Journal of Carbon Neutrality
http://dx.doi.org/10.62762/JCN.2026.592975

RESEARCH ARTICLE

A Bayesian-Optimization-Based Method for Exploring the Energy Optimization Potential of Data Centers

Simian Gu[1], **Yuxuan Wang**[1], **Shunyao Zhu**[1], **Lei Yu**[1,*] and **Frédéric Magoulès**[2]

[1] Sino-French Engineer School, Beihang University, Beijing 100191, China
[2] CentraleSupelec, Universite Paris-Saclay, Gif-sur-Yvette 91192, France

Abstract

With the expansion of artificial intelligence, data center energy consumption has surged, making its optimization critical for achieving carbon neutrality. Traditional fixed-rule controls rely too heavily on manual experience to explore the highest efficiency, while standard AI black-box models often lack interpretability and present evaluation challenges. This study introduces a Bayesian optimization-based methodology to explore the energy optimization potential of data center cooling systems. We enhanced the Resource Allocation and Power Simulator (RAPS) to generate dynamic workloads reflecting the authentic statistical characteristics of supercomputers for rigorous stress testing. This generator integrates with a model of the Frontier supercomputer cooling systems via a Functional Mock-up Unit (FMU) interface, enabling co-simulation. A context-informed Gaussian Process Bayesian optimization algorithm was designed to optimize upper-level setpoints piecewise, targeting Power Usage Effectiveness (PUE) under temperature penalty constraints, to uncover the data center's energy optimization potential across distinct control variables and dynamic workloads. Experimental results under a 24-h random workload demonstrate that optimizing the cooling tower temperature correction parameter (adj) reduced average PUE from 1.1164 to 1.1104—a 0.54% decrease consistent in magnitude with comparable studies—validating the method's efficacy. Extended testing reveals that the primary differential-pressure setpoint provides the largest and most consistent optimization benefits, whereas the pump stage-up threshold shows weak and workload-dependent effects, and the stage-down threshold provides modest but consistently positive improvements. This research establishes a high-fidelity, interpretable gray-box testing framework for control strategy pre-evaluation and key variable screening in large-scale data centers.

Keywords: Bayesian optimization, data center cooling, energy optimization, digital twin, control strategy.

1 Introduction

This section establishes the research context and motivation. It first highlights the escalating energy

Submitted: 08 April 2026
Revised: 08 August 2026
Accepted: 20 August 2026
Published: 03 September 2026

Vol. 1, **No.** 2, 2026.
10.62762/JCN.2026.592975

***Corresponding author:**
Lei Yu
yulei@buaa.edu.cn

Citation
Gu, S., Wang, Y., Zhu, S., Yu, L., & Magoulès, F. (2026). A Bayesian-Optimization-Based Method for Exploring the Energy Optimization Potential of Data Centers. *Journal of Carbon Neutrality*, 1(2), 97–125.

© 2026 by the Authors. Published by Institute of Central Computation and Knowledge. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

consumption of data centers and the imperative for carbon neutrality. Subsequently, it reviews current cooling optimization strategies and their inherent limitations. Finally, it introduces the proposed Bayesian-optimization-based methodology for evaluating optimization potential and outlines the primary contributions of this paper.

1.1 Data Centers for Carbon Neutrality

With the advancement of technologies such as cloud computing and big data, computational power has long been recognized as a novel commercial resource [1]; this trend has been further accelerated by the rapid proliferation of artificial intelligence and large language models [2]. Consequently, commercial data centers and supercomputers have exhibited an explosive trend in construction, continuously expanding in scale; although improved energy efficiency has partially offset this growth, the absolute energy demand of data centers continues to rise [3]. In this context, environmental protection imperatives and the goals of carbon peaking and carbon neutrality have imposed increasingly stringent requirements on data center construction and operation.

The development of data centers is inextricably linked to carbon emissions and the pursuit of carbon neutrality. According to the report [2] published by the International Energy Agency (IEA), the global electricity consumption of data centers was estimated at approximately 415 terawatt-hours (TWh) in 2024, generating about 180 million tons of indirect carbon dioxide emissions annually. With the continued proliferation of AI and related technologies, the power consumption of global data centers is projected to more than double by 2030 [2]. In reality, data center energy consumption is not solely driven by server hardware (e.g., CPUs and GPUs); the power drawn by cooling and power distribution systems is also substantial, contributing to significant additional carbon emissions. Consequently, optimizing data center cooling systems, assessing their efficacy, and uncovering the ultimate optimization potential represents a critical measure for achieving carbon reduction in these facilities.

1.2 Related Work

In recent years, liquid-cooled servers have become increasingly prevalent in large-scale data centers [4], and the optimization of cooling-control components—such as the design and characterization of flow control devices for dynamic liquid cooling of servers—has consequently attracted growing attention [5].

At present, data center cooling systems mainly rely on automatic control logic based on a "sensor–fixed-rule-based control system–actuator (control valve)" architecture [6, 7]. In such fixed-rule-based automatic control systems, the optimization principle lies in the existence of certain hyperparameters within the control logic that allow engineers to fine-tune the system according to the actual operating conditions of the data center and established thermal design guidelines [8], achieving efficient operation of the cooling system [9]. This optimization approach is simple and direct. However, it relies excessively on subjective human judgment and is constrained by human factors such as engineering experience, making it difficult to efficiently test and evaluate the optimization effect as well as the upper and lower performance bounds.

Furthermore, optimizing the control systems of data centers and liquid cooling systems using artificial intelligence has become a focal point of current research [10].

Several recent studies exemplify the breadth of AI-driven optimization approaches in energy-intensive systems. A comprehensive survey on data center cooling technology, power consumption modeling, and control strategy optimization provides a systematic foundation for understanding the design space of cooling-control variables [11]. Data-center-level model-predictive control combined with reinforcement learning has demonstrated safe and effective regulation of temperature and airflow with limited prior knowledge [12]. Deep reinforcement learning has further been shown to substantially reduce cooling energy consumption by transforming setpoint optimization into a sequential decision-making problem [13]. Bayesian optimization has also been formulated as a contextual optimization problem for HVAC setpoint control, demonstrating its sample-efficient applicability to building-level energy systems [14]. Collectively, these works illustrate the progression from survey-level understanding to algorithm-level application of AI optimization in cooling-system control.

Neural-network-based methods can be used to establish data-driven models of data centers by leveraging massive amounts of historical operational data, thereby enabling rapid and efficient evaluation of the influence of different control parameters on data center energy efficiency [15]. However, the training of neural networks requires a large amount of high-quality operational data, which limits their general applicability and ease of use. In addition, as a black-box model, the neural network also provides limited interpretability of the results.

After a dedicated training process, reinforcement-learning-based multi-agent multi-head models can substantially reduce the additional energy consumption of data center cooling systems and significantly improve the energy efficiency of data centers [16]. However, reinforcement learning is likewise a black-box system, and its interpretability as well as the evaluation of optimization performance remain limited. Moreover, it generally requires training a specific model for a specific data center under a specific workload condition for evaluation and optimization, making the overall process relatively complex and costly.

In conclusion, when addressing such complex engineering problems, many black-box artificial intelligence optimization methods suffer from inherent challenges, including difficulties in selecting optimization variables, acquiring training data, evaluating optimization effectiveness, and poor interpretability. Rigorously evaluating these optimization algorithms and the potential of optimization necessitates a sound methodological approach.

1.3 Bayesian-Optimization-Based Method for Exploring the Energy Optimization Potential of Data Centers

To overcome the aforementioned limitations and address the need for methods to evaluate the optimization potential of data centers, this study proposes employing grey-box optimization algorithms, which offer a degree of interpretability, to assess this potential; the grey-box paradigm has demonstrated broad applicability across energy-system domains, including building energy simulation [17], providing methodological precedent for its adoption in data center cooling optimization. However, conducting exploratory or extreme-load experiments directly on physical cooling systems poses unacceptable risks of irreversible hardware damage and service interruption. Consequently, this research plans to utilize digital models of data centers for simulation and experimentation. Nevertheless, relying solely on historical operational data typically fails to capture the highly controllable and extreme workloads required for comprehensive stress testing. Therefore, this paper intends to upgrade the thermal load simulation components to accommodate the requirements of extreme testing on the digital models of the data center.

Compared with those policy-learning approaches, the proposed Bayesian-optimization-based grey-box framework couples a deterministic physics-based evaluator with a sample-efficient Gaussian-process surrogate. Here, "grey-box" refers to the division of roles between these two components: the physics-based digital twin explicitly preserves the known thermo-fluid and control structure of the cooling system, while the Gaussian-process surrogate learns the relationship between operating context, candidate setpoints, and system performance from a limited number of simulation evaluations. The surrogate is therefore used to guide the search rather than to replace the underlying physical model. This combination enables interpretable setpoint screening and workload-dependent optimization-potential assessment with only a small number of costly simulations. It is therefore particularly suitable for safe pre-deployment analysis, whereas reinforcement learning may be more advantageous for continuous closed-loop control after sufficient safe interaction data and training are available.

Overall, the main contributions of this paper are as follows:

1. A method is proposed for exploring the energy optimization potential of data centers.
2. A data center workload generator consistent with statistical characteristics and probability distributions is developed and validated.
3. A context-aware Bayesian-optimization procedure is developed and integrated with the digital-twin simulation framework for exploring data-center energy optimization potential, and its effectiveness is validated.
4. The proposed Bayesian-optimization-based method is applied to investigate the optimization potential of several variables in the Frontier supercomputer cooling system.

2 A Methodology for Exploring Data Center Energy Optimization Potential Based on Bayesian Optimization

To explore the energy optimization potential of data center cooling systems when specific variables are selected as optimization targets, this paper proposes a methodology for evaluating the optimization potential of control variables in data center cooling systems, as shown in Figure 1.

This section presents the implementation of the proposed Bayesian-optimization-based methodology for exploring data-center energy optimization potential. The underlying Frontier cooling-system model is adopted from the ExaDigiT framework [18], and the present study uses it as the physics-based evaluator within the proposed optimization workflow. The methodology is described according to the functional roles of its three main components: the thermal workload generator, the cooling-system simulator, and the Bayesian optimizer. This component-wise organization is intended to clarify how the overall framework is constructed and operated, while the specific contributions of this study are summarized in Section 1.3.

2.1 Thermal Workload Generator

To provide controllable thermal inputs for simulation-based stress testing, this study extends the synthetic-workload path of the Resource Allocation and Power Simulator (RAPS) [18]. Recent HPC simulation frameworks have also emphasized standardized workload representations and simulation-based evaluation as important tools for studying workload-management behavior without relying exclusively on production-system experiments; representative work has proposed high-level simulation and workload data schema frameworks specifically designed to facilitate reproducible HPC workload management research [19]. In parallel, digital-twin approaches have provided physics-informed virtual environments for evaluating HPC system behavior and operational strategies under controlled conditions [20]. The resulting thermal workload generator combines statistically specified job streams with RAPS resource allocation and power models. Its purpose is to generate reproducible Frontier-scale test scenarios whose duration and computational intensity can be controlled independently of a particular historical period. Because private Frontier job logs are unavailable, the generator is not presented as a reconstruction of Frontier operations. Instead, the public Adastra MI250 15-day workload trace released by CINES [21] is used only as a cross-system external structural reference in Section 3.1.

RAPS simulates job submission, queueing, scheduling, node allocation, and power evolution at a 1 s time step. Each synthetic job contains a submit time, requested node count, intended wall time, and CPU and GPU utilization traces. The power model maps the scheduled nodes and utilization traces to node- and rack-level electrical loads. When the cooling-system Functional Mock-up Unit (FMU) is enabled, RAPS exchanges aggregated thermal loads for the 25 coolant distribution units and the required environmental inputs with the FMU every 15 s, and receives cooling-system states and performance metrics, including Power Usage Effectiveness (PUE).

For the Frontier configuration used in this study, interarrival times after initialization are independently sampled from an exponential distribution with a mean of 900 s, corresponding to a Poisson arrival process with an expected rate of four jobs per hour. The requested node count is sampled from a discrete uniform distribution between 1 and 3000 nodes. Before time quantization, the intended wall time is sampled from a normal distribution with mean 23,400 s and standard deviation 6600 s, truncated between 3600 and 43,200 s; the sampled value is then rounded down to a 3600 s quantum. These parameters are read from the Frontier scheduler configuration rather than fitted to the public reference trace.

Computational intensity is controlled through a separate utilization/power profile. For every job, CPU and GPU utilization fractions are sampled independently from the distribution associated with the selected profile: $\mathrm{Beta}(1,5)$ for the low profile, $\mathcal{U}(0,1)$ for the uniform profile, and $\mathrm{Beta}(5,1)$ for the high profile. The sampled fractions are converted to constant per-job CPU and GPU traces at 15 s resolution over the intended wall time. Thus, the three profiles are designed to modify computational and power intensity without intentionally changing the arrival, node-count, or wall-time rules. They should not be interpreted as low-, medium-, and high-node-occupancy categories.

