Framework for Utilizing Multiple Data Sources for Reinforcement Learning in Data Center Cooling

Carlos Moustafa AEM278
Dylan Van Bramer DCV26
Oliver Lopez OJL23
Daniel Cao DYC33

Abstract

We introduce a novel framework for optimizing data center cooling using multiple data sources, including on the Marconi100 dataset and simulated environments. Our approach integrates data processing, efficient simulator implementation, and a unique interface between observed data and simulators. We leverage neural network-based imitation learning to learn reward functions, adapting these methods to the specific challenges of data center cooling. The framework's effectiveness is demonstrated through empirical validation with the Marconi100 dataset, highlighting its potential for energy-efficient and sustainable data center management. Our contributions include streamlined data processing and simulator integration with feature selection. This work not only paves the way for more efficient data center operations but also sets a precedent for future research in large-scale data center optimization. We hope our repository provides good modularity and scalability with any new datacenter dataset to run simulations on in the future. Please see our code in this repository: `https://github.com/oliverjdlopez/6784-final`.

1. Introduction

Alongside a growing body of literature surrounding energy conscious computation Schwartz et al. (2019), there is need for accessible and open-source datasets about datacenter energy usage, power consumption, and general operations. Inspired by the work of recent simulated environments and datasets in the NeurIPS Datasets and Benchmarks Track with a focus on computational sustainability, our work aims to support future development in optimal cooling for data centers, with potential software solutions for varying data availability. We consider two primary data types: a fully observed dataset documenting data center operations, with information including power consumption, specific job loads, building operations; and a configuration file that describes the physical representation of a given data center, from building specifics (cooling systems, geographical location) to relative hardware locations in the building.

We define, in general terms, "optimal cooling," as a series of actions that account for safety and environmental sustainability while not detracting from the full operational power of modern data centers. The precise algorithms and processes define these "actions" varies throughout our paper. Our definition of "optimal cooling" encompasses actions that ensure safety, environmental sustainability, and operational efficiency. We introduce a neural network-based imitation learning framework to learn the reward function, adapting these methods to the unique challenges of data center cooling. The framework leverages the Marconi100 dataset for empirical validation, demonstrating its effectiveness in real-world scenarios.

Data centers cannot get too hot, or the equipment will be damaged. However, the power consumption used by cooling resources is both expensive and damaging to the climate. As such, safe control is paramount. There has been much work in reinforcement learning for data center cooling, but there has not yet been a data-center specific method to minimize the risk of the exploration phase. This paper uses supervised learning on the newly released Marconi100 dataset to warm-start a policy which is then evaluated in a data center physics simulator (the simulated data-center is not from Marconi100). We investigate this transfer learning method to see if arbitrary data centers can be better controlled using a policy from a different data center location.

This research contributes to the field by providing a scalable, adaptable framework for data center cooling optimization. It combines data processing techniques with efficient simulator implementation and a novel interface between observed data and simulators. Our approach gives a pathway to more energy-efficient data center operations by utilizing both simulation and offline data together, rather than apart.

2. Related Work

The existence of this framework is dependent largely on the availability of open-source data center operations data, truthfully something that is still uncommon. Within our proposed framework, we utilize the newly released Marconi100 (M100) dataset Andrea Borghesi (2023), a nearly comprehensive set of operations data from the Marconi100 supercomputer, hosted by CINECA, the cross-Italy consortium on supercomputing. Indeed, the relevancy of our framework is made clear by the recent publication of this dataset, in 2023. To our knowledge, there seems to be only one current paper that uses this dataset, Antici et al. (2023) focusing on job power consumption and creating an alternative PM100 dataset, extracted from the original M100 set. This paper publishes code that allows for analysis of job load, and very limited job prediction. We extend this work, allowing extended functionality for operational predictions, as well as a streamlined approach to interface with a simulator.

