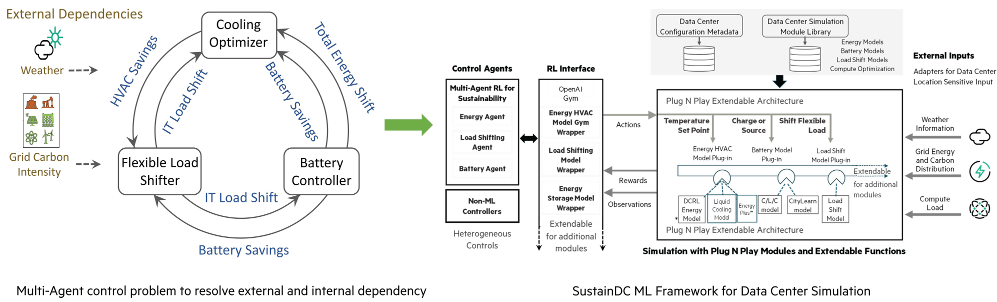


Figure 4: Extendable and plug-and-play design of SustainDC for data center control to address the multi-agent holistic optimization of data centers for resolving multiple dependencies in real-time.

$LS_{Penalty}$ are provided in the supplemental document. Users can opt for custom reward formulations by subclassing the base reward class in *utils/reward_creator.py*.

Based on these individual rewards, we can formulate an independent or collaborative reward structure, where each agent receives partial feedback in the form of rewards from the other agent-environment pairs. The collaborative feedback reward formulation for each agent is formulated as:

$$R_{LS} = \alpha * r_{LS} + (1-\alpha)/2 * r_{DC} + (1-\alpha)/2 * r_{BAT}$$
$$R_{DC} = (1-\alpha)/2 * r_{LS} + \alpha * r_{DC} + (1-\alpha)/2 * r_{BAT}$$
$$R_{BAT} = (1-\alpha)/2 * r_{LS} + (1-\alpha)/2 * r_{DC} + \alpha * r_{BAT}$$

Here, α is the weighting parameter. This reward-sharing mechanism enables agents to incorporate feedback from their actions across environments, making it suitable for independent critic multi-agent RL algorithms, such as IPPO (13). For instance, the adjusted CPU load $\hat{B}_t$ influences data center energy demand ($E_{cool} + E_{it}$), which subsequently affects the battery optimizer's charge-discharge decisions and ultimately impacts the net CO_2 footprint. Consequently, we explore a collaborative reward structure and conduct ablation experiments with varying α values to assess the effectiveness of reward sharing.

4.6 Extendable plug-n-play Data Center Simulation Platform

Figure 4 illustrates the extendable and plug-and-play design of SustainDC framework for data center control to address the multi-agent optimization of data centers for resolving multiple internal and external dependencies of agents in real-time. The three different controllers for **Cooling Optimizer, Flexible Load Shifter** and **Battery Controller** can be substituted with **RL** or **non-RL controllers**. Similarly, the underlying models performing the simulation can be substituted easily using the **Modules and Extendable Functions** block. In the future, we plan to include the models for next generation of fanless direct liquid cooling for AI servers (14) for Energy HVAC Model Plug-in.

5 Evaluation Metrics and Experimental Settings

We consider five metrics to evaluate various RL approaches on SustainDC. The CO_2 *footprint* (CFP) represents the cumulative carbon emissions associated with DC operations over the evaluation period. *HVAC Energy* refers to the energy consumed by cooling components, including the chiller, pumps, and cooling tower. *IT Energy* refers to the energy consumed by the servers within the DC. *Water Usage*, the volume of chilled water recirculated through the cooling system, is a critical metric in DCs where chilled water supply from a central plant is constrained, and efficient use of this resource helps minimize the DC's water footprint. Finally, *Task Queue* tracks the accumulation of compute FLOPs from workloads that are deferred for rescheduling under lower CI periods. Higher Task Queue values indicate poorer SLA performance within the DC.

Experiments were conducted on an Intel® Xeon® Platinum 8470 server with 104 CPUs, utilizing 4 threads per training agent. All hyperparameter configurations for benchmark experiments are detailed in the supplemental document. The codebase and documentation are linked to the paper.