For a requested horizon T, the generator first creates the requested number of candidate jobs and retains the jobs whose cumulative submit times fall within $[0, T)$. During simulation, submitted jobs may be completed, running, or queued at T. The output procedure therefore writes all three states to `job_history.csv`, rather than silently dropping submitted but unfinished jobs at the finite-horizon boundary. Jobs whose submit times lie beyond T are not included in the horizon-specific workload record. The corresponding `generated_workload.csv` stores the generated inputs, while `run_metadata.json` records

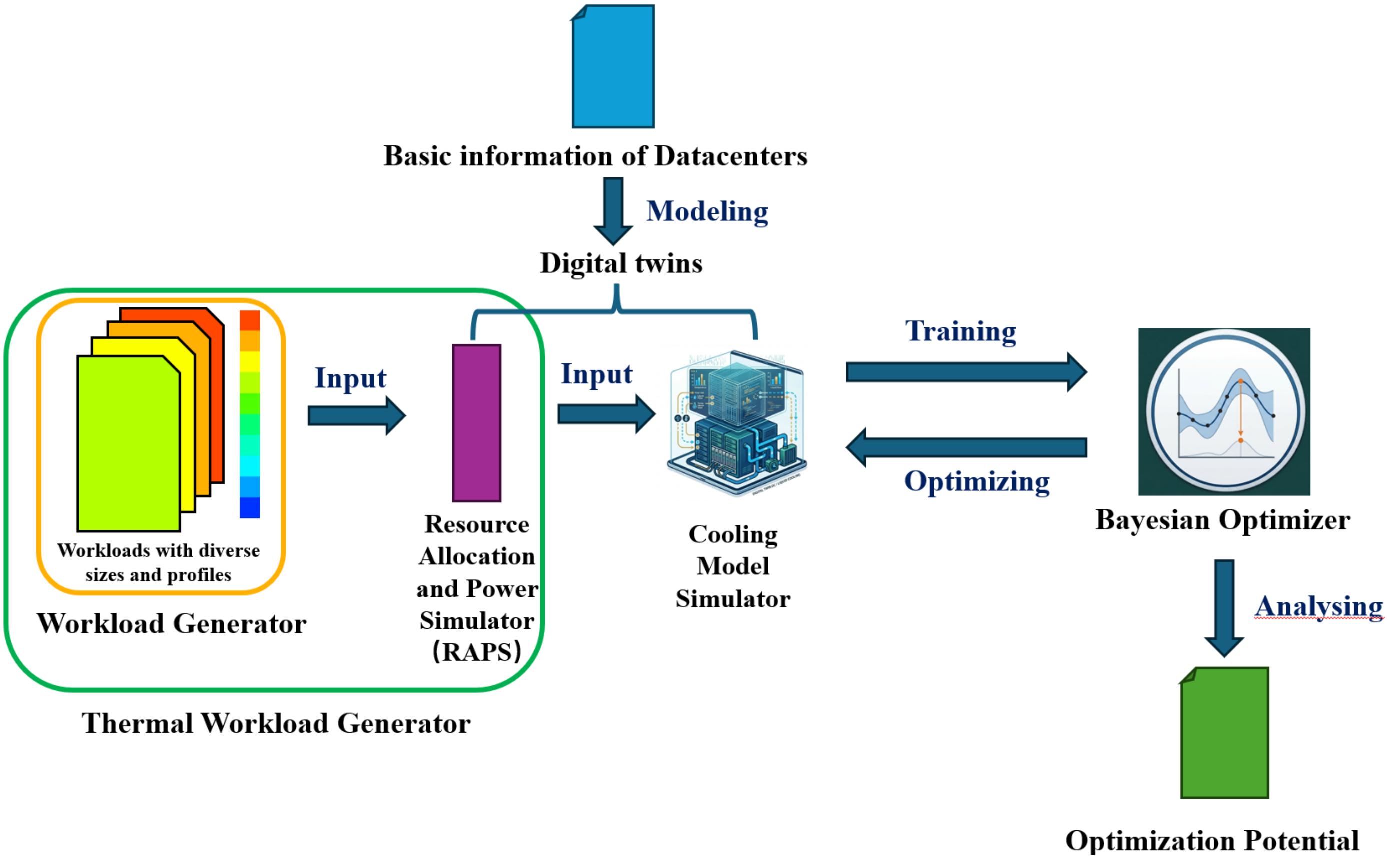


Figure 1. Methodology for evaluating the optimization potential of control variables in data center.

the random seed, requested horizon, system and scheduling configuration, power profile, distribution parameters, and row counts.

For seeded experiments, both the Python and NumPy random-number generators are initialized from the same recorded integer seed. This allows a workload realization to be reproduced and enables matched-seed comparisons across profiles and time horizons. The formal validation in Section 3.1 uses five seeds, three utilization/power profiles, and 24-h and 168 h horizons to quantify profile controllability, run-to-run variation, and temporal robustness. Algorithm 1 summarizes the implemented generation and finite-horizon recording procedure.

2.2 Cooling Model Simulator

Given the risk that extreme-load testing may cause irreversible damage to a physical data center, utilizing comprehensive digital twin frameworks to evaluate scheduling policies and cooling dynamics has become an essential methodology [22]. This study adopts a digital twin model to perform simulation-based evaluation. The digital twin model is derived from the Frontier supercomputing center model developed by the ExaDigiT project team for Oak Ridge National Laboratory [18]. This digital twin consists of two main components: RAPS and a cooling model simulator. Since RAPS was previously introduced in the section on the thermal workload generator, the following discussion focuses primarily on the cooling model simulator, which serves as the mathematical and physical model of the cooling system.

The mathematical and physical modeling of the cooling system in this study fully follows the open-source Frontier supercomputer model developed by the ExaDigiT project team based on real operating parameters [18]. The model was implemented in Modelica within the Dymola environment [23, 24] and consists of two main parts: material-flow modeling and control-flow modeling. Material-flow modeling is implemented by decomposing the data center into system blocks at different hierarchical levels, detailed information and methods can be found in [25]. Figure 2 shows the material-flow relationships among these system blocks, and the system block used to model the HotWaterLoop is illustrated in Figure 3. Control-flow modeling is attached to the material-flow model of each system block in the form of a subpackage named *Controls*, and different system blocks are connected through a control bus, detailed description can be found in [26]. The control-flow modeling of the CoolingTowerLoop, for example, is shown in Figure 4.

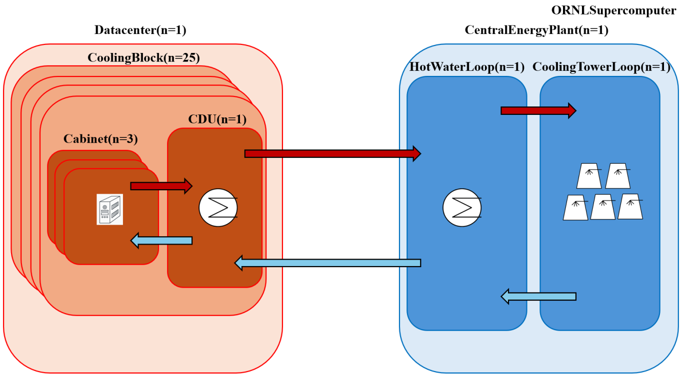


Figure 2. Schematic Diagram of Subpackage Relationships for Data Center System Blocks.

The cooling system model is built on three libraries. The first is the *Buildings* library [27], which mainly provides packaged components for large-scale equipment and control strategies, such as cooling towers and pump groups. The second is the *Transform* library [28], which mainly contains thermo-fluid components, including fluid parameters and properties, pipelines, valves, and heat exchangers, and also provides methods for unit conversion. The third is the *AutoCSM* library, which mainly provides interface and integration components.

By using the packaged components provided by these libraries and connecting them graphically to transmit parameters, the coupled nonlinear differential equations of the system can be constructed. Some component parameters are directly assigned in the corresponding variable-definition files or commands, while the initial values and initial states are defined according to the actual operating parameters of the data center.

With the aid of the methods provided in the *AutoCSM* library, together with the Functional Mock-up Interface (FMI) standard [29] between Modelica and Python and the modeling architecture of the system blocks specified in the `.json` file, the *Coolingmodels* model can be packaged as a FMU, in which the corresponding interfaces to the RAPS module are automatically constructed. Once the path to this FMU is obtained by RAPS, it can be incorporated into the program as an optional component for digital twin simulation of the data center.

In this study, the coupled differential equations in the FMU are solved using the *sdirk34hw* numerical algorithm, namely a third-/fourth-order singly diagonally implicit Runge–Kutta method. The choice of this solver is mainly motivated by the strong stiffness characteristics of the data center cooling system model. Conventional explicit algorithms are prone to divergence under such conditions, whereas *sdirk34hw* can not only rapidly damp high-frequency nonphysical oscillations, but also effectively control the matrix inversion cost introduced by multiphysics coupling through its singly diagonal structure, making it a suitable choice that balances numerical stability and computational efficiency.

In the source ExaDigiT validation against Frontier telemetry, the model-predicted PUE was within 1.4% of the telemetry-based PUE over approximately 24-h, while the RAPS power errors for idle, HPL-core, and peak-power tests were 2.1%, 4.7%, and 3.1%, respectively [18].

2.3 Bayesian Optimizer

To enable the selected setpoints to be adjusted in a targeted manner according to changing operating conditions, this study introduces a context-aware Bayesian optimization method that uses the operating condition of each time segment as contextual input to search for a more suitable setpoint for the corresponding segment.

In this study, Gaussian Process Regression (GPR) [30] is adopted as the probabilistic surrogate model in Bayesian optimization to characterize the posterior distribution of the system performance metric under given contextual states and setpoint conditions [31]. For the i-th time segment, let the contextual state vector be denoted by c_i, the value of the optimization variable by a_i, and the corresponding performance metric by y_i. Then, the training dataset can be expressed as

$$\mathcal{D} = \{(c_i, a_i, y_i)\}_{i=1}^{N} \tag{1}$$

where c_i denotes the contextual state vector of the i-th time segment, a_i denotes the value of the optimization variable corresponding to that segment, and y_i denotes the performance metric obtained from simulation under the corresponding contextual state and setpoint condition.

It should be noted that the present study adopts an offline-trained, segment-wise recommendation scheme for contextual Bayesian optimization, rather than an online rolling-update mechanism across time segments. In other words, the recommended setpoint for each segment is generated independently

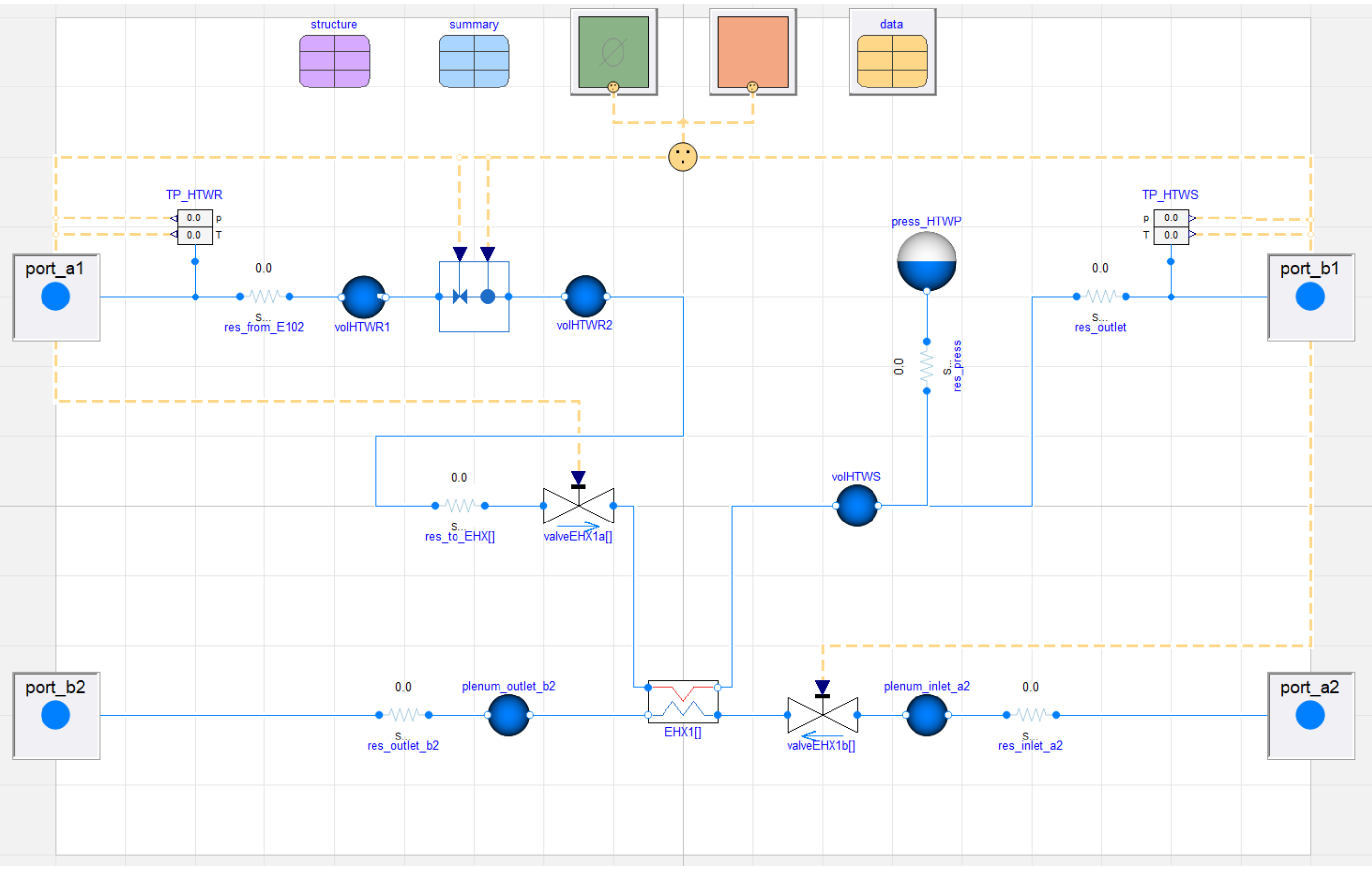


Figure 3. System block used to model the HotWaterLoop.

