Learning a Data Center Model for Efficient Demand Response

QUENTIN CLARK, Boston University, USA
FATIH ACUN, Boston University, USA
IOANNIS C. PASCHALIDIS, Boston University, USA
AYSE COSKUN, Boston University, USA

Data center demand is projected to increase dramatically over the coming decades, creating concerns about their carbon footprint and motivating the design of methods that can scale data center capabilities sustainably. One such method is Demand Response (DR), which provides incentives for power consumers to regulate their consumption in compliance with sustainability and capacity needs in the grid. One limitation of existing work in data center DR is the computational expense of generating accurate average power estimates and flexibility forecasts for a data center, given knowledge about the data center's configuration and activity. We introduce CONDOR, a machine learning (ML) method to learn the relationship between a data center's configuration (e.g., power, performance, and load characteristics) and an objective function incorporating DR savings, energy cost, and workload quality-of-service (QoS) compliance. CONDOR optimizes power and flexible reserve estimates that minimize an objective function, helping the data center efficiently meet compliance with sustainability measures. Our results demonstrate that CONDOR achieves speed increases of around 15,000x in computing accurate forecasts compared to simulation-based estimation, which enables DR participation of large real-world data centers in DR programs without debilitating computational overhead.

CCS Concepts: • **Hardware** → **Power estimation and optimization**; • **Computing methodologies** → *Neural networks*; • **Computer systems organization** → *Reliability*.

Additional Key Words and Phrases: Demand Response, Data Centers, Sustainability, Machine Learning

1 INTRODUCTION

Large data centers are challenged with balancing the requirements of their consumers, making fiscally responsible energy purchases, and operating sustainably. In tandem, power providers are faced with the problem that their power output (supply) does not always match demand. This forces a choice where power providers must either scale up their infrastructure to meet peak demand, which is often much larger than average demand, or fail to meet the power delivery requirements of consumers. Often, new grid infrastructure is reliant on fossil fuels, which contributes to carbon emissions and global warming. The importance of this problem is increasing as data centers account for a larger percent of total energy use due to the projected increase in energy from AI applications (e.g., generative AI). [41].

One approach to alleviating data center peak power usage is demand response (DR), where a consumer modulates its power use (at various time scales) to meet power provider requirements, increasing grid power output flexibility. Data centers are especially qualified for DR participation due to their ability to quickly and accurately regulate their power usage through job scheduling and server power management [2, 6]. The increased flexibility provided by high-power consumers like data centers allows for more ambitious renewable deployment [1, 43]. Two specific mechanisms providing this sustainability benefit are preventing intermittency from disrupting stability [18] and enabling scaling compute without requiring new fossil-fuel infrastructure to meet power peaks [9].

While data center DR has seen success in real-world deployment [31, 47], and there is promising work in providing DR methods with theoretical guarantees [55], an open problem in integrating these methods into real data centers is the lack of efficient solutions to replace computationally heavy simulation-based optimization methods. To allow wider adoption of DR by data centers, there is a need for fast and scalable models of data centers to provide accurate estimations of their future state.

This paper introduces CONDOR (**C**ost-**O**ptimization **N**eural-Network for **D**ata Center **O**perational Demand **R**esponse), a machine learning (ML)-based approach to learning a model of a data center for DR participation that replaces laborious simulation methods. Our key contributions are as follows:

- Designing a deep learning model to approximate the cost of a data center execution for DR.
- Incorporating this model into a fast optimization process to output future power and reserve estimates for a data center.
- Demonstrating our method learns a correct approximation of the data center DR cost, and showing our optimization process can find high-quality DR parameters at a fraction of the inference time compared to simulation.

Through experiments over various configurations of data centers with different workload types, we achieve a 4 orders of magnitude reduction in execution time when predicting data center DR parameters compared to using a simulation of a data center, going from a scale of several hours to a second. Our method aims to enable the use of DR in data centers of any size without introducing computational overhead. This allows for data centers to more easily participate in DR programs and provide crucial flexibility for grid sustainability projects.

We begin with a discussion of our ML-based approach for data center DR in Section 2, followed by results in Section 3. We then discuss the related work in Section 4 and conclude in Section 5.

2 ML-BASED PARAMETER ESTIMATION FOR DATA CENTER DEMAND RESPONSE

Data center sustainability has been studied from various perspectives, such as their carbon footprint and influence on grid stability. Solutions explore shifting the workloads to less carbon-intensive times of the day [39] and migrating those to regions with a surplus of renewable energy [30, 56]. Other studies investigate adjusting

Authors' addresses: Quentin Clark, Boston University, 8 St Mary's St, Boston, Massachusetts, USA, 02215; Fatih Acun, Boston University, 8 St Mary's St, Boston, Massachusetts, USA, 02215; Ioannis C. Paschalidis, Boston University, 8 St Mary's St, Boston, Massachusetts, USA, 02215; Ayse Coskun, Boston University, 8 St Mary's St, Boston, Massachusetts, USA, 02215.

power consumption to comply with the power providers' objective of balancing the power supply and demand [19, 28]. Participating in such programs requires data centers to forecast their average power consumption and power reserve (where "reserve" refers to the flexibility of consumption a data center provides above and below its average power) for certain future intervals [32]. While providing a high reserve bid increases potential monetary benefits, a high average power bid results in high electricity costs. Another dimension that needs to be considered as a cost for the data center is the need to satisfy quality-of-service (QoS[1]) requirements while operating under limited power, resulting from predicted power and reserve bids and the power provider's regulation needs.