Other literature shows the use of simulators for reinforcement learning to optimize cooling decisions, since a simulator offers system exploration via actions. Chervonyi et al. (2022) uses a physics simulation to model industrial-scale cooling policies with reinforcement learning, allowing for policy evaluation. Other approaches directly explore dynamics and employ models within *real* large-scale physical data centers, though of course these require extraordinarily high operational costs, and thus are achievable only by large institutions like Google. Gao (2014); Luo et al. (2022)

Our final contribution, which links algorithms that learn from tabular data, to simulations in which policies can be explored, is similar to another line of work in which models are (attempted to be) made more accurate by interfacing them, or ”grounding them” with physical simulations. Linkerhägner et al. (2023)

3. Framework Description:

3.1. Data Processing and Hosting for Fully Observed Data (M100)

The first contribution of this paper is a developed system for querying the M100 Dataset Andrea Borghesi (2023). We first establish simple processes for data loading and cleaning. While

M100 Variable	Simulator Actuator	Qualitative Meaning	Unit	Obs/Act
weather.Temp	Weather Data, Outdoor Dry-bulb, Environment	outdoors climate, which affects cooling energy use	deg C	OBS
logics.Tot_cdz	Electricity: HVAC	total power used by CRACs (air conditioning)	W	OBS
logics.Tot_chiller	Electricity: HVAC	total power used by the Chillers (cooling unit)	W	OBS
logics.Tot_qpompe	Electricity:Plant	total power used by liquid cooling unit	KW	OBS
vertiv.Ext_Air_Sensor_A_Temperature	Zone Mean Air Temperature", "WEST ZONE"	current temperature in west zone	deg C	OBS
vertiv.Ext_Air_Sensor_A_Humidity	Zone Air Relative Humidity", "WEST ZONE"	current humidity in west zone	% rh	OBS
vertiv.Ext_Air_Sensor_C_Temperature	Zone Mean Air Temperature", "EAST ZONE"	current temperature in east zone	deg C	OBS
vertiv.Ext_Air_Sensor_C_Humidity	Zone Air Relative Humidity", "EAST ZONE"	current humidity in east zone	% rh	OBS
vertiv.Supply_Air_Temperature_1	"System Node Setpoint Temperature" , "WEST AIR LOOP OUTLET NODE"	output temperature from AC in west zone	deg C	OBS
vertiv.Return_Air_Temperature_1	System Node Temperature, WEST ZONE INLET NODE	temperature of air in west zone at AC location	deg C	OBS
vertiv.Supply_Air_Temperature_2	"System Node Setpoint Temperature" , "EAST AIR LOOP OUTLET NODE"	output temperature from AC in east zone	deg C	OBS
vertiv.Return_Air_Temperature_2	System Node Temperature, EAST ZONE INLET NODE	temperature of air in east zone at AC location	deg C	OBS
vertiv.Fan_Speed_1	Mass Flow Rate Setpoint,WEST AIR LOOP OUTLET NODE	fan intensity in west zone	%, m^3/sec	ACT
vertiv.Fan_Speed_2	Mass Flow Rate Setpoint,EAST AIR LOOP OUTLET NODE	fan intensity in east zone	%, m^3/sec	ACT
vertiv.Supply_Air_Temperature_Set_Point_1	WEST AIR LOOP OUTLET NODE	desired output temperature from AC in west zone	deg C	ACT
vertiv.Supply_Air_Temperature_Set_Point_2	EAST AIR LOOP OUTLET NODE	desired output temperature from AC in west zone	deg C	ACT

Table 1: One-to-one relationship for interfacing between simulation and observed data

many rudimentary data processing utilities were provided by Andrea Borghesi (2023) upon the release of the dataset, our work enables a more efficient pipeline from hosting data to training models. In particular, it performs Parquet-file data processing and cleaning based on timestamp, node, and selected features. We propose a modular, configurable framework for downloading and merging M100 data features into a single dataframe of instances.

Example notebooks provided by the M100 dataset crashed commercial laptops with 16GB of RAM. We therefore also offer dataset simplification. We provide functionality to sample one node from each server rack, or to choose only a certain number of racks, thus reducing memory use by a large factor while maintaining important spatial information. Our querying framework gives more intuitive and expressive indexing, while being memory conscious. The user is able to perform operations on chunks of the data that are as large as memory allows, and then combine the results. The features that we used for our experiments are given above in the table. Many metrics did not have overlapping timestamps, so we grouped all timestamps by the minute. Further, any duplicate timestamps from the same metric were dropped.