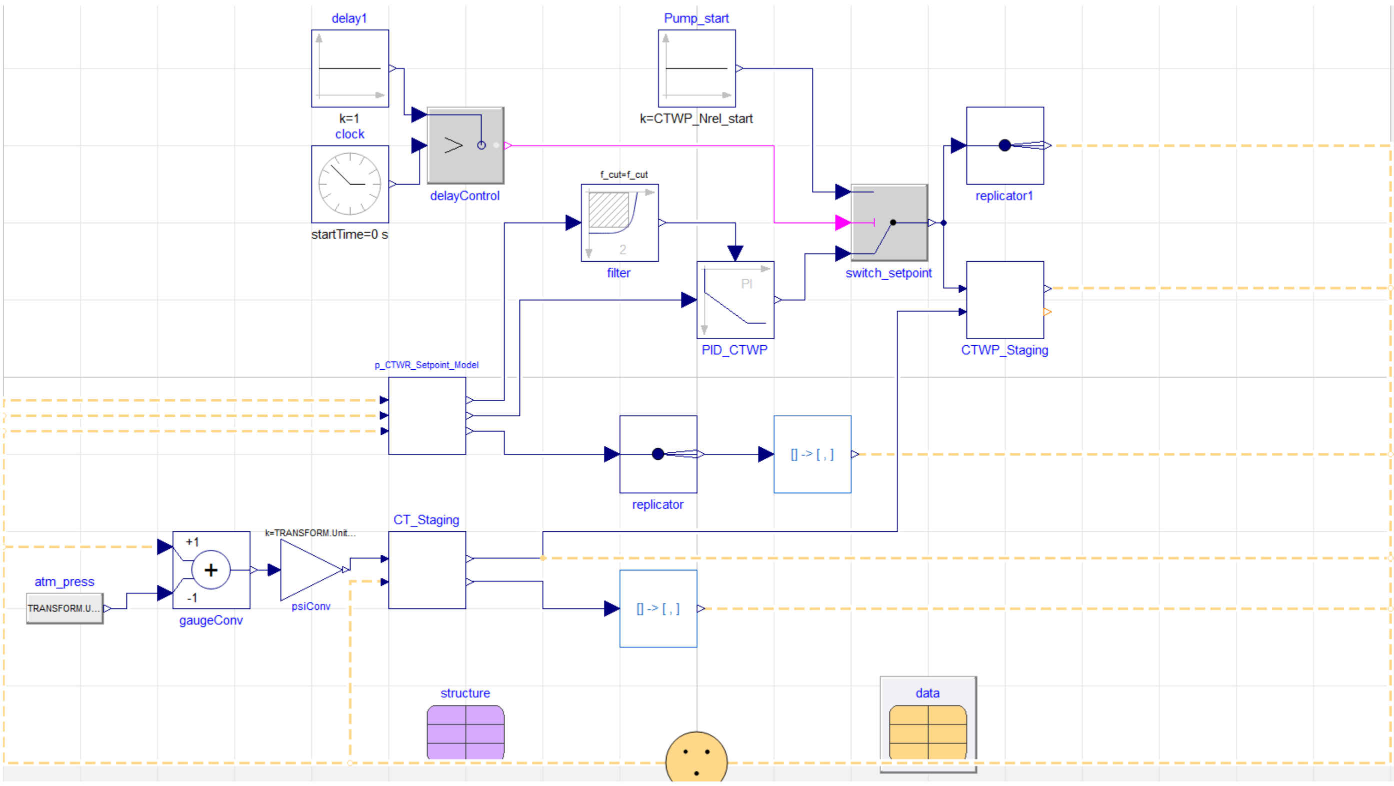


Figure 4. Control flow modeling for the CoolingTowerLoop.

based on its own contextual state using the same pretrained global surrogate model. Although fixing the setpoint within each segment helps preserve the stability of the underlying control system, it may limit responsiveness to transient workload variations occurring within the segment. Future work will investigate safety-constrained dynamic adjustment strategies, including receding-horizon updates, adaptive segmentation, and change-point detection. Rate-of-change limits, minimum holding times, smoothing constraints, and safe fallback mechanisms will also be incorporated to support continuous online operation in real data centers.

In the experiments reported in this study, the surrogate model was trained using the initial training workload and then kept fixed during evaluation on unseen workloads. No test-workload observations were incorporated into the training set. Model updating is retained as a possible extension of the framework but is not used in the zero-shot experiments reported here. The overall procedure is illustrated in Figure 5.

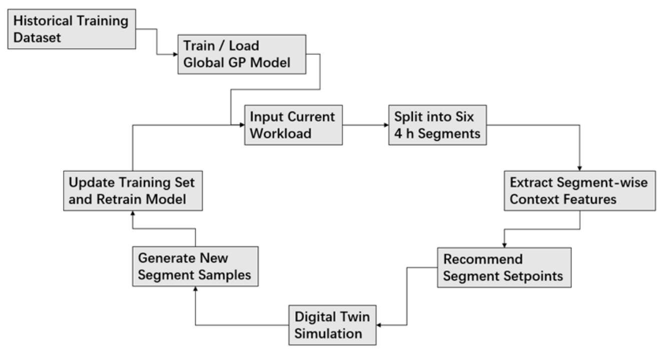


Figure 5. Flowchart of Segment-wise Setpoint Recommendation and Model Updating Based on Offline Contextual Bayesian Optimization.

For the sampling decision, this study adopts the Lower Confidence Bound (LCB) criterion based on the posterior mean and standard deviation [32] to select candidate parameters. For a given context, candidate points are first uniformly generated within the predefined search range of the optimization variable. The Gaussian process surrogate model is then used to predict the average PUE and the associated uncertainty for each candidate point, and the score is calculated according to

$$\text{score}\,(a) = \mu\,(a \mid c) - \beta\sigma\,(a \mid c) \tag{2}$$

where c denotes the context of the current time segment, and $\mu\,(a \mid c)$ and $\sigma\,(a \mid c)$ represent the predicted mean and standard deviation of the surrogate model, respectively, while β is the trade-off coefficient. The candidate optimization variable with the minimum score is selected as the recommended setpoint for the current time segment.

The coefficient β controls the exploration–exploitation trade-off of the LCB criterion. In this study, $\beta = 1.0$ was fixed *a priori* for all variables and workload conditions and was not tuned using the test workloads. This order-one setting introduces an uncertainty incentive corresponding to one posterior standard deviation, while retaining substantial emphasis on the predicted objective value. Such a moderate trade-off was considered appropriate for the present budget-constrained setting, in which each FMU evaluation is computationally expensive and only a small number of evaluations are available. A substantially larger β would emphasize exploration of uncertain regions, whereas a smaller value would make the search increasingly exploitative. Therefore, $\beta = 1.0$ is used here as a consistent design choice rather than being claimed as a universally optimal value.

This strategy balances regions with low predicted values and regions with high uncertainty, thereby enabling effective search under limited-sample conditions [33].

In constructing the contextual state variables, this study considers both thermal load characteristics and cooling system operating characteristics. At the same time, given the high dimensionality of the original variables from 25 compute blocks, variables of the same type are first spatially aggregated, and representative statistical measures are then extracted, so that the state at a single time step can be compressed into a low-dimensional and physically interpretable feature representation. Furthermore, since the raw data are sampled at an interval of 15 s, whereas the control optimization targets segment-level setpoints, this study adopts a fixed-length time segmentation strategy to statistically summarize the single-time-step state features within each segment. A fixed-length contextual vector is thereby constructed and aligned one-to-one with the control variable and performance metric of the corresponding segment, serving as the input for subsequent Bayesian optimization. The detailed feature composition and their segment-level aggregation rules are provided in Appendix Table A1. At the current stage of this study, statistical features are mainly adopted, while temporal features such as first-order differences have not yet been explicitly introduced.

In the performance evaluation design, this study takes PUE as the core optimization objective and further introduces a temperature penalty term in order to jointly consider energy-efficiency improvement and basic thermal safety constraints. For the i-th time segment, the performance value is defined as

$$y_i = \mathrm{PUE}_{i,\mathrm{mean}} + \lambda_{\mathrm{sec}} \max\left(0, T^{\max}_{\mathrm{sec},i} - T_{\mathrm{sec,lim}}\right) + \lambda_{\mathrm{prim}} \max\left(0, T^{\max}_{\mathrm{prim},i} - T_{\mathrm{prim,lim}}\right). \quad (3)$$

where $\mathrm{PUE}_{i,\mathrm{mean}}$ denotes the average PUE of the i-th time segment, while $T^{\max}_{\mathrm{sec},i}$ and $T^{\max}_{\mathrm{prim},i}$ denote the maximum secondary-side and primary-side return-water temperatures within that segment, respectively. In the present experiments, $T_{\mathrm{sec,lim}} = 30°\mathrm{C}$ and $T_{\mathrm{prim,lim}} = 35°\mathrm{C}$ are adopted as conservative reference thresholds for activating the soft penalties, with the corresponding penalty weights fixed at $\lambda_{\mathrm{sec}} = 0.002$ and $\lambda_{\mathrm{prim}} = 0.001$. The penalty weights were determined using a magnitude-matching rule: the penalty increment associated with a moderate temperature exceedance was kept on the same order of magnitude as the PUE variations relevant to the optimization. Specifically, each $1°\mathrm{C}$ exceedance contributes 0.002 and 0.001 to the objective for the secondary and primary sides, respectively. This scaling makes temperature exceedance sufficiently influential to alter the ranking of thermally aggressive candidates, while preventing the penalty term from dominating the PUE-based objective. They were kept fixed throughout all experiments and were not tuned using the test workloads. Therefore, the temperature thresholds should be interpreted as methodological reference values rather than calibrated physical safety limits.

When either reference threshold is exceeded, the corresponding penalty term increases the performance value, thereby discouraging the optimizer from pursuing PUE reduction without considering thermal conditions. In this study, y_i also serves as the prediction target of the surrogate model, and setpoint search in the Bayesian optimization process is carried out by minimizing y_i. Future work may further incorporate pressure-stability constraints, calibrated thermal safety limits, or constrained and multi-objective formulations. At the present stage, however, the objective function remains primarily centered on PUE in order to emphasize the evaluation of energy-efficiency optimization potential.

In summary, the context-aware Bayesian optimization framework constructed in this study essentially uses segment-level operating condition features as conditional inputs to perform segmented static optimization of the decision variables. This method preserves the structural stability of the original control system while enabling the setpoints to be adaptively adjusted according to changing operating conditions, thereby providing a practical upper-level optimization framework for evaluating the optimization potential of different control variables.

At the current stage, this study optimizes upper-level control variables individually to identify key variables with optimization potential and to evaluate their independent contributions under computationally expensive FMU simulations. In higher-dimensional control parameter spaces, increasing the number of jointly optimized variables substantially raises the number of simulation samples required to characterize inter-variable coupling. Meanwhile, the computational costs associated with Gaussian process training and acquisition function optimization also increase. In particular, directly extending the current one-dimensional discrete candidate search to a multidimensional Cartesian grid would cause the number of candidate points to grow exponentially with dimensionality. Future work may improve the scalability of multi-parameter joint optimization by selecting key control variables through sensitivity analysis and combining control-loop decomposition with batch evaluation strategies.

3 Experimental Design and Results

This study conducts two series of experiments to verify the effectiveness of the proposed Bayesian-optimization-based method for exploring the energy optimization potential of data center cooling systems. These experiments include the validation of the proposed probability-based workload generator and the comparison of optimization results obtained by applying the corresponding Bayesian optimization algorithm to data center control parameters.

In addition, this study further designs a set of exploratory experiments to investigate the optimization potential of multiple control variables using the proposed Bayesian-optimization-based framework.

3.1 Validation of Synthetic Workload Controllability and Robustness

To evaluate the workload generator beyond a single realization, we designed a test plan comprising 20 separately executed simulations. The profile analysis comprised five 24-h runs for each of the low, uniform,

Table 1. Run-level validation of synthetic utilization/power profiles. Values are means across five 24-h simulations per profile; the repeated-measures ANOVA (RM-ANOVA) and Friedman tests treat seed as a block.

Run-level metric	Low	Uniform	High	RM-ANOVA p	Partial η_p^2	Friedman p	Kendall's W
Mean CPU utilization (%)	17.4	51.7	82.9	9.77×10^{-12}	0.998	0.00674	1.000
Mean GPU utilization (%)	17.5	49.2	82.8	1.95×10^{-10}	0.996	0.00674	1.000
Completed-job avg. node power (kW/node)	1.024	1.715	2.418	5.02×10^{-10}	0.995	0.00674	1.000
IT power proxy (kW/available node)	0.949	1.597	2.252	5.20×10^{-9}	0.992	0.00674	1.000
Mean node occupancy (%)	93.6	94.5	93.1	0.512	0.154	0.819	0.040

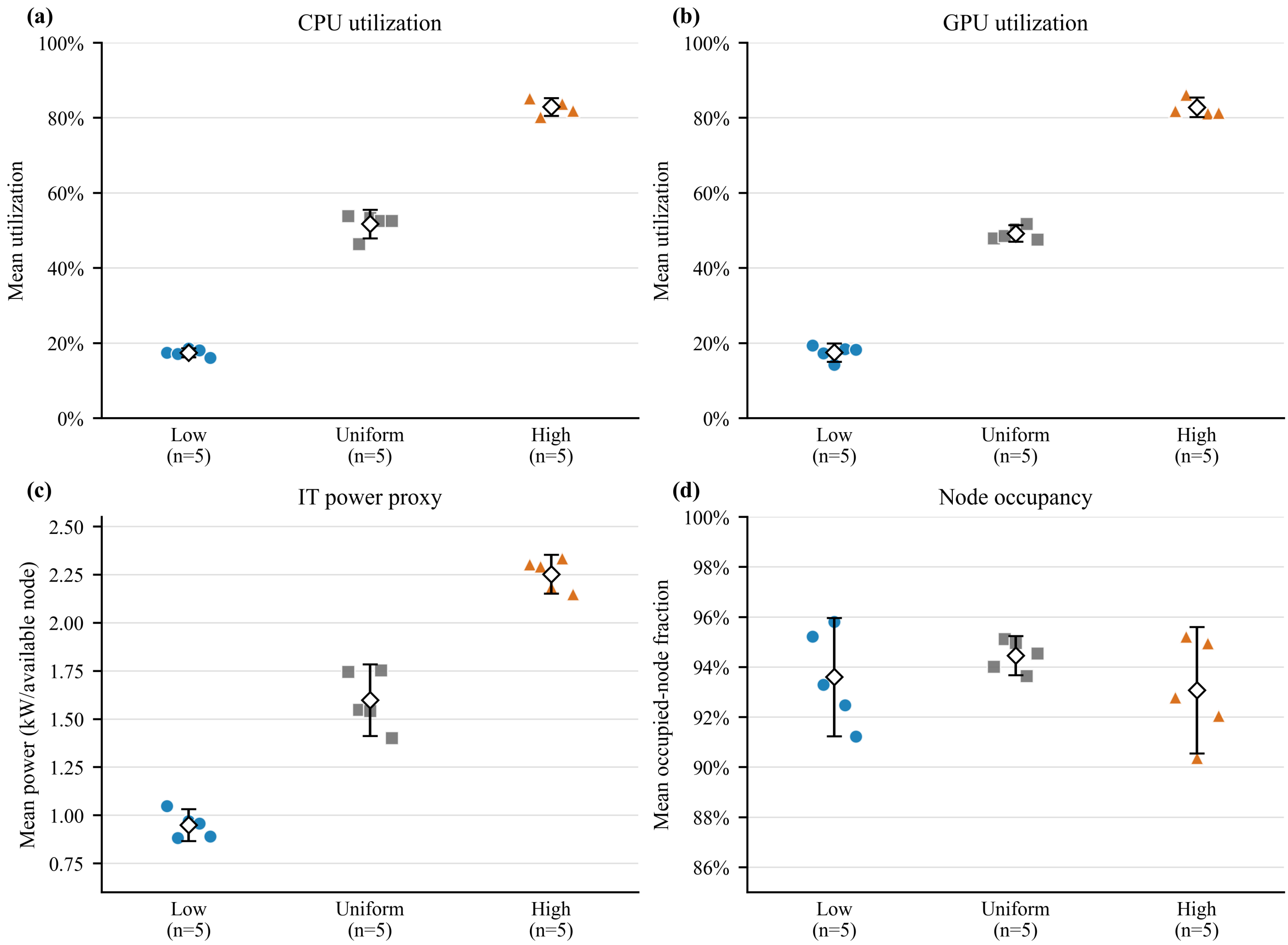


Figure 6. Run-level comparison of the low, uniform, and high utilization/power profiles across five 24-h simulations per profile (seeds 42–46). Symbols show individual runs; diamonds and error bars show means and 95% CIs. The IT power proxy is reconstructed rather than measured facility power.

and high utilization/power profiles, using seeds 42–46. Temporal robustness was evaluated with five seed-matched pairs of 24-h and 168 h uniform-profile runs; the five 24-h uniform runs were shared with the profile analysis. Each complete simulation run, rather than each individual job, was treated as the statistical unit, and run-level means and 95% confidence intervals (CIs) were calculated.