2.1 Estimating Power and Reserve Bids

Our goal is to provide fast and accurate forecasts of average power consumption and power reserve amounts for data centers so that data centers can participate in DR programs such as the regulation service reserves program, where participants need to estimate their parameters for the hour-ahead market [32][2]. An overview of our method is shown in Figure 1. We replace the existing simulation-based methods that require long execution times to forecast accurate parameters. For example, the Adaptive QoS-Assurance (AQA) work [55] uses a data center simulator with an adaptive policy that ensures QoS through job scheduling and power capping. In the AQA framework, data centers periodically provide their expected average future power use $\bar{P}$ and their power reserve capacity R to power providers. This reserve capacity means the power provider will frequently ask the data center to increase or decrease its power use within that reserve amount, with economic penalties if the data center cannot comply. The power and reserve bids, $\bar{P}$ and R, can be calculated by optimizing an objective function that reflects the monetary cost of a data center participating in DR such as the regulation service reserves program:

$$C = \Pi^{\mathrm{P}} C_{\mathrm{Power}} + \Pi^{\mathrm{E}} C_{\mathrm{Error}} + \Pi^{\mathrm{Q}} C_{\mathrm{QoS}}, \tag{1}$$

$$C_{\mathrm{Power}} = (\bar{P} - \Pi^{R} R), \tag{2}$$

$$C_{\mathrm{Error}} = E[|P(t) - P_{target}(t)|], \tag{3}$$

$$C_{\mathrm{QoS}} = \beta \sum_{j} \mathrm{SoftPlus}(\rho(\mathrm{Prob}[Q^{j} - Q^{j}_{thres}] - \delta^{j})), \tag{4}$$

where $\bar{P}$ is the average power of the data center, R is the provided reserve capacity, $P(t)$ is the data center's current power, $P_{target}(t)$ is the power target given by the Independent Service Operator (ISO), each Π term is a weighting constant determined by monetary costs in the energy market, and H represents the time interval in hours. The SoftPlus function $\ln(1 + e^{x})$ penalizes the data center for jobs violating QoS over a certain threshold. The terms β and ρ are the weighting parameters for the QoS constraints. The QoS term is found by comparing $Q^{j} = \frac{T_{so} - T^{j}_{min}}{T^{j}_{min}}$, where T_{so} is the sojourn time (the actual time for job completion) and T^{j}_{min} is the time for processing a job without power caps or queuing delays, to a pre-defined allowable threshold. The summation over j sums over each job type in the workload W. The probability term over the QoS constraints frames QoS in a probabilistic form, where only violations up to $\delta^{j}\%$ of the time are considered acceptable.

Each term in the cost is weighed by a separate weighing parameter Π^{P},Π^{R}, Π^{E}, and Π^{Q}, which are tuned to ensure the scale of each component is similar. Each job type is given a proportion of servers equal to w_j, which cumulatively add up to 1 to represent the full data center. We call the vector of these job weight parameters W.

The AQA framework iteratively runs simulations of a data center's operation to calculate a stochastic estimate of C, while optimizing $\bar{P}$, R, and the job weights W through gradient descent (GD). This is enabled by treating the data center problem as a queuing-theoretic control problem, which allows for a closed-form approximation of the cost function gradient to be calculated. This makes AQA an hour-scale DR program (bids are generated once per-hour) but it runs a second-scale policy to determine which jobs are given execution priority and specific power capping settings of servers.

However, using simulated data centers can be computationally expensive enough to be infeasible for large data centers under certain DR programs. For our hour-scale DR program, new $\bar{P}$, R, bids and W job weights must be generated at the top of each hour as the data center's workload or target utilization changes as a function of the time of day. Other DR programs require minute or second-level bids [20, 21, 38]. If the bidding process using the simulator takes longer than the cycle time of the DR program, than that solution is practically infeasible to use.

Our data center model CONDOR replaces the simulator with a substantially cheaper ML model of the objective function. To learn a model of the objective function, we treat the data center simulator as an oracle for sampling the cost function at different data center configurations, generate a diverse dataset from this oracle, and then perform supervised learning over this data using a neural network.

The input features of CONDOR is a description of the configuration of our data center. This includes the $\bar{P}$ and R bids, a desired utilization ratio (how much of the data center servers should be utilized at a given time), information about the workload mix (including job weights W), and the server size. The model outputs an estimate of each component of the cost function described above, which are then weighed using the Π^{P}, Π^{E}, and Π^{Q} terms. Then, we use CONDOR to find accurate $\bar{P}$, R, and W values for a given data center configuration. This is done by calculating the gradient of the model inputs $\bar{P}$,R, and W with respect to the weighted output C, performing one step of gradient descent (projecting W onto the unit sphere), and repeating, using our updated bids as new inputs to the model, shown in Algorithm 1.

Using GD on the model inputs with respect to the output cost is enabled by the fact that functions represented by neural networks are differentiable through backpropagation. Our method is computationally efficient because each CONDOR GD step takes milliseconds compared to several minutes for the simulator. Because the AQA cost function is convex, we find optimizing over our learned cost

[1]Meeting QoS refers to execution of submitted jobs within given time deadlines, which we treat as a constraint for the delay tolerance of each job.

[2]Although our method could be applied generically (see Section 2.2), we focus on the regulation service reserves program because of its compatibility with renewable-enabling programs [36].