3.2. Efficient Implementation of Simulators

We propose a method for training and evaluating a neural network policy model on Parquet-file data. Additionally, we propose a method for fine-tuning these pre-trained policy models on an EnergyPlus simulator.

Since obtaining large amounts of data in this domain is expensive, our proposed method for learning a policy in one data center and fine-tuning it in others is valuable. Of course, since the variation of data center layouts is significant, a pre-trained model's generalization performance is important. Later, in Section 3.3 and 5, we show via experimentation, that pre-training a neural network model on the M100 data set lead to a significant reward improvement compared to a model solely trained in a simulator.

3.3. Interface Between Observed Data and Simulators

In cases where both physical simulations are possible *and* fully observed operational data is present, the framework we provide can be used fully. We begin by creating a one-to-one relationship between variables in the fully observed dataset, and actuators in our simulator. The specific relationship defined in Table 1 is a feasible configuration of this interface, with observation and action space each multi-dimensional, but low-dimensional enough to minimize computational costs. The network is first trained on the M100 data. Then, a network with either identical, or similar, architecture is defined as a subclass of RayRllib's TorchModelV2. The state dictionary parameters are loaded into the new model, which can then be used by Ray for training.

In our framework, PPO is used to train agents that can make optimal cooling decisions based on real-time data. The algorithm iteratively improves its policy by taking actions in the simulated environment, observing the results, and adjusting its strategy to maximize the cumulative reward. This reward is computed based on factors like energy efficiency, maintaining environmental conditions within desired thresholds, and other features in the config it is to optimize. Furthermore, our framework supports pretrained PPO models or by default trains a new PPO learner per simulation. We also define a Multi-Layer Perception policy in our PPO but custom policies can also be loaded in.

The integration of PPO into our simulator environment allows for the development of robust and efficient cooling strategies. By simulating various scenarios and learning from the outcomes, the PPO-trained agents can discover innovative approaches to optimize cooling in data centers, potentially leading to significant energy savings and reduced environmental impact.

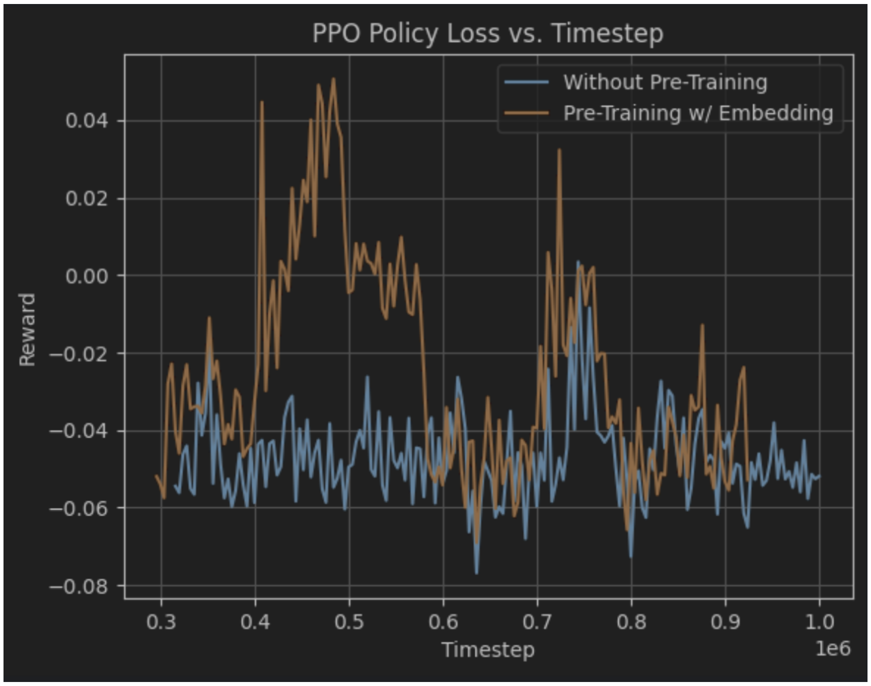

Figure 1: Performance of Models With and Without Using Pretraining (Combining Data Sources)

4. Methods and Experimentation

In our investigation, we aimed to understand how the newly defined state and action space could be utilized to learn effective policies and rewards. The experiments were conducted on a Nvidia RTX 4070 GPU.