Because the same seed set was reused across profiles, the primary profile analysis treated seed as a block. Omnibus effects were assessed using one-way repeated-measures analysis of variance (ANOVA), with the Friedman test as a non-parametric sensitivity analysis. Pairwise contrasts used paired t-tests and exact Wilcoxon signed-rank tests, with Holm correction across the three profile contrasts within each metric and test family. Unblocked one-way ANOVA and Kruskal–Wallis tests, followed by pairwise Welch tests, were also calculated as sensitivity checks and yielded the same qualitative profile-separation conclusion. Temporal comparisons used paired t-tests and Wilcoxon signed-rank tests, with Holm correction across 13 prespecified run-level metrics.

As shown in Table 1 and Figure 6, mean CPU utilization increased from 17.4% under the low profile to 51.7% under the uniform profile and 82.9% under the high profile. Mean GPU utilization followed the same ordering (17.5%, 49.2%, and 82.8%), while the IT power proxy increased from 0.949 to 1.597 and 2.252 kW per available node. The four utilization/power metrics exhibited large profile effects in the repeated-measures ANOVA ($p = 9.77 \times 10^{-12}$ to 5.20×10^{-9}; partial $\eta_p^2 = 0.992$–0.998). For each of these metrics, the Friedman test gave $p = 0.00674$ and Kendall's $W = 1.0$, indicating the same low-to-high ordering for all five seeds. All 12 paired t-test contrasts for these four metrics remained significant after Holm correction (adjusted $p = 2.91 \times 10^{-7}$ to 1.93×10^{-4}). With only five blocks, however, the minimum exact Wilcoxon p-value was 0.0625 and the Holm-adjusted pairwise values were 0.1875; pairwise non-parametric significance is therefore not claimed.

In contrast, the three profiles produced similar mean node occupancy (93.6%–94.5%). No systematic profile effect was detected for the nine prespecified workload and operational control metrics, including job arrival rate, requested nodes, intended wall time, offered node-time, occupancy, occupancy change, and terminal completed/queued fractions (repeated-measures ANOVA, $p = 0.221$–0.969; Friedman test, $p = 0.165$–0.819). The profile parameter therefore controlled computational and power intensity without a detected systematic shift in these sampled demand and occupancy metrics.

The analyzed first-day metrics from every 168 h run reproduced those from the paired standalone 24-h run within a numerical tolerance of 10^{-9}. Over the full horizon, mean node occupancy increased from 94.45% to 98.16% (Holm-adjusted paired t-test, $p = 0.002$), and daily means stabilized near 98%–99% after the first day (Table 2 and Figure 7). The exact Wilcoxon result did not remain significant after correction ($p_{\text{Holm}} = 0.813$), reflecting the limited resolution of five pairs. In contrast, no horizon effect was detected for the IT power proxy (1.597 versus 1.653 kW per available node; $p_{\text{Holm}} = 1.000$). The P95 absolute 15-min occupancy change decreased from 4.98 to 2.61 percentage points, but the difference did not remain significant after correction ($p_{\text{Holm}} = 0.075$). These observations support deterministic prefix reproduction and sustained long-horizon stress generation, while the occupancy increase shows that this configuration should not be interpreted as a stationary occupancy process.

Because private Frontier job logs were unavailable, the public Adastra MI250 15-day workload trace released by CINES [21] was used only as a cross-system external structural reference rather than as Frontier ground truth. For this analysis, we used a CSV export of the published Parquet table. A derived analysis table retained job identifiers, states, allocated nodes, and runtimes; expressed start and end times as seconds relative to the earliest valid start; and calculated job-average node power from the reported node-energy arrays and runtime. The reference trace was not used to fit the synthetic arrival, node-count, wall-time, or utilization-profile parameters. The Adastra and Frontier-scale configurations have 356 and 9472 available nodes, respectively; absolute node counts and power levels were therefore not interpreted as cross-system fidelity measures. The primary public-trace samples comprised 4902 completed jobs with positive runtime for the duration analysis and 4770 completed jobs with positive runtime and node power for the power analysis. These data were compared with five uniform-profile 24-h Frontier-scale simulations. Job duration was weighted by node-time, job-average node power was weighted by an energy proxy, and both were normalized by their dataset- or run-specific weighted medians. Absolute changes in reconstructed node occupancy were evaluated at 15-min intervals. Weighted empirical distribution separation was summarized by the KS distance without a p-value because the standard two-sample KS significance test does not apply to weighted empirical distributions.

The mean weighted KS distances were 0.30 ± 0.05 for node-time-weighted duration, 0.30 ± 0.10 for energy-weighted node power, and 0.29 ± 0.05 for absolute 15-min occupancy change (Figure 8). The empirical distributions overlap in their principal contribution ranges, while KS distances near 0.30 also indicate moderate cross-system separation. This comparison supports external plausibility for selected operational contribution shapes, but it does not

Table 2. Seed-matched comparison of 24-h and 168 h uniform-profile simulations. Paired t-test and Wilcoxon p-values are Holm adjusted across 13 prespecified metrics; pp denotes percentage points.

Run-level metric	24-h mean	168 h mean	Paired difference (168 h–24-h)	Paired t p_{Holm}	Wilcoxon p_{Holm}
Mean node occupancy (%)	94.45	98.16	+3.71 pp	0.002	0.813
IT power proxy (kW/available node)	1.597	1.653	+0.056	1.000	1.000
P95 absolute 15-min occupancy change (pp)	4.98	2.61	-2.37 pp	0.075	0.813

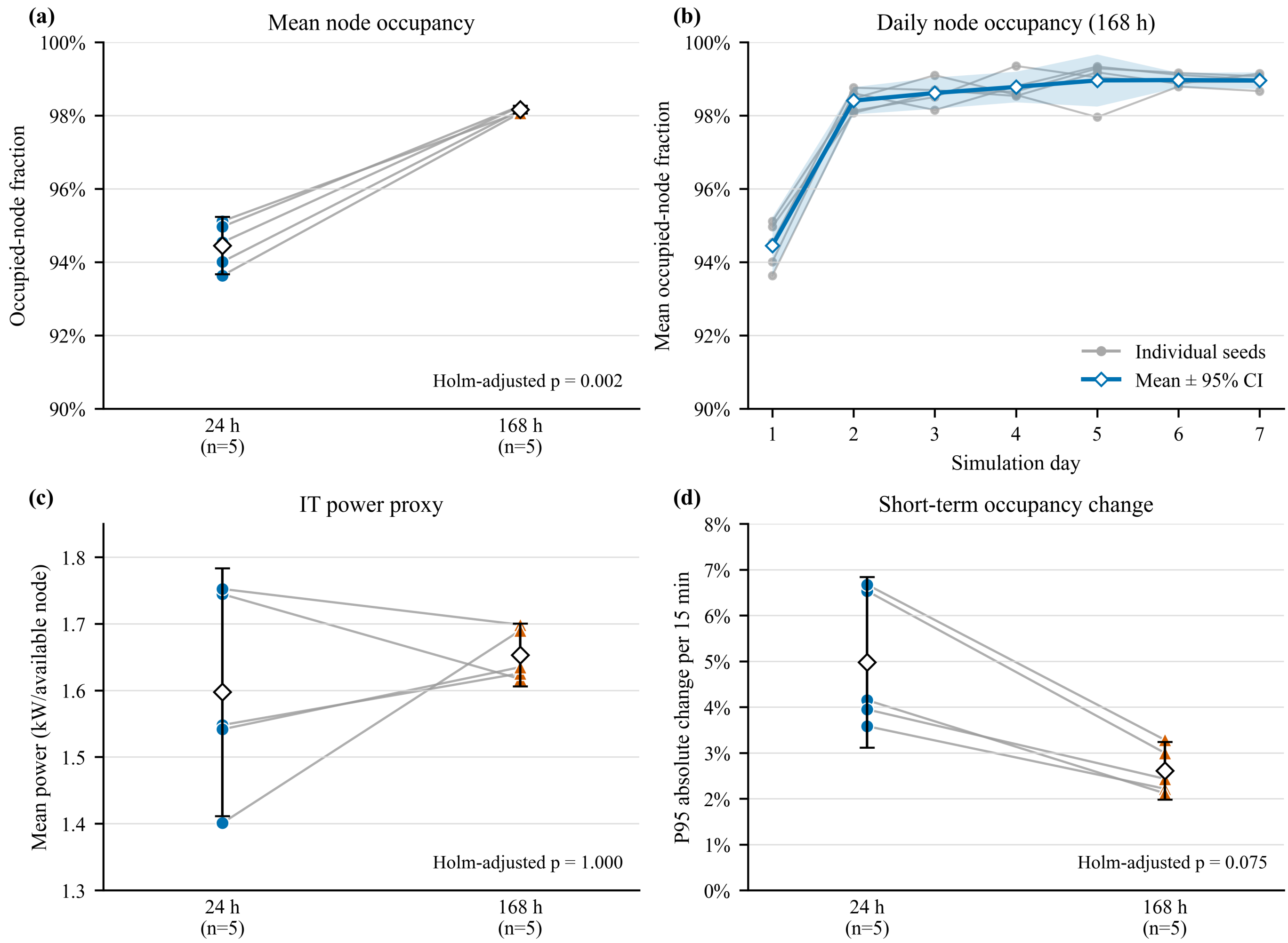


Figure 7. Seed-matched comparison of five uniform-profile simulations at 24 and 168 h (seeds 42–46). Gray lines show paired seeds or daily trajectories; diamonds and error bars show means and 95% CIs. Annotations give Holm-adjusted paired t-test p-values.

establish statistical identity or reconstruction of private Frontier operations.

Three limitations qualify the validation. First, the public Adastra trace originates from a different system and is an external reference only. Second, five seeds quantify run-to-run variation but limit the resolution of exact non-parametric pairwise tests. Third, the offered node-time demand drives the 168 h simulations into a sustained high-occupancy regime after the initial day. This behavior is useful for cooling stress testing, but it is not evidence that all workload statistics are temporally stationary.

3.2 Experimental Design for Bayesian Optimization

3.2.1 Selection and Analysis of Optimization Variables

An analysis of the established digital twin model of the data center cooling system shows that the tunable parameters of the system are mainly distributed across three levels. The first category consists of parameters in the Coolant Distribution Unit (CDU) loop, which are primarily used to maintain the stability of the

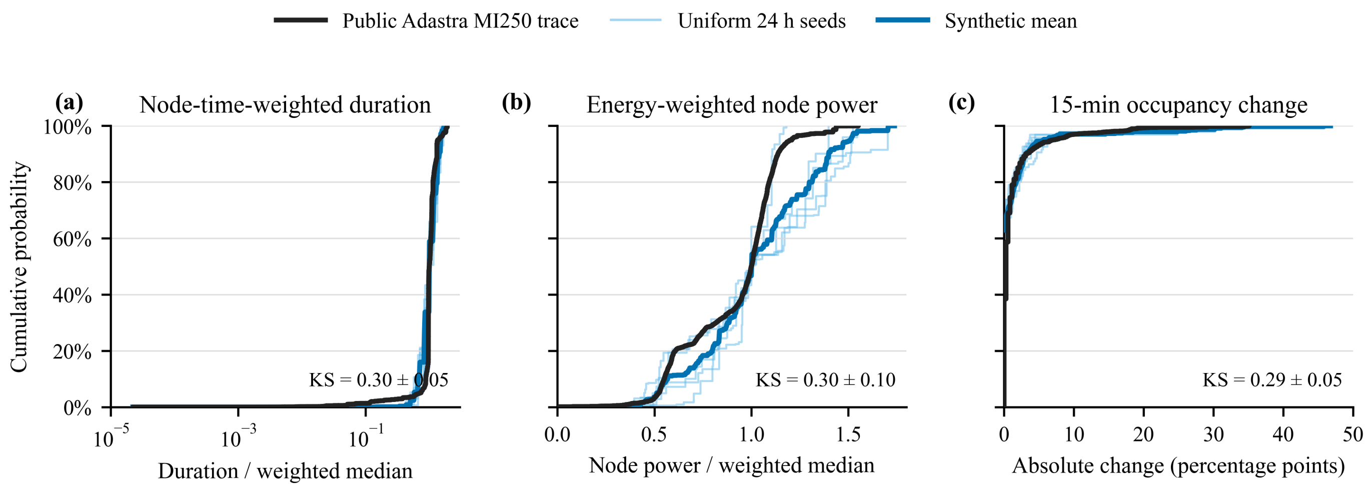


Figure 8. Operationally weighted comparison of the public Adastra MI250 15-day trace and five 24-h synthetic uniform-profile runs. Panels show node-time-weighted duration, energy-weighted node power, and 15-min occupancy change; annotations give the mean ± standard deviation of the weighted KS distance.

secondary-side supply temperature and loop pressure differential, including the CDU pump speed and the primary-side valve control signal. The second category consists of parameters in the Hot Water (HTW) loop, which mainly concern the pressure differential setpoint and pump staging of the hot-water loop and are used to regulate the primary-side flow rate. The third category consists of parameters in the Cooling Tower Water (CTW) loop, including the cooling tower water pump speed, pump staging, cooling tower staging, and setpoints related to the tower-side target temperature, which are used to maintain the CTW return header pressure and ensure the stability of the HTW supply temperature.

This study gives priority to upper-level setpoint parameters that do not directly alter the structure of the underlying actuators or controllers, but instead indirectly influence the system operating state by adjusting control targets. Such variables usually possess better physical interpretability and engineering implementability, making them more suitable as research objects for Bayesian optimization. It should be emphasized that the selected variables serve as representative examples for exploring and comparing optimization potential, rather than constituting an exhaustive set of all tunable parameters in the cooling system. Actuator-level parameters, such as pump-speed commands, valve positions, and equipment on/off signals, may also significantly affect system performance. However, their optimization involves faster control time scales and stronger coupling with embedded feedback and staging logic, and is therefore beyond the scope of the present study. Future work will investigate hierarchical or joint optimization frameworks that incorporate these actuator-level parameters.