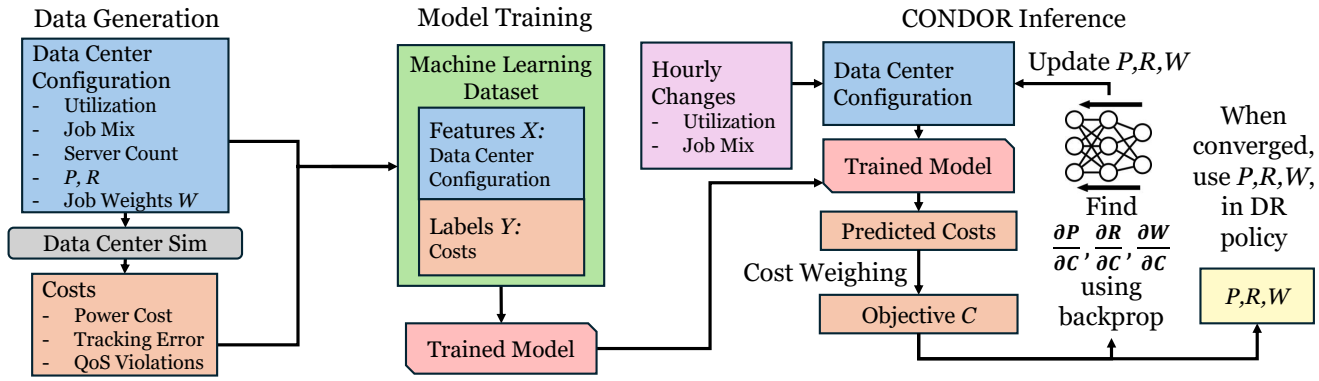

Fig. 1. An overview of our method. We generate data from a computational simulation of a data center, train a neural network to learn the relationship between the data center configuration and demand-response participation cost, then optimize for the flexible $\bar{P}, R,$, and W parameters before each hour.

Algorithm 1 Demand-Response Bidding with CONDOR

Require: CONDOR Model f, server configuration S including workload mix of size N, cost function weights Π^P, Π^E, Π^Q, learning rate η

Initialize:

$\bar{P} = 1, R = 0.6, W = [w_1, w_2, ...w_n] = \frac{1}{N}$

while Not Converged **do**

$C_{Power}, C_{Error}, C_{QoS} \leftarrow f(S, \bar{P}, R, W)$ ▷ Model Estimation

$C \leftarrow \Pi^P C_{Power} + \Pi^E C_{Error} + \Pi^Q C_{QoS}$ ▷ Cost Weighing

Use backprop to find $\frac{\partial \bar{P}}{\partial C}, \frac{\partial R}{\partial C}, \frac{\partial W}{\partial C}$ ▷ GD Update

$z \leftarrow$Softmax$(W - \frac{\partial W}{\partial C}) - W$ ▷ Job Weight Projection

Update $\bar{P} \leftarrow \bar{P} - \eta \frac{\partial \bar{P}}{\partial C}, R \leftarrow R - \eta \frac{\partial R}{\partial C}, W \leftarrow W + \eta z$

end while

return $\bar{P}, R, W$ for use in AQA runtime policy

function did not suffer from getting stuck in saddle points or local minima.

An important consideration at inference time is properly weighing the three components of the cost function (Π^P, Π^E and Π^Q). Because the AQA framework treats the bid/weight search process as a constrained optimization problem, where the solution optimizes for the power and error cost within QoS constraints, in practice we set Π^Q to a much larger value than Π^P and Π^E.

2.2 A Recipe for Creating Data Center Models

While we only present a model *bespoke* to a specific DR strategy (i.e., participating in regulation service reserves markets), our approach of learning an power consumer model to use for optimizing DR parameters is agnostic to the DR formulation, the parameters optimized, and the specific power consumer. For instance, a data center participating in market-based DR program [40] could use our approach to optimize for price bids. To help others apply this technique, we describe a short recipe for using our method with any DR program.

Deep learning models are constructed from blocks which process data with certain properties. These blocks transform data from one form into a feature-vector. The most common approach is to use these blocks with the appropriate feature data, collect the feature vectors generated by each block, combine the information from each feature vector (called “fusion"), and then pass them through several final multi-layer perceptron (MLP) layers. Common blocks include Transformers [48] or RNNs [54] for sequences, CNNs for image [27], and graph networks for data equivariant E(3) transformations [7]. We use aggregate fusion, where the feature vectors are concatenated, but more sophisticated approaches exist [52] [13]. Our problem's cost function is convex, and we found optimizing over our learned cost did not converge to poor local minima. For problems where the cost function is more complex, imposing an input-convexity constraint on the output through the neural network architecture is popular [3].

Broadly, the recipe is as follows. First, create a dataset of power consumer configurations with the corresponding demand response program cost. This can be sampled from a simulator or a real-world consumer. Next design a model, using appropriate blocks for invariances or feature covariances in the configuration data. Train the model to predict the cost from the consumer configuration using a modern first-order optimizer. Finally, at inference time, keep gradients of the configuration parameters you wish to optimize for and update them with gradient descent. Figure 1 shows a visual overview of this procedure.

In our case, our features are the information about our data center (or other energy consumer) that can be changed, and that will impact the ability of the consumer to participate in the DR program.