Data Preprocessing The experiments were based on a subset of the M100 dataset.

For our experiments, we focused on key variables representing the system in a low-dimensional subspace, simplifying the model and facilitating the learning process. The selected variables included ambient temperatures from different sensors, external temperature, and air temperatures related to the cooling system. The data was preprocessed to align timestamps and ensure consistency across different data sources. Additionally, it was important that we selected features that map directly to observable features in the EnergyPlus-simulated data centers, so that we could test our fine-tuning protocol.

Model Architecture A neural network model was implemented to predict a 4-dimensional action controlling HVAC temperature setpoints and air flow rates–crucial control variables in the data center's cooling system. The model architecture was defined as follows:

- A linear layer with input dimension equal to the number of features (14) and an output dimension of 128.
- Batch normalization followed by a ReLU activation function.
- Another linear layer reducing the dimensionality to half of the previous layer.

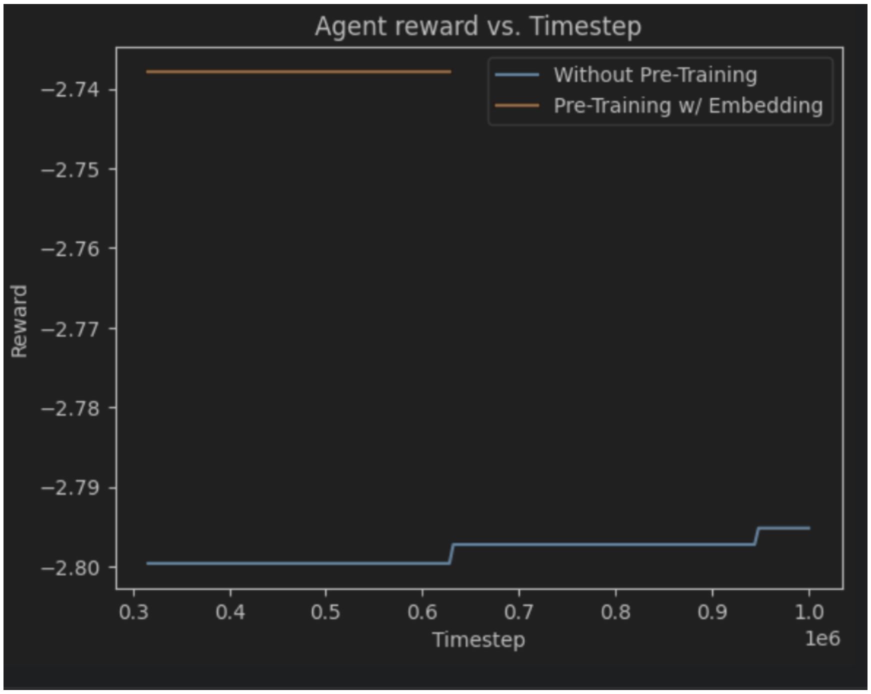

Figure 2: Agent Reward Over Time With and Without Using Pretraining (Combining Data Sources)

- A final linear output layer.

Pre-training Process The model was pre-trained using the following procedure:

1. The dataset was split into training and testing sets.
2. A DataLoader was used to iterate over the dataset in batches.
3. The model was trained for 25 epochs using Adam with a learning rate of 1×10^{-3}.
4. The Mean Squared Error (MSE) loss function was used to evaluate the model's performance.

Fine-Tuning Process After running the described pre-training procedure, we fine-tuned the model on an EnergyPlus data center simulator using the following method:

1. The model was initialized using the pre-trained weights.
2. An extra Linear output layer was added to the model.
3. The model was trained further using PPO for 10,000,000 timesteps.