For the selection of the known optimization variable, this study chooses adj in the CTW loop as the validation target for the effectiveness of the proposed method. On the one hand, the temperature setpoint layer affected by adj has a clear correspondence with the cooling-tower-side temperature setpoint variables commonly adopted in existing liquid-cooling optimization studies such as LC-Opt [16]. On the other hand, adj does not directly modify the underlying PI control or equipment staging logic, but instead affects the overall system operating state by adjusting an upper-level setpoint. It therefore has both representativeness and interpretability. The specific location of its control action and the related formula derivations are provided in Appendix C.

To verify whether adj is suitable as an optimization variable, a single-factor sensitivity analysis is conducted under the same 24-h randomly generated workload. With all other control parameters kept unchanged, the PUE time series corresponding to different global values of adj are compared. The results, shown in Figure 9, indicate that different values of adj lead to clear changes in the overall PUE level, with a relatively stable ordering across settings. This suggests that adj has a significant impact on system energy efficiency and is therefore suitable as the core setpoint for subsequent Bayesian optimization.

It should be clarified that the term ‘known

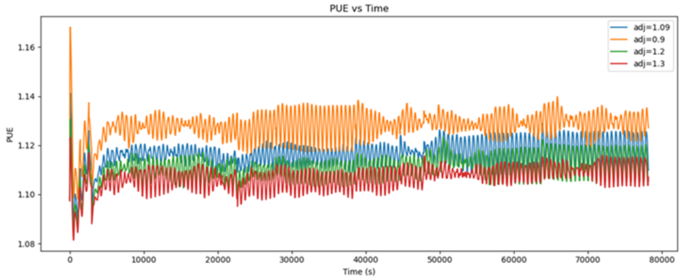


Figure 9. Time-series comparison of PUE across different global adj values under an identical 24-h random workload.

optimization variable' only indicates that the influence of adj on PUE was independently confirmed through the preceding single-factor sensitivity analysis. This information was not encoded in the Bayesian optimization process as prior knowledge, a preferred search direction, or a predefined optimum; adj was optimized using the same Gaussian-process surrogate model and LCB acquisition procedure as the other candidate variables.

On this basis, to further investigate the ability of the proposed method to identify the optimization potential of different types of control parameters, three additional potential optimization variables are selected for testing. The first is ps_stage_down, the pressure threshold for cooling tower stage-down, which mainly affects the timing at which cooling towers are taken offline; the training samples indicate that this variable exhibits a certain nonlinear relationship with the segment-wise mean PUE. The second is CDU_dp_nom, the target pressure difference setpoint for CDU pump differential-pressure control, which mainly affects the primary-side circulation capacity by changing the pump-speed control target; within the current sample range, this variable shows a strong monotonic relationship with the mean PUE. The third is $CTWP\_Nrel\_stageUp$, the stage-up threshold for the cooling tower water pumps, which determines when the pump group is staged up; the training samples show that this variable exhibits a certain non-monotonic relationship with the mean PUE. These results indicate that the optimization landscapes of these variables differ in shape, making them suitable for further testing the ability of the proposed method to identify different variable characteristics. The corresponding sample relationship plots are provided in Appendix D.

In summary, this study uses adj as a known effective variable to validate the feasibility of the proposed method, and further adopts ps_stage_down, CDU_dp_nom, and $CTWP\_Nrel\_stageUp$ as potential variables to examine the ability of the method to identify variables with different optimization-space characteristics.

3.2.2 *Experimental Design*

To validate the effectiveness of the Bayesian optimization method, this study adopts a unified experimental protocol for both the known variable and the potential variables. For each control variable, training samples are first constructed under one randomly generated 24-h workload. A Gaussian-process surrogate model is then trained using the correspondence among the segment-level contextual states, the corresponding control-variable setting, and the resulting segment-level penalized objective value defined in Eq. (3). The trained model is subsequently evaluated on six unseen workload scenarios with different durations and load intensities, namely `24-h_highjob`, `24-h_lowjob`, `36h_highjob`, `36h_lowjob`, `48h_highjob`, and `48h_lowjob`, as illustrated in Appendix E.

For these six test scenarios, each surrogate model was kept fixed after training on the initial 24-h workload, and no test data were used for retraining or model updating. This setting was used to evaluate zero-shot generalization across workload duration and load-intensity conditions. In addition, to examine robustness to stochastic workload realizations within the same load category, the fixed `adj_seed42` model was evaluated on 15 independently generated unseen 24-h workloads, including five random realizations each under high-, medium-, and low-load conditions.

To evaluate the sensitivity of the optimization results to model-fitting randomness, the Gaussian-process fitting and recommendation procedure for each of the four control variables was independently repeated using random seeds 7, 17, and 42. The initial training samples, six test workload traces, search bounds, acquisition-function settings, and FMU simulation configurations were kept unchanged across the three seeds. The trained models were also kept frozen during testing. Consequently, this experiment isolates algorithmic repeatability associated with Gaussian-process fitting and recommendation, rather than variability introduced by newly generated workload realizations.

The three validation settings serve complementary purposes. The six unseen scenarios assess cross-condition generalization, the 15 independently generated workloads assess robustness to workload-realization randomness, and the three model-fitting seeds assess algorithmic repeatability. The results are evaluated using the mean PUE improvement, standard deviation, 95% confidence interval, positive-improvement rate, and consistency of the recommended settings. Together, these experiments assess the effectiveness and robustness of the proposed method, as well as its ability to distinguish the optimization potential and workload dependence of different control variables.

3.2.3 Experimental Results and Analysis

Budget-constrained efficiency comparison

Considering that the digital-twin-based co-simulation framework adopted in this study involves a high cost per evaluation, with each candidate parameter evaluation requiring one complete FMU simulation, sample efficiency is of great practical importance for the optimization method.

To this end, this study further designs a supplementary budget-constrained comparison experiment to compare the search efficiency of Bayesian optimization and simple search methods under expensive black-box evaluation conditions. In this experiment, the representative control variable *adj* is selected and treated as a single global static setpoint over a fixed 24-h workload for one-dimensional optimization. Unlike the context-aware optimization framework presented earlier in this study, this supplementary experiment does not introduce segment-wise state features. Its purpose is not to replace the contextual optimization method, but rather to provide a more intuitive demonstration of the performance differences among different search strategies under a limited simulation budget through a simplified static scalar optimization task. Specifically, under the same small evaluation budget, this study compares the best-so-far mean PUE convergence trajectories of the original setting, coarse-grained grid search, random search, and Bayesian optimization.

As shown in Figure 10, Bayesian optimization can approach the low-PUE region more rapidly with fewer simulation runs, demonstrating higher search efficiency. By contrast, although the coarse-grained grid search exhibits a relatively stable downward trend, it requires more evaluations to approach the optimal boundary, while random search may achieve some early improvement due to a few favorable samples but shows relatively limited convergence capability thereafter. These results indicate that, in the high-cost evaluation scenario driven by high-fidelity FMU simulation, Bayesian optimization offers better sample efficiency than simple search baselines, which further supports its use in subsequent segment-wise contextual setpoint optimization.

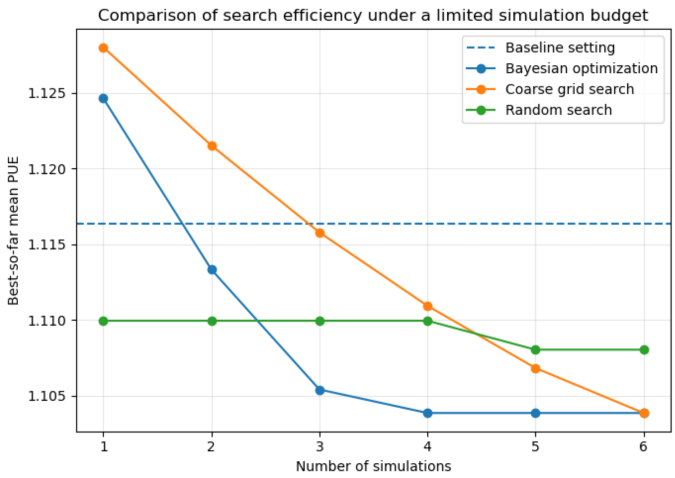


Figure 10. Comparison of search efficiency of different search methods for the static adj optimization task under a limited simulation budget.

Zero-shot robustness across repeated unseen workloads To further evaluate zero-shot generalization, the `adj_seed42` model trained on the initial 24-h workload was kept fixed, with no retraining or parameter updating during testing. The test set consisted of 15 independently generated 24-h unseen workloads, including five random realizations each under high-, medium-, and low-load conditions.

Figure 11 presents the paired comparisons of the baseline and optimized mean PUE for each workload. All paired lines decrease from the baseline to the optimized setting, indicating that the optimized mean PUE is lower for every tested workload.

Figure 12 shows the relative PUE improvement achieved by the fixed adj_seed42 model across 15 unseen workloads. Each load group contains five independent workload realizations. The colored points represent individual test results, while the black markers and error bars indicate the group means and 95% confidence intervals, respectively. The medium-load group achieves the largest average improvement, the high-load results are the most concentrated, and the medium- and low-load groups show relatively greater variability.

Detailed statistics are summarized in Table 3. The consistent PUE reductions across all unseen workloads demonstrate that the fixed model retains robust zero-shot transfer performance under different load

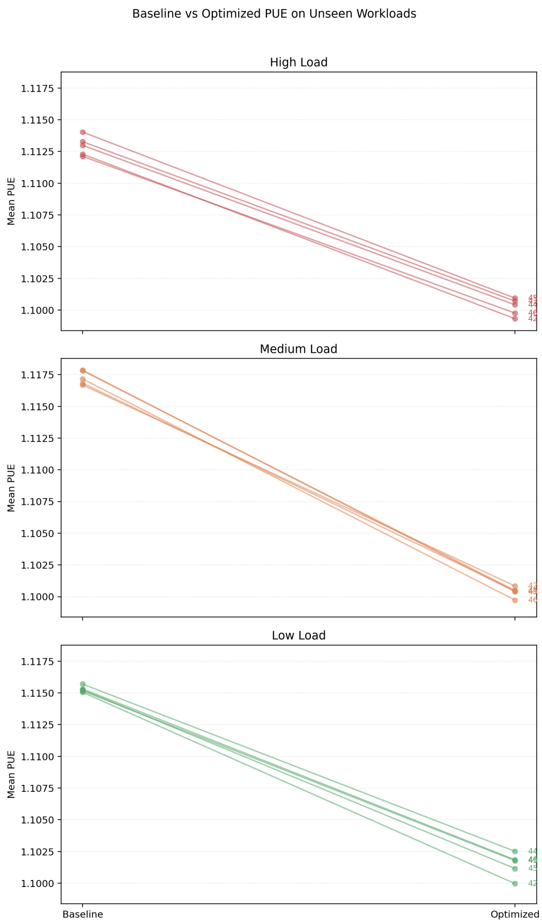


Figure 11. Paired comparison of baseline and zero-shot optimized mean PUE across 15 unseen workloads.

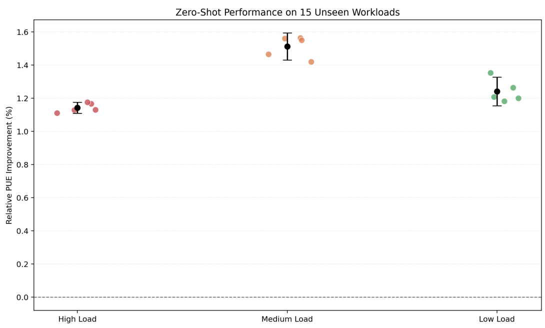


Figure 12. Relative PUE improvement of zero-shot optimization across 15 unseen workloads.

levels and workload realizations.

Results for the known variable adj For the known optimization variable adj, Figure 13 shows the PUE variation obtained using the representative model trained with random seed 42. Under the same 24-h randomly generated workload, the baseline control setting is compared with the setting obtained by Bayesian optimization. The results show that the average PUE of the system decreases from 1.1164 to 1.1104 after optimization, corresponding to an absolute

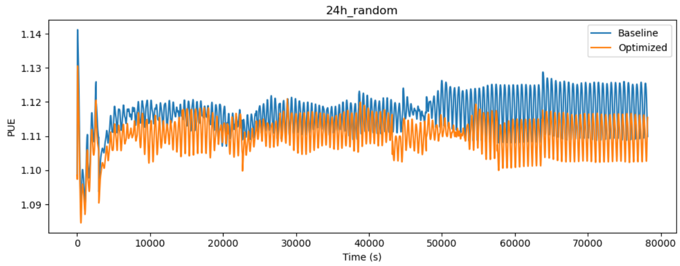


Figure 13. Time-series comparison of PUE under baseline and optimized control strategies for the same 24-h random workload using the representative seed-42 model.

reduction of 0.0060 and a relative improvement of approximately 0.54%. This magnitude is consistent with PUE improvements reported in comparable liquid-cooling optimization studies; for instance, LC-Opt reported cooling-energy reductions of a similar order under controlled workload conditions [16], and Bayesian optimization applied to HVAC setpoint control has similarly demonstrated measurable energy savings under comparable budget constraints [14]. From the time-series results, it can be observed that, except for the initial transition stage, the optimized PUE curve remains below the baseline curve for most of the time, indicating that the proposed method can effectively reduce the overall energy-efficiency metric during system operation while preserving the original control structure.