2.3 Model Details

2.3.1 Architecture Details. One important consideration in data center DR is handling different workload mixes. Each job type is described by a power performance profile (maximum power, minimum power, etc.), which is represented by a fixed-length vector. Workload mixes are sets of these job types, so order permutations of workload mixes cannot be treated differently by our model. Workload mixes can also be different sizes, so our model must be able to process workload mixes of arbitrary length. Our model includes a small Set Transformer block [25], which is permutation-invariant

and variable-input length compatible. This is followed by fully-connected residual layers [42]. We found that residual layers prevents the smaller tabular configuration information from being "forgotten" by the later layers [17].

2.3.2 Training and Implementation Details. We implement CONDOR in PyTorch [37], which enables variables to keep a computational graph of their derivative with respect to other variables. In our case, the $\bar{P}$ and R input bids and job weights W have their derivatives with respect to the cost tracked, allowing for simple implementation of gradient descent on the network's input.

We train our model to minimize mean-squared error (MSE) loss over our cost vector containing C_{Power}, C_{Error}, and C_{QoS}. C_{Power} and C_{Error} are divided by the number of servers, and C_{QoS} by the number of jobs in the workload mix. This means our model only needs to learn the server/job average component costs. While our model could learn these relationships by itself, we found this normalization improved the shape of each cost component's distribution and accelerated training. We use the AdamW optimizer [29] for 150 epochs, with a learning rate of $1e-4$ and batch size of 512. To validate our model's ability to generalize, we split our data into a 70-30 train/test split and found a final MSE difference of only around 2%.

Considering the importance of reducing carbon costs in training ML models, we calculate an estimate of the carbon it took to train CONDOR [11]. Our model is small due to not requiring many layers to encode features (as in language or vision) and only took around 6.78 minutes to train on a laptop with an RTX 4080 GPU and an Intel i9-13950HX, using at most $2.616e-5$ metric tons of Co2, roughly equal to the power needed to charge a smartphone twice. The carbon costs on a production data center would be slightly higher from the operator running the model to find bids and retraining the model with new data collected from the data center's operation. However, our very low carbon estimate shows that our model is lightweight enough to have a negligible impact on energy usage.

2.4 Data Generation

We generate training data from the AQA simulator [55]. We sample over a range of workload mixes, utilization ratios, server counts, and $\bar{P}$ and R values.

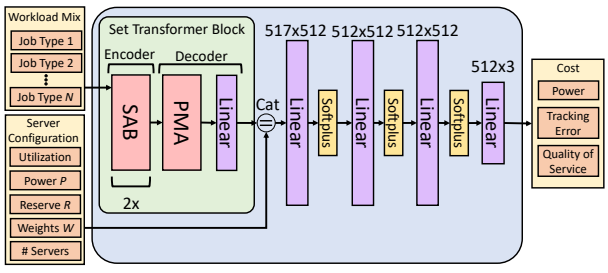

Fig. 2. The architecture of our deep learning model. The set transformer block generates a feature vector of the workload mix, which is fused with other simulator configuration information, including the desired parameters to optimize. Residual connections are omitted for clarity.

To ensure our feature space is well-sampled, we vary the following simulator parameters:

- Workload Mix W: we include 12 different mixes, including small variations of other mixes in our dataset. Mixes are designed to sample the range of possibilities, including longer/shorter jobs, more/fewer types of jobs, etc. Section 2.4.1 describes our workloads in more detail.
- $\bar{P}$ and $\bar{R}$ bids: we randomly sample these over a feasible set from $[0.2, 0.6]$ kW per/server for $\bar{P}$, and $[0.0, 0.12]$ kW per/server for R
- Workload weights W: we randomly sample these, and apply a softmax function to ensure the total weights add to 1.
- Data center size: we vary the size of our simulated data centers from the set $\{64, 100, 500, 1000\}$.
- Data center utilization: we vary the utilization ratio from the set $\{0.5, 0.75, 0.8, 0.85, 0.9, 0.95\}$.

Additionally, to sample the parameter space near *optimal bids* more densely, we also include intermediate steps of the AQA gradient descent optimization process in our training data. This ensures that the model learns a general relationship through the randomized data, but is specifically focused on high-quality predictions near the optimal bid for each configuration. We collected 80,809 samples of training data.

2.4.1 Workload Mix Representation. We represent our workload mixes as lists of job types, each of which is a fixed-length vector that describes the power/performance profile of an individual data center job. These features include the following: m_j (the number of servers used to run the job), T_{min} and T_{max} (the minimum and maximum processing time in seconds), p_{min} and p_{max} (the minimum and maximum power consumption in watts), and Q_{thres} (the quality of service threshold discussed in Section 2.1).

Because of CONDOR's invariance to the order of the job types in each workload (Section 2.3) we do not vary them during training. To handle the difference in magnitude of the job type features, we normalize the features by dividing each by that feature's empirical average in our dataset before being processed by CONDOR.

Our workload mixes are taken from prior work in data center DR [55]. These mixes were chosen to ensure a high diversity of potential workloads. These include mixes focused on short and long jobs, more and less types of jobs, higher and lower power, and incrementally adding more jobs to a mixed-type workload. For our experiments in Section 3 we specifically used workloads W3 (large server requirements), W4 (short application mix), W5 (long application mix), W6 (small number of applications), W7 (large number of applications), and W8 (low power applications).