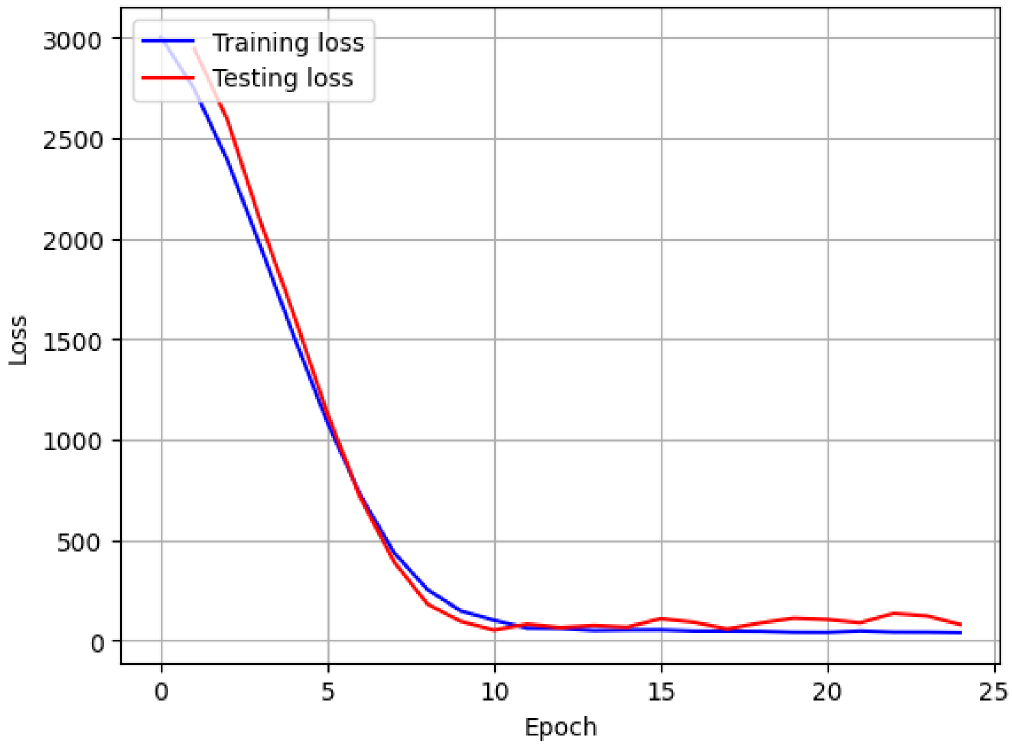

Figure 3: Test and Training Loss of M100 Data Model in Predicting Policy Over Epochs

5. Evaluation Metrics

The model's performance was evaluated based on the loss over epochs, indicating its accuracy and efficiency in learning the underlying patterns in the data. The average training and testing losses were plotted against the number of epochs to visualize the learning progress.

The first, most naive method we approached the problem from was that of a neural network for supervised learning, on the fully observed data we defined above. More specifically, we take a features vector (with each of the variables listed in the "state space" above), at time t, and predict the action (at time $t+1$). In our initial approach to optimizing data center cooling, we implemented a neural network for supervised learning using PyTorch, focusing on the fully observed data. The model's performance is evaluated every epoch, assessing its generalization capabilities and effectiveness in predicting future states under varying conditions.

6. Conclusion and Future Directions

In this paper, we presented a comprehensive framework for utilizing multiple data sources in the context of data center cooling using the RayLib EnergyPlus Shaw et al. (2021) Liang et al. (2018) simulation on the novel M100 dataset. Our approach integrates a variety of techniques, including neural network-based prediction and imitation learning, to optimize cooling strategies in data centers. The use of the Marconi100 dataset as a primary data source has enabled us to develop and test our models with real-world data, ensuring the practical applicability of our methods. Additionally, our method for pre-training a model and transferring its learned knowledge to a simulated environment enables easier future work in this domain, as it effectively decreases the need for large amounts of costly data.