6 Benchmarking Algorithms on SustainDC

The purpose of SustainDC is to explore the benefits of jointly optimizing the *Workload*, *Data Center*, and *Battery Environments* to reduce the operating CFP of a DC. To investigate this, we can perform ablation studies in which we evaluate net operating CFP by running trained RL agents in only one or two of the SustainDC environments while employing baseline methods (B_*) in the other environments. For the *Workload Environment* (Env_{LS}), the baseline (B_{LS}) assumes no workload shifting over the horizon, which aligns with current standard practices in most data centers. For the *Data Center Environment* (Env_{DC}), we use the industry-standard ASHRAE Guideline 36 as the baseline (B_{DC}) (15). In the *Battery Environment* (Env_{BAT}), we adapt the method from (12) for real-time operation, reducing the original optimization horizon from 24 hours to 3 hours as our baseline (B_{BAT}). Future work will include further baseline comparisons using Model Predictive Control (MPC) and other non-ML control algorithms.

Next, we perform ablations on the collaborative reward parameter α, followed by benchmarking various multi-agent RL approaches. This includes multi-agent PPO (16) with an independent critic for each actor (IPPO) (13) and a centralized critic with access to states and actions from other MDPs (MAPPO) (17). Given the heterogeneous nature of action and observation spaces in SustainDC, we also benchmark several heterogeneous multi-agent RL (HARL) methods (18), including HAPPO (Heterogeneous Agent PPO), HAA2C (Heterogeneous Agent Advantage Actor Critic), HAD3QN (Heterogeneous Agent Dueling Double Deep Q Network), and HASAC (Heterogeneous Agent Soft Actor Critic). MARL agents were trained on one location and evaluated across different locations.

In Figure 5, we compare the relative performance of different RL algorithms using a radar chart based on the evaluation metrics in Section 5. Since reporting absolute values may lack context, we instead plot relative performance differences, offering insights into the *pros* and *cons* of each approach. (Absolute values for these benchmark experiments are provided in the supplementary document in tabular format.) Metrics are normalized by their mean and standard deviation, with lower values positioned at the radar chart periphery and higher values toward the center. Hence, the larger the area for an approach on the radar chart, the better its performance across the evaluated metrics.

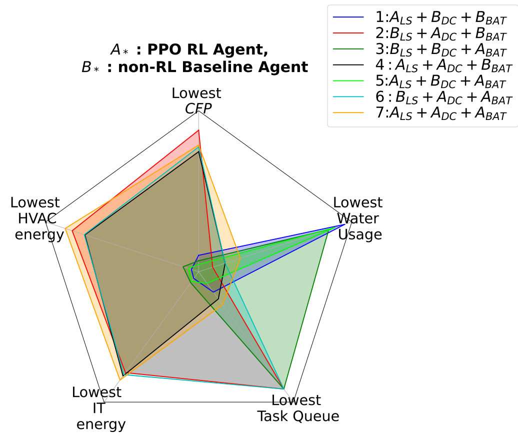
6.1 Single vs multi-agent Benchmarks

Figure 5a compares the relative performance of a single RL agent versus multi-agent RL benchmarks, highlighting the advantages of a MARL approach for sustainable DC operations. Among single RL agent approaches, the workload manager RL agent (Experiment 1) and the battery agent (Experiment 3) perform similarly in reducing water usage. The standalone DC (cooling) RL agent (Experiment 2) demonstrates strong performance in both energy and CFP reduction. Note that for Experiments 1 and 3, the Lowest Task Queue metric should be disregarded, as the baseline workload manager does not shift workloads and thus inherently has the lowest task queue.

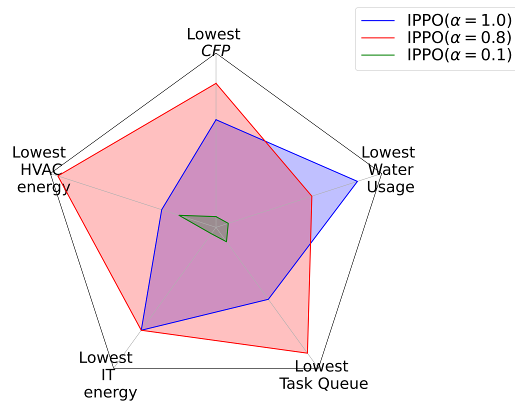
When we evaluate pairs of RL agents working simultaneously, the absence of a cooling optimization agent (e.g., Experiment 5) results in performance similar to single RL agent implementations (Experiments 1 and 3), where only A_{LS} or A_{BAT} are used with baseline agents. This indicates that the RL-based cooling optimizer significantly improves overall performance compared to the rule-based ASHRAE Guideline 36 controller (as seen in Experiments 2 and 4). Finally, when all three RL agents operate simultaneously without a shared critic (Experiment 7 using IPPO), they achieve better outcomes in energy consumption, water usage, and task queue management, with a CFP relatively similar to other experiments. The combined performance across all three agents highlights the benefits of a MARL approach for DC optimization.