3 RESULTS

Our experiments show that CONDOR generates $\bar{P}$, R, and W bids of similar quality to AQA at a fraction of the time. We ran the AQA optimization process using both the simulator and CONDOR over six different workloads taken from recent work [55] representing a diverse possibility of mixes, for the same data center configuration representing a data center with 500 servers and a target server utilization ratio of 0.8. This utilization ratio represents a typical scenario of a data center wishing to use most of its compute, but

Workload Mix \ Method	$\bar{P}$ (kW)		R (kW)		Execution Time		Norm. Cost	% Violation	
	Simulator	Model	Simulator	Model	Simulator	Model	Model	Simulator	Model
W3	160.7	175.4	26.3	31.2	236 m	**0.911 s**	1.171	12.5%	0%
W4	154.3	159.1	21.1	33.4	610 m	**0.814 s**	**0.980**	0%	0%
W5	154.4	147.3	23.5	26.4	531 m	**0.790 s**	**0.920**	0%	0%
W6	175.1	166.8	31.5	29.4	613 m	**0.828 s**	**0.95**	0%	0%
W7	159.5	171.6	23.9	29.9	547 m	**0.841 s**	1.119	0%	0%
W8	139.4	155.1	14.3	17.7	591 m	**0.822 s**	1.191	0%	0%

Table 1. The bids generated by the gradient descent optimization process for the simulator and model, and the running times for each. % Violation refers to the percentage of job types in that run which violated QoS. Norm. Cost refers to the cost of the model's bids normalized by dividing by the cost of the simulator's bid for the same workload mix (lower is better).

willing to provide some grid flexibility. For fairness, we kept the number of GD iterations (150) fixed for both. Our results showed a substantial speedup of around 15, 000x from CONDOR compared to the simulator (Table 1).

We then use the bids generated by the simulator and CONDOR and run them on the simulator with the AQA runtime policy for 1 simulated hour to compare how effectively the model's bids reduce cost within QoS constraints. Our results show that CONDOR finds $\bar{P}, R$, and W bids that do not incur substantial performance penalties compared to bids found using the simulator. Table 1 shows the exact bids found by both, and compares the bids with three metrics: the percentage of job types that violated QoS, the cost C of the runtime policy over the simulated hour, and the execution time. The cost is normalized to the cost that the original AQA optimization process produces, so values lower than 1 indicate CONDOR produced bids superior to AQA. CONDOR finds superior bids to the simulator on three of the tasks, and meets the QoS constraint for all six. In total, CONDOR only suffers a 5% average increase in cost, while dramatically decreasing the time to find these bids. This demonstrates that our model is not only efficient, but correctly optimizes the desired objective function within QoS constraints. In total, CONDOR finds bids that let a data center offer between 11% and 21%, with an average of 17%, of its average power consumption as flexible reserves to the power grid.

Crucially, this increased speed enables the AQA DR program to be run on larger data centers. Figure 3 shows that finding bids for large data centers (more than ~100 servers) is infeasible with the simulation-based approach for optimization, as the computation time quickly outpaces the timescale of the DR program. In contrast, CONDOR determines bids equally quickly regardless of data center scale, as the relationship between the data center size and cost is learned functionally instead of modeling each server individually.

Lastly, we generate plots of the cost function surface approximated by our model by sampling our model along a 12 by 12 grid representing possible simulator $\bar{P}/R$ bids to gain qualitative insight into the model's learned cost function. Figure 4 shows that the cost is more sensitive to the power bids than reserve bids, and that the costs of higher power bids are smaller than lower power bids. This is expected, as power bids insufficient to meet QoS incur large penalties, while higher power is only penalized by the economic costs of buying unneeded power. This relationship is consistent with the analysis of similar cost functions for similar DR programs [8]. Our plots also show an expected difference in the cost function between the larger workload mix W5 compared to the smaller workload mix W6. The W6 plot has a shallower slope on the low power end, caused by the fact that fewer job types incurs a smaller penalty to the cost function C. These observations indicate that our model learns the correct objective function to optimize over.

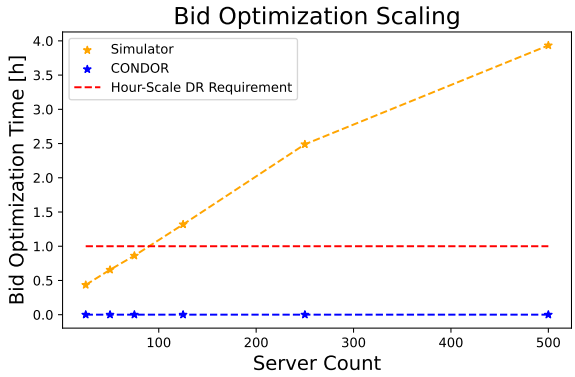

Fig. 3. Optimization time versus simulation data center size. This plot shows that simulations of large data centers scale linearly with time, which makes participating in lower time-frame DR intractable for large data centers.

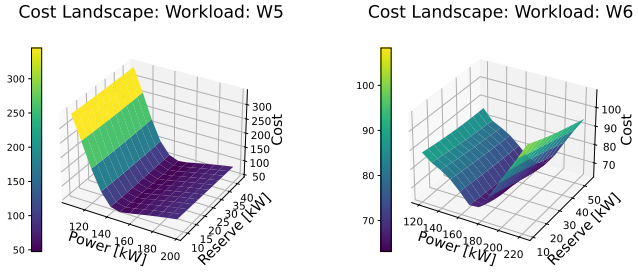

Fig. 4. Plots of the cost surface approximation generated by the model. The P/R space is sampled on a 12 by 12 grid.