6.1. Key Contributions

We developed a streamlined process for handling and processing fully observed data center operation data, enhancing the efficiency of data utilization. Secondly, we created an interface between observed data and simulators, allowing for more accurate and realistic simulations that closely mirror real-world scenarios. Lastly, we applied advanced neural networks and policy gradient learning techniques, such as Proximal Policy Optimization (PPO), to predict and optimize cooling strategies. These contributions collectively represent a significant advancement in the field, offering new tools and methodologies for energy-efficient data center management.

6.2. Limitations and Future Work

While our framework demonstrates promising results, there are limitations that need to be addressed in future work. These include expanding the feature set to include more variables that can influence cooling strategies, enhancing the scalability of our models to handle larger and more complex data centers, integrating real-time data feeds to enable dynamic and adaptive cooling strategies, and exploring the integration of renewable energy sources and their impact on cooling strategies.

For future research, it would be interesting to compare the performance of, through the field of AI planning, deep RL, other recent algorithms such as Actor to Critic (A2C) and Deep Q Learning on baseline policies. We also would like to capture the physical dynamics relations within a datacenter and use that to run imitation learning algorithms such as Dagger and Iterative Quadratic Regulator to be able to generalize an optimal policy and learn other environments like different configurations of a data center. In conclusion, our work lays the foundation for future research and development in the area of data center cooling optimization. We believe that the continued exploration and enhancement of this framework will lead to significant advancements in the field, contributing to the overall goal of creating more efficient, sustainable, and cost-effective data centers through much more accurate simulations.

References

Martin Molan et. al Andrea Borghesi, Carmine Di Santi. M100 exadata: a data collection campaign on the cineca's marconi100 tier-0 supercomputer. 2023. doi: 0.1038/s41597-023-02174-3. URL https://doi.org/10.1038/s41597-023-02174-3.

Francesco Antici, Mohsen Seyedkazemi Ardebili, Andrea Bartolini, and Zeynep Kiziltan. Pm100: A job power consumption dataset of a large-scale production hpc system. In *Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, SC-W '23, page 1812–1819, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400707858. doi: 10.1145/3624062.3624263. URL https://doi.org/10.1145/3624062.3624263.

Yuri Chervonyi, Praneet Dutta, Piotr Trochim, Octavian Voicu, Cosmin Paduraru, Crystal Qian, Emre Karagozler, Jared Quincy Davis, Richard Chippendale, Gautam Bajaj, Sims Witherspoon, and Jerry Luo. Semi-analytical industrial cooling system model for reinforcement learning, 2022.

Jim Gao. Machine learning applications for data center optimization, 2014.

Eric Liang, Richard Liaw, Philipp Moritz, Robert Nishihara, Roy Fox, Ken Goldberg, Joseph E. Gonzalez, Michael I. Jordan, and Ion Stoica. Rllib: Abstractions for distributed reinforcement learning, 2018.

Jonas Linkerhägner, Niklas Freymuth, Paul Maria Scheikl, Franziska Mathis-Ullrich, and Gerhard Neumann. Grounding graph network simulators using physical sensor observations, 2023.

Jerry Luo, Cosmin Paduraru, Octavian Voicu, Yuri Chervonyi, Scott Munns, Jerry Li, Crystal Qian, Praneet Dutta, Jared Quincy Davis, Ningjia Wu, Xingwei Yang, Chu-Ming Chang, Ted Li, Rob Rose, Mingyan Fan, Hootan Nakhost, Tinglin Liu, Brian Kirkman, Frank Altamura, Lee Cline, Patrick Tonker, Joel Gouker, Dave Uden, Warren Buddy Bryan, Jason Law, Deeni Fatiha, Neil Satra, Juliet Rothenberg, Mandeep Waraich, Molly Carlin, Satish Tallapaka, Sims Witherspoon, David Parish, Peter Dolan, Chenyu Zhao, and Daniel J. Mankowitz. Controlling commercial cooling systems using reinforcement learning, 2022.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green ai, 2019.

Rachael Shaw, Enda Howley, and Enda Barrett. Applying reinforcement learning towards automating energy efficient virtual machine consolidation in cloud data centers. *Information Systems*, 107:101722, 01 2021. doi: 10.1016/j.is.2021.101722.