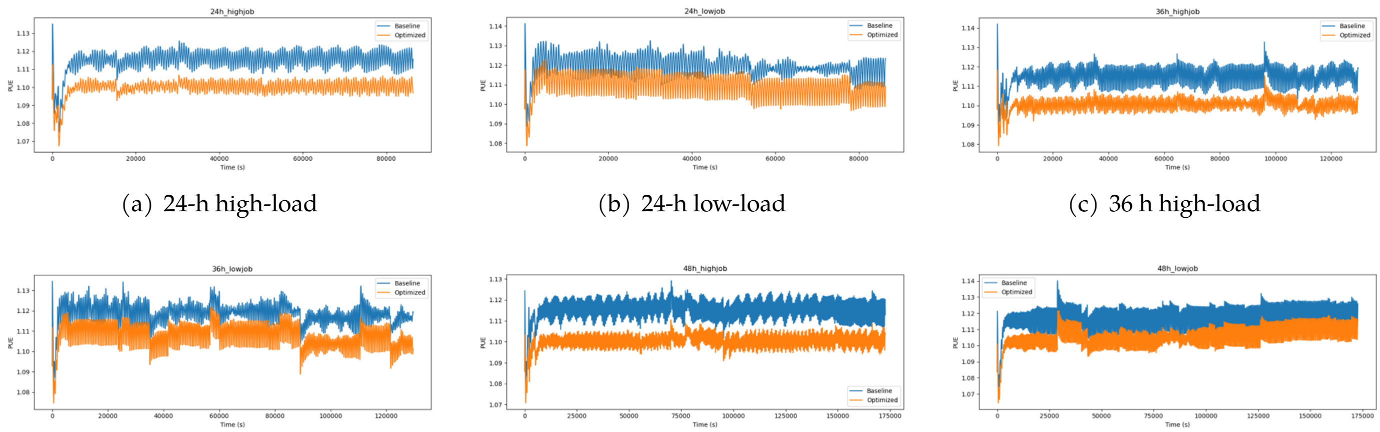
To further validate the applicability of the proposed method under new operating conditions, this study conducts additional tests on six different workloads that were not involved in training and compares the optimized control settings with the original control settings, as shown in Figure 14.

The time-series results show that the optimized PUE curve remains below the original control curve. This indicates that, for the selected known variable, the proposed method can provide setpoint recommendations with practical benefits under new workload conditions based on the contextual state.

As shown in Table 4, the optimized adj settings achieve consistent PUE reductions under all six unseen workload conditions with absolute improvements ranging from 0.0104 to 0.0152 and relative improvements ranging from 0.93% to 1.37%. This demonstrates that the proposed contextual Bayesian optimization framework is able to generalize beyond the training workload and provide practically beneficial setpoint recommendations for the known variable adj under new operating conditions.

Table 3. Statistical summary of zero-shot optimization performance across 15 unseen workloads.

Group	n	Baseline Mean PUE	Optimized Mean PUE	Mean PUE reduction (%)	SD (%)	95% CI (%)	Positive Improvement
High Load	5	1.1129	1.1002	1.142	0.027	[1.108, 1.176]	5/5 (100.0%)
Medium Load	5	1.1173	1.1004	1.512	0.066	[1.430, 1.593]	5/5 (100.0%)
Low Load	5	1.1153	1.1015	1.241	0.069	[1.155, 1.327]	5/5 (100.0%)



(a) 24-h high-load (b) 24-h low-load (c) 36 h high-load

(d) 36 h low-load (e) 48 h high-load (f) 48 h low-load

Figure 14. Comparison of PUE under baseline and BO-optimized control strategies for *adj* under six workload conditions using the representative seed-42 model.

These results correspond to the representative seed-42 model; the consistency of the recommendations and realized improvements across additional random seeds is examined separately in the robustness and repeatability analysis.

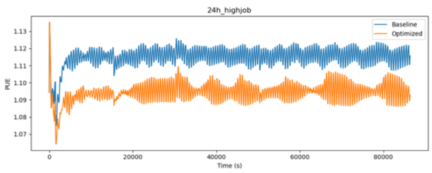
Table 4. Summary of PUE improvement after *adj* optimization under six unseen workload conditions using the representative seed-42 model.

Case	Baseline Mean	Optimized Mean	Absolute PUE reduction	Relative PUE reduction (%)
24-h_highjob	1.1151	1.0998	0.0152	1.3673
24-h_lowjob	1.1196	1.1092	0.0104	0.9262
36h_highjob	1.1150	1.1005	0.0146	1.3058
36h_lowjob	1.1186	1.1076	0.0110	0.9856
48h_highjob	1.1155	1.1003	0.0152	1.3588
48h_lowjob	1.1177	1.1060	0.0117	1.0509
Overall (weighted)	1.1168	1.1038	0.0131	1.1720
Overall (avg. of 6 cases)	1.1169	1.1039	0.0130	1.1655

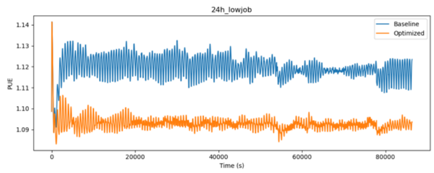
Generalization and variable screening results Unless otherwise stated, the time-series results and summary statistics presented in this section for the six unseen workloads were obtained using the representative models trained with random seed 42. Repeatability and statistical variability across different random seeds are analyzed separately in the subsequent robustness analysis.

For the potential optimization variable CDU_dp_nom, the experimental results are shown in Figure 15. The results indicate that, across the six test cases, the optimized PUE curves are consistently lower than those under the baseline condition, and both the average PUE and the major quantiles decrease. This suggests that CDU_dp_nom provides relatively stable optimization benefits under different workload conditions. Therefore, CDU_dp_nom has a strong influence on system energy efficiency and exhibits clear optimization potential. However, the post-hoc thermal assessment shows that this energy-saving benefit can be accompanied by increased maximum return-water temperatures. It should therefore be regarded as a promising candidate for further safety-constrained optimization and validation, rather than as a directly deployable setpoint based solely on its PUE improvement. More detailed results are provided in Table 5.

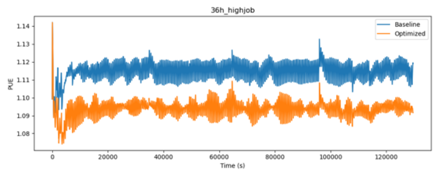
For the potential optimization variable CTWP_Nrel_stageUp, the experimental results are shown in Figure 16. The changes in PUE remain close



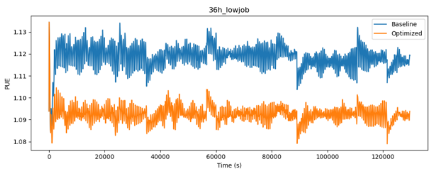
(a) 24-h high-load



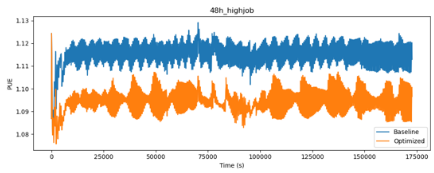
(b) 24-h low-load



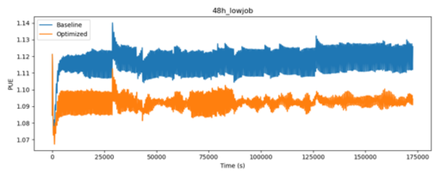
(c) 36 h high-load



(d) 36 h low-load



(e) 48 h high-load



(f) 48 h low-load

Figure 15. Comparison of PUE under baseline and BO-optimized control strategies for *CDU_dp_nom* under six workload conditions.

Table 5. Summary of PUE improvement after *CDU_dp_nom* optimization under six unseen workload conditions.

Case	Baseline Mean	Optimized Mean	Absolute PUE reduction	Relative PUE reduction (%)
24-h_highjob	1.1151	1.0945	0.0205	1.8398
24-h_lowjob	1.1196	1.0929	0.0267	2.3826
36h_highjob	1.1150	1.0939	0.0212	1.8975
36h_lowjob	1.1186	1.0926	0.0260	2.3267
48h_highjob	1.1155	1.0948	0.0207	1.8578
48h_lowjob	1.1177	1.0926	0.0252	2.2523
Overall (weighted)	1.1168	1.0935	0.0233	2.0869
Overall (avg. of 6 cases)	1.1169	1.0935	0.0234	2.0928

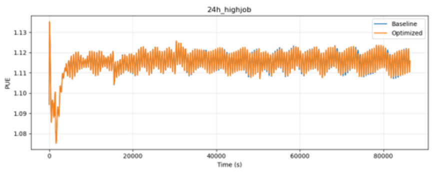
Table 6. Summary of PUE improvement after *CTWP_Nrel_stageUp* optimization under six unseen workload conditions.

Case	Baseline Mean	Optimized Mean	Absolute PUE reduction	Relative PUE reduction (%)
24-h_highjob	1.1151	1.1151	-0.0000	-0.0002
24-h_lowjob	1.1196	1.1195	0.0000	0.0043
36h_highjob	1.1150	1.1151	-0.0000	-0.0033
36h_lowjob	1.1186	1.1187	-0.0000	-0.0032
48h_highjob	1.1155	1.1155	-0.0000	-0.0008
48h_lowjob	1.1177	1.1177	0.0000	0.0001
Overall (weighted)	1.1168	1.1169	-0.0000	-0.0008
Overall (avg. of 6 cases)	1.1169	1.1169	-0.0000	-0.0005

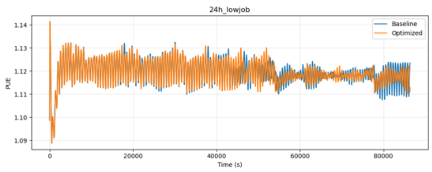
to zero across most of the six workload conditions, and no consistent reduction is observed. This indicates that the influence of CTWP_Nrel_stageUp is strongly dependent on whether the corresponding pump-staging threshold is activated by the workload trajectory. Under the current search range and experimental conditions, this variable therefore exhibits limited and unstable optimization potential. More detailed results are provided in Table 6.

For the potential optimization variable ps_stage_down, the experimental results are shown in Figure 17. Compared with the baseline control strategy, the optimized PUE is reduced under all six test workloads, indicating a consistent positive optimization effect. However, the magnitude of improvement is generally smaller than that achieved for adj and CDU_dp_nom. Overall, ps_stage_down exhibits modest but relatively stable optimization potential, although its realized benefit depends on the workload state and the corresponding staging-control behavior. More detailed results are provided in Table 7.

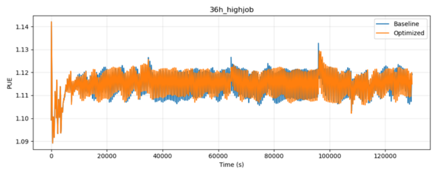
The differences among these variables are consistent with their distinct roles in the cooling-system control hierarchy. *CDU_dp_nom* directly modifies the differential-pressure reference continuously tracked by the CDU pump-speed feedback loop and can therefore continuously influence the pump-speed command and circulation state without requiring a staging transition. By contrast,



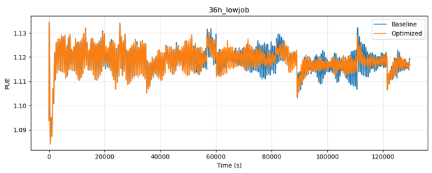
(a) 24-h high-load



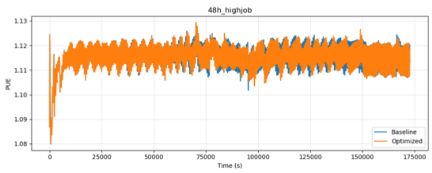
(b) 24-h low-load



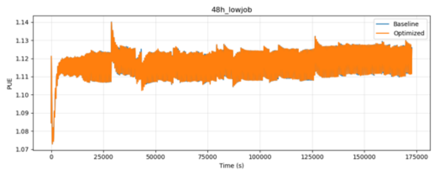
(c) 36 h high-load



(d) 36 h low-load

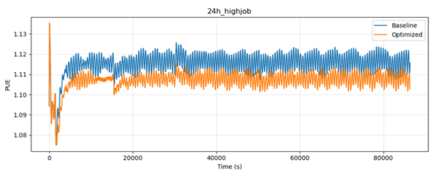


(e) 48 h high-load

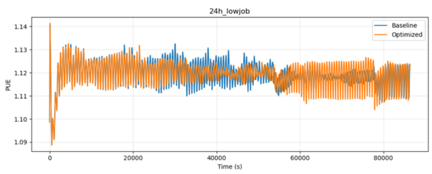


(f) 48 h low-load

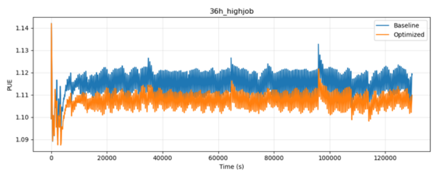
Figure 16. Comparison of PUE under baseline and BO-optimized control strategies for *CTWP_Nrel_stageUp* under six workload conditions.



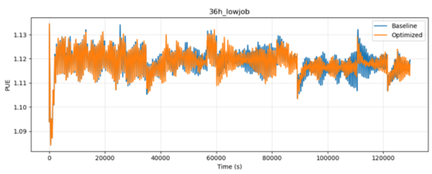
(a) 24-h high-load



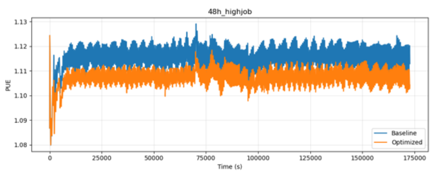
(b) 24-h low-load



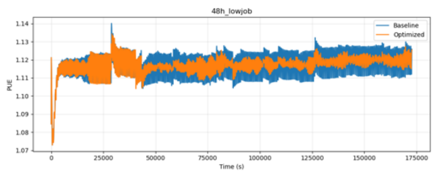
(c) 36 h high-load



(d) 36 h low-load



(e) 48 h high-load



(f) 48 h low-load

Figure 17. Comparison of PUE under baseline and BO-optimized control strategies for *ps_stage_down* under six workload conditions.

$CTWP\_Nrel\_stageUp$ and ps_stage_down are event-triggered staging thresholds associated with pump-group stage-up and cooling-tower stage-down, respectively. $CTWP\_Nrel\_stageUp$ is expected to affect system energy use mainly when the normalized pump-speed signal crosses the stage-up threshold and consequently changes the number of operating pumps. Similarly, the influence of ps_stage_down is expected to become more apparent when its pressure-based stage-down condition changes the number of active cooling towers or alters the staging sequence.

Accordingly, when the workload trajectory does not cross the relevant threshold, or when a change in the threshold does not alter the original staging sequence, the system may remain within an approximately flat response region. In contrast, when the threshold change leads to a different staging state, discrete changes in pump-group or cooling-tower configuration and energy consumption may occur. This interpretation is broadly consistent with the results obtained using the representative seed-42 models under the six test workloads. CDU_dp_nom produced directionally consistent PUE reductions in all six cases, with relative improvements ranging from 1.84% to 2.38%. By contrast, the PUE changes associated with $CTWP\_Nrel\_stageUp$ remained close to zero overall and did not show a consistent positive improvement, indicating that its influence strongly depends on whether the workload alters the corresponding pump-staging behavior. ps_stage_down produced positive improvements in all six workloads, with an average relative improvement of approximately 0.37% for the seed-42 model, although its recommended values varied mainly between 15 and 35 depending on the operating

Table 7. Summary of PUE improvement after *ps_stage_down* optimization under six unseen workload conditions.