6.2 Reward Ablation on α

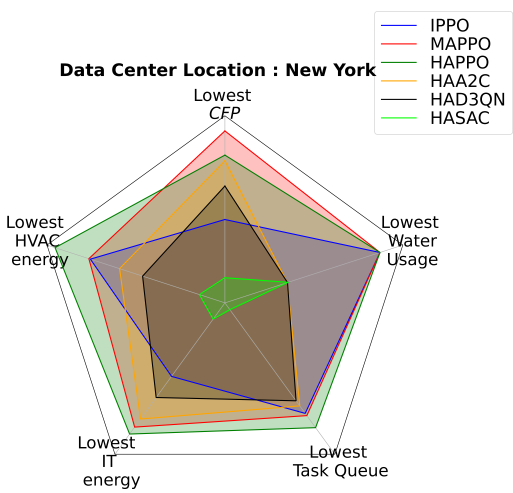
Figure 5b, shows the relative differences in performance when considering collaborative reward components. We considered 2 values of α at the extremes to indicate no collaboration ($\alpha = 1.0$) and relying only on the rewards of other agents ($\alpha = 0.1$). An intermediate value of $\alpha = 0.8$ was chosen based on similar work on reward-based collaborative approach in (19; 20). The improvement in setting $\alpha = 0.8$ shows that considering rewards from other agents can improve performance w.r.t. no collaboration ($\alpha = 1.0$) especially in a partially observable MDP.



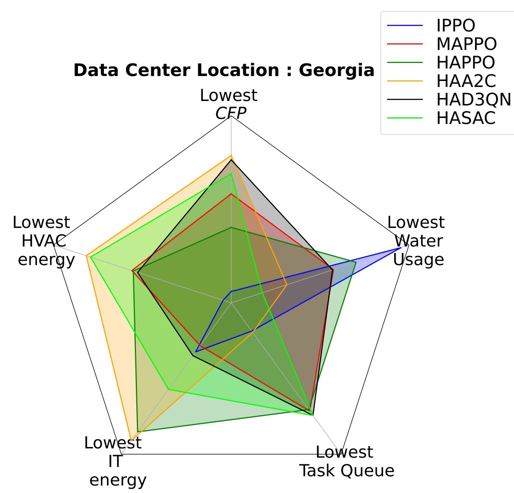
(a) Single RL agent, two RL agents and three RL agents For single agents, PPO was used (Average result over 5 runs)



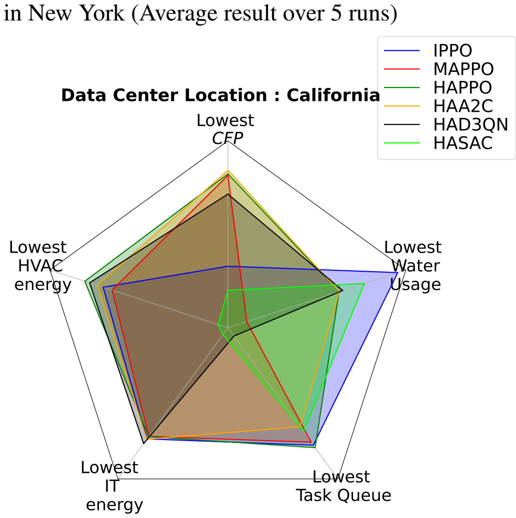
(b) IPPO with different values of collaborative reward coefficient α (Average result over 12 runs)



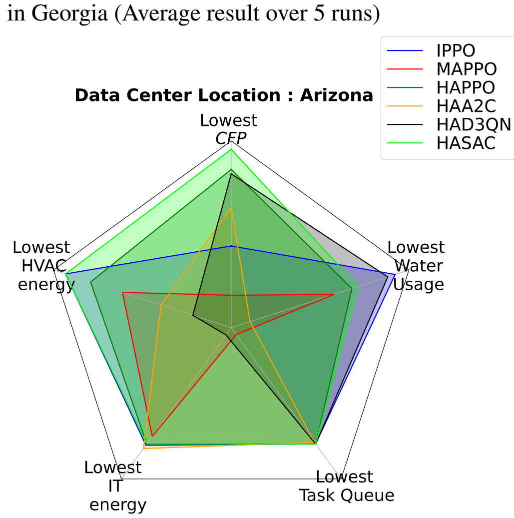
(c) Multiagent RL frameworks for a data center located in New York (Average result over 5 runs)