4 RELATED WORK

There is robust work using ML for data center DR. Some approaches include classical ML, deep learning, Reinforcement Learning (RL), and evolutionary algorithms [4]. Most approaches use ML to forecast useful quantities used by a control policy. Examples include using SVMs and LSTMs [35], decision trees, [34], ensemble models [53], and shallow neural networks [57] to predict future power usage and/or future temperature. RL solutions are common, but often suffer from simple pricing models, single-agent assumptions, and limited fidelity environments [49].

Most work for data center DR leverages the opportunities from increased communication and load flexibility. Approaches include Vickrey–Clarke–Groves auctions [58], queuing for workload scheduling [5], energy management modeling [51], and bespoke models [45]. These systems, when an objective function is centrally optimized, are usually tuned using convex optimization or linear programming. Some use neural networks, including for weather forecasting to estimate solar power generation [45] and power usage [50].

Our approach can be thought of as a form of energy based model (EBM) [24], which learn an energy quantity that various inputs to the model can be optimized for. In our case, the demand response cost is a proxy for energy. Use of EBMs in control tasks are common [15, 44].

Our approach shares similarities to model predictive control (MPC), except that our system has no temporal dependence (past bids do not affect future bids), so we do not use planning or dynamic programming. Existing DR approaches incorporating learning algorithms in their MPC models includes SVMs to learn temperature predictions [46] and neural networks to predict control signals from a simulator ground-truth predictor[16].

There is existing work in optimizing learned cost functions in inverse control [10, 22] and RL [23]. Using neural network backpropagation to modify the input with respect to the network output was inspired by the image domain [14] and was originally applied in music generation [26].

5 CONCLUSION AND FUTURE WORK

Current DR methods for data centers show great potential in helping increase data center's sustainability by providing renewable-enhancing grid flexibility and room to scale compute without requiring new power infrastructure. To fix the intractable computational expense of prior methods that provide data center power and reserve forecasts, this paper introduced an ML-based approach that is computationally inexpensive and scales well to large data centers. Our experiments showed that CONDOR offers similar quality DR parameters for $15,000x$ lower computational expense.

There are several avenues for future work. Our paper focused on modeling a data center with a mostly fixed per-server configuration (e.g., fixed idle node power use). A future model could be extended to also learn how to optimize for data centers with more sophisticated configuration flexibility. To bypass concerns about simulation accuracy or fidelity, future work could learn directly from a real-world data center instead of a simulation.

Our specific DR formulation, while powerful, is not exhaustive. For instance, while we can adjust job QoS thresholds and execution times to approximate jobs with tight latency restrictions, new DR works may benefit from modeling latency explicitly. Similarly, we assume compute homogeneity, which ignores how certain jobs may require certain compute resources that are limited in the data center. For instance, AI training requires GPU/TPU resources, which not every server in the data center may possess. Future work could focus on expanding the simulation and representation of the DR problem to incorporate these elements.

This study learned an approximation of a known closed-form cost function, but it could be used for cost functions without a known functional form, similar to the common practice in RL of using ML to learn a reward function from human preference [33]. For instance, extensions of this method might include training a similar model on a more sophisticated evaluation metric sampled from a real-world data center's operation[12].

Lastly, our work focuses on comparing different optimization strategies within the same data center simulator. Future work could extend this comparison to real-world data centers, by running the AQA policy with both the original simulation-based and CONDOR bidding strategies on a real-world workload.

ACKNOWLEDGEMENTS

We thank the reviewers for their helpful feedback which improved the final version of the paper. We thank Dr. Daniel C. Wilson for helpful discussions and feedback. This work was partially supported by the Research Council of Norway award 344115, 2023-2027, and the Boston University STARS and UROP programs.

REFERENCES

[1] Jamshid Aghaei and Mohammad-Iman Alizadeh. 2013. Demand response in smart electricity grids equipped with renewable energy sources: A review. *Renewable and Sustainable Energy Reviews* 18 (2013), 64–72.
[2] Ilari Alaperä, Samuli Honkapuro, and Janne Paananen. 2018. Data centers as a source of dynamic flexibility in smart girds. *Applied energy* 229 (2018), 69–79.
[3] Brandon Amos, Lei Xu, and J Zico Kolter. 2017. Input convex neural networks. In *International conference on machine learning*. PMLR, 146–155.
[4] Ioannis Antonopoulos, Valentin Robu, Benoit Couraud, Desen Kirli, Sonam Norbu, Aristides Kiprakis, David Flynn, Sergio Elizondo-Gonzalez, and Steve Wattam. 2020. Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review. *Renewable and Sustainable Energy Reviews* 130 (2020), 109899.
[5] Shahab Bahrami, Yu Christine Chen, and Vincent WS Wong. 2019. An online algorithm for data center demand response. In *2019 53rd Annual Conference on Information Sciences and Systems (CISS)*. IEEE, IEEE, Baltimore, MD 21218, 1–6.
[6] Robert Basmadjian. 2019. Flexibility-based energy and demand management in data centers: a case study for cloud computing. *Energies* 12, 17 (2019), 3301.
[7] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. 2022. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications* 13, 1 (2022), 2453.
[8] Hao Chen, Yijia Zhang, Michael C Caramanis, and Ayse K Coskun. 2019. EnergyQARE: QoS-aware data center participation in smart grid regulation service reserve provision. *ACM Transactions on Modeling and Performance Evaluation of Computing Systems (TOMPECS)* 4, 1 (2019), 1–31.
[9] Andrew A Chien, Chaojie Zhang, and Liuzixuan Lin. 2022. Beyond PUE: Flexible datacenters empowering the cloud to decarbonize. In *Proceedings of the 1st Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon'22)*. Association for Computing Machinery, New York, NY, USA, 1–6.
[10] Peter Englert and Marc Toussaint. 2016. Combined Optimization and Reinforcement Learning for Manipulation Skills. In *Proceedings of Robotics: Science and Systems*. RSS Foundation, AnnArbor, Michigan, 1–9.
[11] US EPA. 2024. Greenhouse Gas Equivalencies Calculator. https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator#results

[12] A Gandhi, K Ghose, K Gopalan, S Hussain, D Lee, Y Liu, Z Liu, P McDaniel, S Mu, and E Zadok. 2022. Metrics for Sustainability in Data Centers. In *Proceedings of the 1st Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon'22)*. Association for Computing Machinery, New York, NY, USA, 1–6.