Case	Baseline Mean	Optimized Mean	Absolute PUE reduction	Relative PUE reduction (%)
24-h_highjob	1.1151	1.1070	0.0081	0.7238
24-h_lowjob	1.1196	1.1193	0.0003	0.0282
36h_highjob	1.1150	1.1074	0.0077	0.6886
36h_lowjob	1.1186	1.1185	0.0002	0.0167
48h_highjob	1.1155	1.1075	0.0080	0.7197
48h_lowjob	1.1177	1.1172	0.0005	0.0451
Overall (weighted)	1.1168	1.1127	0.0041	0.3706
Overall (avg. of 6 cases)	1.1169	1.1128	0.0041	0.3704

condition. This indicates that the variable has modest but relatively consistent optimization potential, while its realized benefit remains dependent on the workload state and cooling-tower staging behavior.

The LCB acquisition function used in this study does not require gradients of the FMU response. The practical challenge is that each BO run uses only three full FMU simulations, providing the GPR surrogate with limited observations for distinguishing approximately flat response regions from discrete jumps induced by equipment-staging transitions. The additional random-seed experiments also show that the staging-threshold variables exhibit greater model sensitivity than the continuously acting setpoint variables. Because the threshold-crossing frequency, equipment-staging sequence, and dwell time in the relevant pump-group and cooling-tower states were not recorded in this study, the foregoing discussion should be regarded as a mechanism-based interpretation consistent with the implemented control logic and observed results, rather than as a rigorous causal demonstration.

In terms of computational cost, one complete Bayesian optimization search consisted of two initial sampling simulations and one subsequent iterative evaluation, resulting in a total of three full FMU simulations. For a 24-h workload, each simulation required approximately 20 min, leading to a total wall-clock time of approximately 1 h for one BO search. All experiments were conducted on a personal computer equipped with an Intel(R) Core(TM) i7-10510U CPU @ 1.80 GHz, with four physical cores and eight logical processors. For longer workloads, the simulation time increased approximately linearly with workload duration. For comparison, an exhaustive search over the adj range $[0.9, 1.4]$ with a step size of 0.01 would require 51 full simulations. Therefore, in terms of the number of simulation calls required for a single parameter search, the proposed framework reduces the search cost by approximately 17× relative to exhaustive search. This comparison refers only to the search cost of one BO run and does not include the additional FMU evaluations used for the six unseen workload tests, repeated BO-seed experiments, or the 15 independent workload-realization tests.

Robustness and Repeatability across Bayesian-Optimization Seeds

To evaluate the sensitivity of Bayesian optimization to Gaussian-process random initialization under a limited simulation budget, the model was independently trained using three random seeds, namely 7, 17, and 42, for each of the four control variables. Each trained model was evaluated on the same six unseen workload scenarios, without retraining or updating using test data. Consequently, 18 seed–scenario optimization results were obtained for each variable. The training data, search bounds, acquisition function, and simulation budget were kept identical across repeated runs so that the observed differences primarily reflected the influence of random initialization.

Repeatability was evaluated in terms of both the recommended settings and the realized simulation improvements. Recommendations were classified according to whether the three seeds converged to the same value, the same region, or produced clearly divergent results under the same scenario. Realized performance was evaluated using the mean PUE improvement, 95% confidence interval, and number of positive improvements across the 18 evaluations. In addition, the mean improvement over the six scenarios was calculated for each seed, and the standard deviation across these seed-level means was used to quantify sensitivity to random initialization. The overall PUE improvement distributions for all four variables are illustrated in Figure 18.

The results are summarized in Table 8. adj and CDU_dp_nom are highly robust to random initialization. All three models consistently recommended 1.4 and 10, respectively, across all test scenarios. Their seed-level standard deviations were both zero, and all 18 evaluations produced

Table 8. Robustness and repeatability of Bayesian optimization across three random seeds.

Variable	Recommended range	Mean PUE improvement (%)	Seed-level SD (%)	95% CI across 18 evaluations (%)	Positive results	Recommendation consistency
adj	[1.4, 1.4]	1.166	0.000	[1.072, 1.259]	18/18	Same value in 6/6 cases
CDU_dp_nom	[10, 10]	2.093	0.000	[1.975, 2.211]	18/18	Same value in 6/6 cases
$CTWP\_Nrel\_stageUp$	[35, 99]	0.016	0.028	[-0.019, 0.051]	6/18	Divergent in 1/6 cases
ps_stage_down	[15, 35]	0.402	0.054	[0.230, 0.573]	18/18	Same region in 6/6 cases

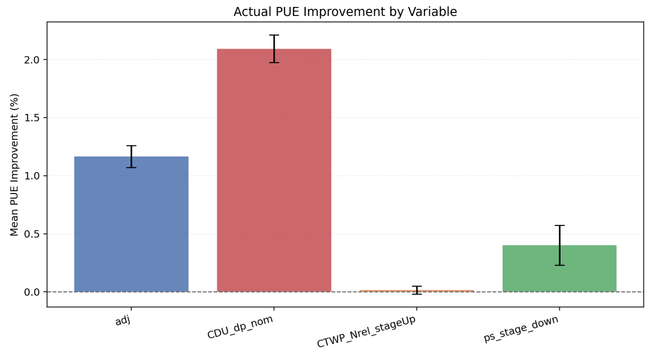


Figure 18. Overall actual PUE improvement for the four control variables across three Gaussian-process random seeds and six unseen workload scenarios.

positive PUE improvements (sign test: $p < 0.001$ under the null hypothesis of equal probability of positive and negative outcomes). The corresponding mean improvements were 1.166% and 2.093%, indicating that their observed benefits were not attributable to a particular random seed.

In contrast, the recommendations for CTWP_Nrel_stageUp mainly switched between 35 and 99, with seed-dependent divergence occurring in the 48-h low-load scenario. Its mean improvement was only 0.016%, its 95% confidence interval included zero, and only 6 of the 18 evaluations showed positive improvement. These findings indicate that the optimization effect of this variable is weak and more sensitive to both workload conditions and model initialization.

The recommendations for ps_stage_down were primarily concentrated around 15 and 35. Although some segment-level differences appeared among seeds under high-load conditions, all 18 evaluations achieved positive improvements, with a mean improvement of 0.402%. Overall, repeated runs did not change the principal assessment of variable-specific optimization potential: adj and CDU_dp_nom exhibited high repeatability, whereas equipment-staging thresholds showed stronger dependence on workload conditions.

3.2.4 *Post-hoc Thermal Risk Assessment*

In addition to comparing energy-efficiency metrics, this study further conducts a post-hoc thermal risk assessment of the optimization results. Specifically, the maximum secondary-side return water temperature and the maximum primary-side return water temperature under the baseline strategy and the optimized strategy are extracted and compared for each operating condition, in order to examine whether the optimization process introduces any evident increase in thermal risk while reducing PUE.

This study first performs a comparative analysis of return water temperatures for the optimization results of the representative variable adj. Specifically, the maximum secondary-side return water temperature and the maximum primary-side return water temperature under the baseline strategy and the optimized strategy are calculated for each test workload, and the resulting differences are used to assess the trend of thermal-state changes after optimization. Among them, Table 9 presents the comparison results for the maximum secondary-side return water temperature of adj under the six test workload conditions, while Table 10 gives the corresponding comparison results for the maximum primary-side return water temperature.

For CDU_dp_nom, the thermal response shows a clearer trade-off. Although it achieves the largest PUE reduction among the tested variables, increases in maximum return-water temperature are observed under several workload conditions. Therefore, its strong energy-saving potential should not be interpreted as direct deployment readiness. The current soft temperature penalty does not constitute a hard safety constraint, and practical deployment would require calibrated temperature limits, appropriate safety margins, or constrained/multi-objective optimization.

It should be noted that the temperature threshold adopted in the objective function of this study is mainly used as a conservative soft-penalty reference for suppressing parameter settings with potential high-temperature risk, rather than as a strict physical safety boundary of the digital twin model under all operating conditions. If explicit real-world engineering temperature thresholds become available in future work, the current reference threshold in the objective function can be correspondingly adjusted and mapped in combination with model calibration results. This may be achieved, for example, through safety-margin contraction, empirical multiplier correction, or equivalent-threshold transformation, so that the safety requirements of the real system can be translated into constraint criteria within the model and thereby better satisfy practical operational needs. Accordingly, the focus of the present analysis is not to determine whether an absolute threshold violation occurs, but rather to examine whether the optimized strategy leads to a significant deterioration in return-water temperature risk and whether the temperature penalty term effectively suppresses overly aggressive candidate parameters. For the remaining candidate variables, including CDU_dp_nom, $CTWP\_Nrel\_stageUp$, and ps_stage_down, the corresponding temperature comparison results are summarized in Appendix F.

Table 9. Comparison of maximum secondary-side return water temperature for *adj* under baseline and optimized strategies.

Case	Base.Max $\mathbf{T}_{\mathrm{sec,r}}$ (°C)	Opt.Max $\mathbf{T}_{\mathrm{sec,r}}$ (°C)	Δ Max $\mathbf{T}_{\mathrm{sec,r}}$ (°C)	Thres. (°C)
24-h_highjob	38.290	38.518	0.228	30.0
24-h_lowjob	34.183	34.546	0.363	30.0
36h_highjob	37.272	37.281	0.009	30.0
36h_lowjob	36.593	36.749	0.156	30.0
48h_highjob	38.386	38.535	0.149	30.0
48h_lowjob	37.107	35.938	-1.169	30.0

Table 10. Comparison of maximum primary-side return water temperature for *adj* under baseline and optimized strategies.

Case	Base.Max $\mathbf{T}_{\mathrm{prim,r}}$ (°C)	Opt.Max $\mathbf{T}_{\mathrm{prim,r}}$ (°C)	Δ Max $\mathbf{T}_{\mathrm{prim,r}}$ (°C)	Thres. (°C)
24-h_highjob	38.778	38.943	0.165	35.0
24-h_lowjob	34.837	34.670	-0.167	35.0
36h_highjob	37.926	37.667	-0.259	35.0
36h_lowjob	36.498	36.723	0.225	35.0
48h_highjob	38.820	38.840	0.020	35.0
48h_lowjob	37.417	36.645	-0.772	35.0

The comparison results show that, under the tested workload conditions, the optimized strategy does not necessarily reduce the maximum return water temperature in every case, and slight increases can still be observed under some individual conditions. However, the overall magnitude of such increases remains limited, indicating that the current optimization process does not drive the system toward a clearly worse thermal state solely in pursuit of lower PUE. In other words, the temperature penalty term introduced in this study retains the ability to explore optimization potential while also providing a certain degree of suppression against overly aggressive search conducted purely along the energy-efficiency direction.

4 Conclusion

This study presents and validates a high-fidelity, gray-box methodology driven by Bayesian optimization for systematically exploring the energy optimization potential of data center liquid-cooling control systems. The core contribution of this work lies in providing a secure, interpretable pre-evaluation framework that translates theoretical optimization into actionable engineering insights without risking physical infrastructure. Through rigorous stress testing under statistically authentic dynamic workloads generated by the enhanced RAPS model, this study quantitatively reveals that different control parameters exhibit distinct optimization profiles and effectiveness.

Across the six unseen workload scenarios, CDU_dp_nom produced the largest and most directionally consistent PUE reductions, followed by the stable improvements obtained with adj. However, the post-hoc thermal assessment showed that the PUE improvement of CDU_dp_nom was accompanied by increases in maximum return-water temperature under several workload conditions. Therefore, its performance should be interpreted as evidence of strong energy-saving potential rather than immediate deployment readiness, and practical application would require more conservative thermal constraints or constrained/multi-objective optimization.

By contrast, $CTWP\_Nrel\_stageUp$ produced changes close to zero overall and did not demonstrate a consistently beneficial optimization effect. ps_stage_down yielded more modest but consistently positive improvements, although its recommended settings remained dependent on the workload condition. These differences are consistent with their positions in the control hierarchy: CDU_dp_nom and adj act through continuously tracked upper-level setpoints, whereas $CTWP\_Nrel\_stageUp$ and ps_stage_down influence the system through event-triggered staging logic.

The repeated experiments further strengthened these conclusions. Across Bayesian-optimization models trained with random seeds 7, 17, and 42, adj and CDU_dp_nom showed highly consistent recommendations and realized improvements, whereas the staging-threshold variables exhibited greater workload and model sensitivity. In addition, the fixed `adj_seed42` model achieved positive PUE improvements in all 15 independently generated unseen 24-h workloads, providing additional evidence of zero-shot robustness to workload-realization variability. These results indicate that continuously acting pressure and temperature setpoints should be prioritized during upper-level variable screening, while staging thresholds require workload-aware and trigger-aware evaluation.

This study has several limitations that also motivate future research. First, current optimizations evaluate variables individually or in limited combinations. Future work must transition towards high-dimensional co-optimization, exploring the complex coupling effects when multiple diverse variables are optimized simultaneously. Second, safety-constrained dynamic setpoint adjustment will be explored to improve responsiveness to transient workload changes and facilitate the transition from offline segment-wise optimization to continuous online deployment in real data centers. Finally, with the rapid evolution of artificial intelligence, exploring the integration of Large Language Models (LLMs) as advanced reasoning optimizers [34] or automated optimization potential evaluators represents a highly promising direction for future work to further enhance the interpretability and adaptability of data center thermal management, though it lies beyond the scope of the present study.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The authors declare no conflicts of interest.