(d) Multiagent RL frameworks for a data center located in Georgia (Average result over 5 runs)



(e) Multiagent RL frameworks for a data center located in California (Average result over 5 runs)



(f) Multiagent RL frameworks for a data center located in Arizona (Average result over 5 runs)

Figure 5: Benchmarking RL Algorithms on the Sustain DC environment

6.3 Multiagent Benchmarks

We evaluated and compared the relative performances of various MARL approaches, including PPO with independent actor-critics (IPPO, $\alpha = 0.8$), centralized critic PPO (MAPPO), heterogeneous multi-agent PPO (HAPPO), HAA2C, HAD3QN, and HASAC. Figures 5c, 5d, 5e, and 5f illustrate the relative performance of these methods for DCs located in New York, Georgia, California, and Arizona. Our results reveal a consistent trend where PPO-based shared actor-critic methods (MAPPO, HAPPO) outperform the independent agent counterpart, IPPO. On further analysis, we observed that while IPPO effectively reduces HVAC and IT energy, the battery agent struggles to optimally schedule charging and discharging from the grid to meet data center demand. Among MAPPO, HAPPO, and HAA2C, HAPPO consistently performs best (except in Georgia).

For the off-policy methods (HAD3QN and HASAC), performance varies significantly across regions, with HASAC achieving the highest performance in Arizona. The reasons for these regional performance variations are not fully understood and may be partially due to differences in weather and carbon intensity. We plan to further investigate these variations in future work.

7 Limitations

The absence of an oracle that already knows the best results possible for the different environments makes it difficult to quantify the threshold for performance compared to simpler environments. For computational speed in RL, we used reduced order models for certain components like pumps and cooling towers. We could not exhaustably tune the hyperparameters for all the networks.

8 Next Steps

We are planning to deploy the trained agents to real data centers and are working towards domain adaptation for deployment with safeguards. We will augment the codebase with these updates. In order to have a smooth integration with current systems where HVAC runs in isolation, we plan a phased deployment with recommendation to the data center operative followed by direct integration of the control agents with the HVAC system with safeguards. For real-world deployment, a trained model should be run on a production server using appropriate checkpoints within a containerized platform with necessary dependencies. Security measures must restrict the software to only read essential data, generate decision variables, and write them with limited access to secure memory for periodic reading by the data center's HVAC management system. To ensure robustness against communication loss, a backup mechanism for generating decision variables is essential.

9 Conclusion

This paper introduced SustainDC, a fully Python-based benchmarking environment for multi-agent reinforcement learning (MARL) in sustainable, cost-effective, and energy-efficient data center operations. SustainDC provides comprehensive customization options for modeling multiple aspects of data centers, including a flexible RL reward design, an area we invite other researchers to explore further. We benchmarked an extensive collection of single-agent and multi-agent RL algorithms in SustainDC across multiple geographical locations, comparing their performance to guide researchers in sustainable data center management with reinforcement learning.

Additionally, we are collaborating with consortiums like ExaDigiT, which focuses on high-performance computing (HPC) and supercomputing, as well as with industry partners, to implement some of these approaches in real-world scenarios. SustainDC's complexity and constraints, rooted in realistic systems, make it a suitable platform for benchmarking hierarchical RL algorithms. We plan to implement continual reinforcement learning to accommodate dynamic data center environments and prevent out-of-distribution errors during equipment upgrades and accessory changes.

Moreover, SustainDC features an extendable, plug-and-play architecture of data center modeling compatible with digital twin frameworks, supporting research into other aspects of data center optimization for joint and multi-objective goals.

Acknowledgement

We would like to thank Paolo Faraboschi for sharing his expertise in machine learning and practical implementation approaches, and Torsten Wilde for his feedback on energy optimization and sustainability.

Additionally, we extend our gratitude to Wes Brewer, Feiyi Wang, Vineet Kumar, Scott Greenwood, Matthias Maiterth, and Terry Jones of Oak Ridge National Laboratory for their feedback and leadership within the ExaDigiT consortium, which helped refine our solution.

References

[1] J. R. Vázquez-Canteli, J. Kämpf, G. Henze, and Z. Nagy, "Citylearn v1.0: An openai gym environment for demand response with deep reinforcement learning," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '19, (New York, NY, USA), p. 356–357, Association for Computing Machinery, 2019.

[2] P. Scharnhorst, B. Schubnel, C. Fernández Bandera, J. Salom, P. Taddeo, M. Boegli, T. Gorecki, Y. Stauffer, A. Peppas, and C. Politi, "Energym: A building model library for controller benchmarking," *Applied Sciences*, vol. 11, no. 8, 2021.