[13] Jing Gao, Peng Li, Zhikui Chen, and Jianing Zhang. 2020. A survey on deep learning for multimodal data fusion. *Neural Computation* 32, 5 (2020), 829–864.

[14] Google Research 2015. *Inceptionism: Going Deeper into Neural Networks*. Google Research. Retrieved March 25, 2024 from https://blog.research.google/2015/06/inceptionism-going-deeper-into-neural.html

[15] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. 2017. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*. PMLR, 1352–1361.

[16] Ouzhu Han, Tao Ding, Chenggang Mu, Yuhan Huang, Xiaosheng Zhang, and Zhoujun Ma. 2023. Multi-Time Scale Optimal Dispatch for the Wind Power Integrated System With Demand Response of Data Centers Based on Neural Network-Based Model Predictive Control. *IEEE Transactions on Industry Applications* 59, 6 (2023), 7238–7249.

[17] Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov, Przemysław Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, et al. 2020. Urban driving with conditional imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Association for Computing Machinery, New York, NY, United States, 251–257.

[18] IEA 2023. *Demand response - IEA*. IEA. Retrieved January 17, 2024 from https://www.iea.org/energy-system/energy-efficiency-and-demand/demand-response#tracking

[19] Ali Jahanshahi, Nanpeng Yu, and Daniel Wong. 2022. PowerMorph: QoS-Aware Server Power Reshaping for Data Center Regulation Service. *ACM Trans. Archit. Code Optim.* 19, 3, Article 36 (aug 2022), 27 pages.

[20] Tingyu Jiang, CY Chung, Ping Ju, and Yuzhong Gong. 2022. A multi-timescale allocation algorithm of energy and power for demand response in smart grids: A Stackelberg game approach. *IEEE Transactions on Sustainable Energy* 13, 3 (2022), 1580–1593.

[21] Harjeet Johal, Krishna Anaparthi, and Jason Black. 2012. Demand response as a strategy to support grid operation in different time scales. In *2012 IEEE Energy Conversion Congress and Exposition (ECCE)*. IEEE, Raleigh, NC, USA, 1461–1467.

[22] Arezou Keshavarz, Yang Wang, and Stephen Boyd. 2011. Imputing a convex objective function. In *2011 IEEE International Symposium on Intelligent Control*. IEEE, Denver, CO, USA, 613–619.

[23] Jeongho Kim, Jaeuk Shin, and Insoon Yang. 2021. Hamilton-jacobi deep q-learning for deterministic continuous-time systems with lipschitz continuous controls. *Journal of Machine Learning Research* 22, 206 (2021), 1–34.

[24] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. 2006. A tutorial on energy-based learning. *Predicting structured data* 1, 0 (2006).

[25] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set Transformer: A Framework for Attention-based Permutation-Invariant Neural Networks. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, CA, United States, 3744–3753. https://proceedings.mlr.press/v97/lee19d.html

[26] John Peter Lewis. 1988. Creation by refinement: a creativity paradigm for gradient descent learning networks. In *IEEE 1988 International Conference on Neural Networks*. IEEE, San Diego, CA, USA, 229–233 vol.2.

[27] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. 2021. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems* 33, 12 (2021), 6999–7019.

[28] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, and Niangjun Chen. 2013. Data center demand response: avoiding the coincident peak via workload shifting and local generation. *SIGMETRICS Perform. Eval. Rev.* 41, 1 (jun 2013), 341–342.

[29] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. arXiv:1711.05101 [cs.LG]

[30] Diptyaroop Maji, Ben Pfaff, Vipin PR, Rajagopal Sreenivasan, Victor Firoiu, Sreeram Iyer, Colleen Josephson, Zhelong Pan, and Ramesh K Sitaraman. 2023. Bringing Carbon Awareness to Multi-cloud Application Delivery. In *Proceedings of the 2nd Workshop on Sustainable Computer Systems*. Association for Computing Machinery, New York, NY, USA, 1–6.

[31] Varun Mehra and Raiden Hasegawa. 2023. Using demand response to reduce data center power consumption | google cloud blog. https://cloud.google.com/blog/products/infrastructure/using-demand-response-to-reduce-data-center-power-consumption

[32] New York Independent System Operator (NYISO). 2020. *Ancillary Services Manual, v6.0*. NYISO. [Online]. Available: https://www.nyiso.com/manuals-tech-bulletins-user-guides.

[33] Malayandi Palan, Nicholas C. Landolfi, Gleb Shevchuk, and Dorsa Sadigh. 2019. Learning Reward Functions by Integrating Human Demonstrations and Preferences. arXiv:1906.08928 [cs.RO]

[34] Fabiano Pallonetto, Mattia De Rosa, Federico Milano, and Donal P Finn. 2019. Demand response algorithms for smart-grid ready residential buildings using machine learning models. *Applied energy* 239 (2019), 1265–1282.