[3] T. Moriyama, G. D. Magistris, M. Tatsubori, T. Pham, A. Munawar, and R. Tachibana, "Reinforcement learning testbed for power-consumption optimization," *CoRR*, vol. abs/1808.10427, 2018.

[4] J. Jiménez-Raboso, A. Campoy-Nieves, A. Manjavacas-Lucas, J. Gómez-Romero, and M. Molina-Solana, "Sinergym: A building simulation and control framework for training reinforcement learning agents," in *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, (New York, NY, USA), p. 319–323, Association for Computing Machinery, 2021.

[5] C. Yeh, V. Li, R. Datta, Y. Yue, and A. Wierman, "Sustaingym: A benchmark suite of reinforcement learning for sustainability applications," in *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track. PMLR*, p. 1, 2023.

[6] D. B. Crawley, L. K. Lawrie, C. O. Pedersen, and F. C. Winkelmann, "Energy plus: energy simulation program," *ASHRAE journal*, vol. 42, no. 4, pp. 49–56, 2000.

[7] M. Wetter, W. Zuo, T. S. Nouidui, and X. Pang, "Modelica buildings library," *Journal of Building Performance Simulation*, vol. 7, no. 4, pp. 253–270, 2014.

[8] W. Zuo, M. Wetter, J. VanGilder, X. Han, Y. Fu, C. Faulkner, J. Hu, W. Tian, and M. Condor, "Improving Data Center Energy Efficiency Through End-to-End Cooling Modeling and Optimization. Final Report," Apr. 2021. [Online; accessed 14. Oct. 2024].

[9] K. Sun, N. Luo, X. Luo, and T. Hong, "Prototype energy models for data centers," *Energy and Buildings*, vol. 231, p. 110603, 2021.

[10] Alibaba Group, "Alibaba production cluster data." `https://github.com/alibaba/clusterdata`, 2017. Accessed: 2024-06-05.

[11] Google, "Google cluster workload traces." `https://github.com/google/cluster-data`, 2019. Accessed: 2024-06-05.

[12] B. Acun, B. Lee, F. Kazhamiaka, K. Maeng, U. Gupta, M. Chakkaravarthy, D. Brooks, and C.-J. Wu, "Carbon explorer: A holistic framework for designing carbon aware datacenters," in *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ACM, Jan. 2023.

[13] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?," 2020.

[14] G. C. Team, "HPE announces industry's first 100% fanless direct liquid cooling systems architecture," *Hewlett Packard Enterprise*, Oct. 2024.

[15] K. Zhang, D. Blum, H. Cheng, G. Paliaga, M. Wetter, and J. Granderson, "Estimating ashrae guideline 36 energy savings for multi-zone variable air volume systems using spawn of energyplus," *Journal of Building Performance Simulation*, vol. 15, no. 2, pp. 215–236, 2022.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[17] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," in *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[18] Y. Zhong, J. G. Kuba, X. Feng, S. Hu, J. Ji, and Y. Yang, "Heterogeneous-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 25, no. 32, pp. 1–67, 2024.

[19] S. Sarkar, V. Gundecha, S. Ghorbanpour, A. Shmakov, A. R. Babu, A. Naug, A. Pichard, and M. Cocho, "Function approximation for reinforcement learning controller for energy from spread waves," in *IJCAI '23: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 6201–6209, Unknown publishers, Aug. 2023.

[20] S. Sarkar, V. Gundecha, A. Shmakov, S. Ghorbanpour, A. R. Babu, P. Faraboschi, M. Cocho, A. Pichard, and J. Fievez, "Multi-agent reinforcement learning controller to maximize energy efficiency for multi-generator industrial wave energy converter," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 12135–12144, Jun. 2022.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the introduction are shown in mainly across the SustainDC Overview and Benchmarking sections

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please see section Limitations

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Since this is an paper with an extensive set of benchmarking experiments, we provide the experimental details for reproducibility in the supplemental document.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We are providing the links to the code, documentation and data in the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the details to the experimental settings in the supplemental as well as the linked codebase

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: In this paper we do not show any results that are worth statistical significance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: This is provided in the Evaluation Metrics and Experimental Settings section

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper is mainly focused on sustainable data center computing and as such aspects of this are discussed in the Introduction and Conclusion

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer:[NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use open source datasets, certain repositories that are cited and our own models for developing the environment.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We provide the documentation

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer:[NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: [NA]

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.