[35] Fabiano Pallonetto, Changhong Jin, and Eleni Mangina. 2022. Forecast electricity demand in commercial building with machine learning models to enable demand response programs. *Energy and AI* 7 (2022), 100121.

[36] Ioannis Ch Paschalidis, Binbin Li, and Michael C Caramanis. 2011. A market-based mechanism for providing demand-side regulation service reserves. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 21–26.

[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), Vol. 32. Curran Associates, Inc., Vancouver, BC, Canada. https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

[38] S Ali Pourmousavi, M Hashem Nehrir, and Ratnesh K Sharma. 2015. Multi-timescale power management for islanded microgrids including storage and demand response. *IEEE Transactions on Smart Grid* 6, 3 (2015), 1185–1195.

[39] Ana Radovanović, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, Saurav Talukdar, Eric Mullen, Kendal Smith, MariEllen Cottman, and Walfredo Cirne. 2023. Carbon-Aware Computing for Datacenters. *IEEE Transactions on Power Systems* 38, 2 (2023), 1270–1280.

[40] Farrokh Rahimi and Ali Ipakchi. 2010. Demand response as a market resource under the smart grid paradigm. *IEEE Transactions on smart grid* 1, 1 (2010), 82–88.

[41] Timothy Rooks. 2022. Data centers keep energy use steady despite big growth. https://www.dw.com/en/data-centers-energy-consumption-steady-despite-big-growth-because-of-increasing-efficiency/a-60444548

[42] Muhammad Shafiq and Zhaoquan Gu. 2022. Deep residual learning for image recognition: A survey. *Applied Sciences* 12, 18 (2022), 8972.

[43] Farshid Shariatzadeh, Paras Mandal, and Anurag K Srivastava. 2015. Demand response for sustainable energy systems: A review, application and implementation strategy. *Renewable and Sustainable Energy Reviews* 45 (2015), 343–350.

[44] K Sharma, Bhopendra Singh, Edwin Herman, R Regine, S Suman Rajest, and Ved P Mishra. 2021. Maximum information measure policies in reinforcement learning with deep energy-based model. In *2021 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*. IEEE, 19–24.

[45] Seyed Mohammad Sheikholeslami, Amir Masoud Rabiei, Mahmoud Mohammad-Taheri, and Jamshid Abouei. 2022. Cloud data center participation in smart demand response programs for energy cost minimisation. *IET Smart Grid* 5, 5 (2022), 380–394.

[46] Rui Tang, Cheng Fan, Fanzhe Zeng, and Wei Feng. 2021. Data-driven model predictive control for power demand management and fast demand response of commercial buildings using support vector regression. *Building Simulation* 15 (06 2021), 1–15.

[47] Zachary Tzavelis. 2020. *Decarbonizing New York's Power Sector*. The Cooper Union for the Advancement of Science and Art.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc., Long Beach, CA, United States. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[49] José R Vázquez-Canteli and Zoltán Nagy. 2019. Reinforcement learning for demand response: A review of algorithms and modeling techniques. *Applied energy* 235 (2019), 1072–1089.

[50] Andreea Valeria Vesa, Tudor Cioara, Ionut Anghel, Marcel Antal, Claudia Pop, Bogdan Iancu, Ioan Salomie, and Vasile Teodor Dadarlat. 2020. Energy flexibility prediction for data center engagement in demand response programs. *Sustainability* 12, 4 (2020), 1417.

[51] Dongxiao Wang, Changhong Xie, Runji Wu, Chun Sing Lai, Xuecong Li, Zhuoli Zhao, Xueqing Wu, Yi Xu, Loi Lei Lai, and Jinxiao Wei. 2021. Optimal energy scheduling for data center with energy nets including CCHP and demand response. *IEEE Access* 9 (2021), 6137–6151.

[52] Yikai Wang, Wenbing Huang, Fuchun Sun, Tingyang Xu, Yu Rong, and Junzhou Huang. 2020. Deep multimodal fusion by channel exchanging. *Advances in neural information processing systems* 33 (2020), 4835–4845.

[53] Xinran Yu and Semiha Ergan. 2022. Estimating power demand shaving capacity of buildings on an urban scale using extracted demand response profiles through machine learning models. *Applied Energy* 310 (2022), 118579.

[54] Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural computation* 31, 7 (2019), 1235–1270.

[55] Yijia Zhang, Daniel Curtis Wilson, Ioannis Ch Paschalidis, and Ayse K Coskun. 2021. HPC data center participation in demand response: an adaptive policy with QoS assurance. *IEEE transactions on sustainable computing* 7, 1 (2021), 157–171.

[56] Jiajia Zheng, Andrew A. Chien, and Sangwon Suh. 2020. Mitigating Curtailment and Carbon Emissions through Load Migration between Data Centers. *Joule* 4, 10 (2020), 2208–2222. https://www.sciencedirect.com/science/article/pii/S2542435120303470

[57] Yuekuan Zhou and Siqian Zheng. 2020. Machine-learning based hybrid demand-side controller for high-rise office buildings with high energy flexibilities. *Applied Energy* 262 (2020), 114416.

[58] Zhi Zhou, Fangming Liu, Shutong Chen, and Zongpeng Li. 2018. A truthful and efficient incentive mechanism for demand response in green datacenters. *IEEE Transactions on Parallel and Distributed Systems* 31, 1 (2018), 1–15.