AI Use Statement

The authors declare that ChatGPT-5 was used for language editing and Chinese-to-English translation of the manuscript. The authors have carefully reviewed, revised, and verified the AI-assisted output and take full responsibility for the content of the manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., ... & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM, 53*(4), 50-58. [CrossRef]

[2] International Energy Agency. (2025). *Energy and AI*. IEA. Retrieved from https://www.iea.org/reports/energy-and-ai

[3] Masanet, E., Shehabi, A., Lei, N., Smith, S., & Koomey, J. (2020). Recalibrating global data center energy-use estimates. *Science, 367*(6481), 984-986. [CrossRef]

[4] Zhou, D., Du, L., Xu, R., Song, J., Chen, N., Zheng, S., & Fan, Y. (2025). Accelerating liquid cooling adoption in data centers: Necessity, challenges and solutions. *Technology Review for Carbon Neutrality, 1*, 9550014. [CrossRef]

[5] Shahi, P., Deshmukh, A. P., Hurnekar, H. Y., Saini, S., Bansode, P., Kasukurthy, R., & Agonafer, D. (2022). Design, development, and characterization of a flow control device for dynamic cooling of liquid-cooled servers. *Journal of Electronic Packaging, 144*(4), 041008. [CrossRef]

[6] Chen, H., Han, Y., Tang, G., & Zhang, X. (2020). A dynamic control system for server processor direct liquid cooling. *IEEE Transactions on Components, Packaging and Manufacturing Technology, 10*(5), 786-794. [CrossRef]

[7] Lucchese, R., Varagnolo, D., & Johansson, A. (2020). Controlled direct liquid cooling of data servers. *IEEE Transactions on Control Systems Technology, 29*(6), 2325-2338. [CrossRef]

[8] ASHRAE Technical Committee 9.9. (2015). *Thermal Guidelines for Data Processing Environments* (4th ed.). Atlanta, GA: ASHRAE. Retrieved from https://www.ashrae.org/technical-resources/bookstore/datacom-series#thermalguidelines

[9] Fulpagare, Y., & Bhargav, A. (2015). Advances in data center thermal management. *Renewable and Sustainable Energy Reviews, 43*, 981-996. [CrossRef]

[10] Kahil, H., Sharma, S., Välisuo, P., & Elmusrati, M. (2025). Reinforcement learning for data center energy efficiency optimization: A systematic literature review and research roadmap. *Applied Energy, 389*, 125734. [CrossRef]

[11] Zhang, H., Shao, S., Xu, H., Zou, H., & Tian, C.

(2021). A survey on data center cooling systems: Technology, power consumption modeling and control strategy optimization. *Journal of Systems Architecture*, *119*, 102253. [CrossRef]

[12] Lazic, N., Lu, T., Boutilier, C., Ryu, M., Wong, E., Roy, B., & Imwalle, G. (2018). Data center cooling using model-predictive control. *Advances in Neural Information Processing Systems*, *31*.

[13] Li, Y., Wen, Y., Tao, D., & Guan, K. (2020). Transforming cooling optimization for green data center via deep reinforcement learning. *IEEE Transactions on Cybernetics*, *50*(5), 2002-2013. [CrossRef]

[14] Lin, X., Guo, Q., Yuan, D., & Gao, M. (2023). Bayesian optimization framework for HVAC system control. *Buildings*, *13*(2), 314. [CrossRef]

[15] Gao, J. (2014). *Machine learning applications for data center optimization*. Google Research. Retrieved from https://research.google/pubs/pub42542/

[16] Naug, A., Guillen-Perez, A., Kumar, V., Greenwood, S., Brewer, W., Ghorbanpour, S., ... & Sarkar, S. (2026). Lc-opt: Benchmarking reinforcement learning and agentic ai for end-to-end liquid cooling optimization in data centers. *Advances in Neural Information Processing Systems*, *38*.

[17] Li, Y., O'Neill, Z., Zhang, L., Chen, J., Im, P., & DeGraw, J. (2021). Grey-box modeling and application for building energy simulations-A critical review. *Renewable and Sustainable Energy Reviews*, *146*, 111174. [CrossRef]

[18] Brewer, W., Maiterth, M., Kumar, V., Wojda, R., Bouknight, S., Hines, J., ... & Wang, F. (2024, November). A digital twin framework for liquid-cooled supercomputers as demonstrated at exascale. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1-18). IEEE. [CrossRef]

[19] Wang, L., Rodriguez, M. A., & Lipovetzky, N. (2025, October). HPCsim: A High-Level Simulation and Workload Data Schema Framework for HPC Workload Management Research. In *2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 5314-5320). IEEE. [CrossRef]

[20] Maiterth, M., Brewer, W. H., Kuruvella, J. S., Dey, A., Islam, T. Z., Kabir, R., ... & Wang, F. (2025, November). HPC digital twins for evaluating scheduling policies, incentive structures and their impact on power and cooling. In *Proceedings of the SC'25 Workshops of the International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1959-1969). [CrossRef]

[21] Cirou. (2024). *Adastra jobs MI250 15days* [Dataset]. CINES. [CrossRef]

[22] Athavale, J., Bash, C., Brewer, W., Maiterth, M., Milojicic, D., Petty, H., & Sarkar, S. (2024). Digital twins for data centers. *Computer*, *57*(10), 151-158. [CrossRef]

[23] Modelica Association. (2023). *Modelica standard library version 4.0.0*. Retrieved from https://github.com/modelica/ModelicaStandardLibrary

[24] Casella, F., & Leva, A. (2003, November). Modelica open library for power plant simulation: design and experimental validation. In *Proceeding of the 2003 Modelica conference*, Linkoping, Sweden. https://modelica.org/events/Conference2003/papers/h08_Leva_original.pdf

[25] Greenwood, M. S., Fugate, D. L., & Cetiner, M. S. (2017). *A templated approach for multi-physics modeling of hybrid energy systems in Modelica*. Oak Ridge National Laboratory (ORNL), ORNL/TM-2017/399. Retrieved from https://www.osti.gov/biblio/1427611

[26] Kumar, V., Greenwood, S., Brewer, W., Grant, D., Parkison, N., & Williams, W. (2024). Thermo-fluid modeling framework for supercomputer digital twins: Part 1, demonstration at exascale. *Proceedings of the American Modelica Conference 2024*, Storrs, CT, USA, October 14-16, 2024. Oak Ridge National Laboratory (ORNL). Retrieved from https://www.osti.gov/biblio/2480044]

[27] Wetter, M., Zuo, W., Nouidui, T. S., & Pang, X. (2014). Modelica buildings library. *Journal of Building Performance Simulation*, *7*(4), 253-270. [CrossRef]

[28] Greenwood, M. S., Cetiner, M. S., Fugate, D. L., Hale, R. E., Harrison, T. J., & Qualls, A. L. (2017). *TRANSFORM - TRANsient Simulation Framework of Reconfigurable Models* [Computer software]. U.S. Department of Energy, Oak Ridge National Laboratory. Retrieved from https://www.osti.gov/biblio/1395459

[29] Blochwitz, T., Otter, M., Åkesson, J., Arnold, M., Clauss, C., Elmqvist, H., ... & Viel, A. (2012). Functional mockup interface 2.0: The standard for tool independent exchange of simulation models. In *9th international modelica conference* (pp. 173-184). The Modelica Association. [CrossRef]

[30] Rasmussen, C. E. (2003). Gaussian processes in machine learning. In *Summer school on machine learning* (pp. 63-71). Berlin, Heidelberg: Springer Berlin Heidelberg. [CrossRef]

[31] Krause, A., & Ong, C. S. (2011). Contextual Gaussian process bandit optimization. *Advances in Neural Information Processing Systems*, *24*.

[32] Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. (2009). Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*. [CrossRef]

[33] Frazier, P. I. (2018). A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*. [CrossRef]

[34] Yang, C., Wang, X., Lu, Y., Liu, H., Le, Q. V., Zhou, D., & Chen, X. (2024). Large language models as optimizers. *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*.

Appendix

A Pseudocode for Synthetic Workload Generation

Algorithm 1: Seeded Synthetic Workload Generation and Finite-Horizon Recording

Input: candidate count N; horizon T; integer seed s; power profile p; Frontier configuration $\mathcal{C}$
Output: `generated_workload.csv`, `job_history.csv`, and `run_metadata.json`

```
Initialize the Python and NumPy random-number generators with s;
Reset the cumulative arrival clock and set a ← 0;
Read τ = 900 s, n_max = 3000, q = 15 s, w_min = 3600 s,
  w_max = 43200 s, μ_w = 23400 s, and σ_w = 6600 s from C;
if p = low then
    D_p ← Beta(1, 5);
else
    if p = high then
        D_p ← Beta(5, 1);
    else
        D_p ← U(0, 1);
    end
end
J ← [];
for j ← 1 to N do
    Draw n_j ~ DiscreteUniform{1, ..., n_max};
    Sample the configured job name and account;
    Draw u_j^CPU, u_j^GPU ~iid D_p;
    Draw w_j^* ~ TruncatedNormal(μ_w, σ_w; w_min, w_max);
    w_j ← ⌊w_j^*/3600⌋ × 3600;
    Form constant CPU and GPU utilization traces from
      u_j^CPU and u_j^GPU with length w_j/q;
    Sample the configured end-state field and scheduling
      priority;
    if j > 1 then
        a ← a + Exponential(mean = τ);
    end
    Append job J_j = (j, a, n_j, w_j, traces, metadata) to J;
end
J_T ← {J_j ∈ J : 0 ≤ a_j < T};
Write the generated fields of J_T to generated_workload.csv;
P ← SortBySubmitTime(J);
Initialize the scheduler queue Q, running set R, and
  completed set F as empty;
for t ← 0 to T − 1 do
    Move jobs with a_j ≤ t from P to Q;
    Apply the selected scheduling policy to Q and R;
    Advance resource, power, and optional cooling models
      by one second;
    Move jobs completed at this tick from R to F;
end
Tag F, R, and Q as COMPLETED, RUNNING, and QUEUED,
  respectively;
Write their concatenation to job_history.csv; write s, T, p, C,
  and row counts to run_metadata.json;
```

B Contextual State Features for Bayesian Optimization

Table A1. Contextual state features used for Bayesian optimization and their segment-level aggregation rules.

Feature Category	Instantaneous Base Variables	Segment-level Aggregation Rule
Heat load	Q_flow_total_sum	Q_flow_total_sum: mean, std, max
Primary-side flow	m_flow_prim_mean, m_flow_prim_min	m_flow_prim_mean: mean, std; m_flow_prim_min: min
Secondary-side flow	m_flow_sec_mean, m_flow_sec_min	m_flow_sec_mean: mean, std; m_flow_sec_min: min
Primary-side temperature difference	delta_T_prim_mean, delta_T_prim_max	delta_T_prim_mean: mean, std; delta_T_prim_max: max
Secondary-side temperature difference	delta_T_sec_mean, delta_T_sec_max	delta_T_sec_mean: mean, std; delta_T_sec_max: max
Primary-side return temperature	T_prim_r_C_mean, T_prim_r_C_max	T_prim_r_C_mean: mean; T_prim_r_C_max: max

C Control Mechanism and Formula Description of Variable adj

In the data center cooling system model adopted in this study, adj is embedded in the control logic of the cooling tower water (CTW) loop. In essence, it is a correction parameter for the cooling tower target temperature setpoint, and it is located in the control module CSPumpAndStagingControl. This module serves as the core dynamic control unit of the cooling tower water system. Based on input signals such as the CTWR return header pressure, the hot-side supply temperature, and the outdoor wet-bulb temperature, it jointly determines the cooling tower water pump speed, the number of operating pumps, the number of active cooling towers, and the tower-side target temperature setpoint. Its control objectives can be summarized in two aspects. First, it maintains the CTWR return header pressure within the prescribed range through a PI controller. Second, it ensures the stability of the HTW supply temperature through pump staging and cooling tower staging control while coordinating the overall operating state of the system.

Within this control logic, adj is located inside the $p\_CTWR\_setpoint$ module, where it participates in the calculation of the cooling tower approach temperature difference and further determines the cooling tower target temperature setpoint $T_{CT,\text{setpoint}}$. Specifically, the controller first calculates the cooling tower approach based on the outdoor wet-bulb temperature T_{owb}, which is expressed as

$$T_{AppWat} = CT_{corr}\Big(T_{WetBul} = to\_degF(T_{owb})\,,\, adj = adj \Big). \tag{4}$$

Here, the function CT_{corr} expresses the cooling tower approach temperature difference T_{AppWat} as a function of the outdoor wet-bulb temperature and adjusts it through the correction parameter adj. According to the model implementation, its expression can be written as

$$T_{AppWat} = c_1 + c_2 T_{owb} + c_3 T_{owb}^2, \tag{5}$$

where the coefficients satisfy

$$c_1 = 25.0 - 0.6 \times (27 - 17\,adj), \tag{6}$$
$$c_2 = 0.73 + 0.0067 \times (27 - 17\,adj), \tag{7}$$
$$c_3 = -1. \tag{8}$$

It can thus be seen that adj does not act directly on the actuators. Instead, it first affects the values of c_1 and c_2, thereby regulating the magnitude of the cooling tower approach temperature difference T_{AppWat}. Then, by adding the wet-bulb temperature and the approach temperature difference, an intermediate setpoint for the cooling tower target temperature is obtained:

$$CT\_temp\_setpoint = T_{owb,F} + T_{AppWat}, \tag{9}$$

where $T_{owb,F}$ denotes the outdoor wet-bulb temperature expressed in degrees Fahrenheit. Furthermore, to prevent the setpoint from becoming excessively low, the model imposes a lower bound on this temperature, and the actual cooling tower target temperature setpoint is finally obtained as

$$T_{CT,\text{setpoint}} = from\_degF\Big(\max\big(CT\_temp\_setpoint,\, 54^{\circ}\text{F}\big)\Big). \tag{10}$$

As can be seen, the effect of adj on the system does not lie in directly changing the pump speed, valve opening, or equipment on/off status. Instead, it indirectly adjusts the cooling tower target temperature setpoint $T_{CT,\text{setpoint}}$ by modifying the cooling tower approach temperature difference T_{AppWat}. When adj increases, T_{AppWat} increases accordingly, which raises $T_{CT,\text{setpoint}}$, making the control strategy relatively more conservative. Conversely, when adj decreases, $T_{CT,\text{setpoint}}$ is reduced, and the control strategy becomes more aggressive. Therefore, the essence of adj can be understood as an upper-level correction parameter in the cooling tower temperature setpoint layer.

Furthermore, the influence of adj continues to propagate along the control chain. The pressure setpoint and temperature setpoint generated by the $p\_CTWR\_setpoint$ module are first fed into the downstream control loops. Among them, the pressure setpoint and the measured pressure signal are filtered and then input into the PI controller to generate the CTWP speed command. The speed command is subsequently passed to the $CTWP\_Staging$ module to determine whether the pump group should be staged up or staged down. Meanwhile, the tower-side target temperature setpoint, together with signals such as the HTWS temperature and the CTWR gauge pressure, is also fed into the $CT\_Staging$ module to determine the switching of the number of active cooling towers. Therefore, although adj acts only at the temperature setpoint layer, it can further influence the cooling tower heat exchange condition, the pump operating state, and the total system energy consumption through the combined effects of temperature targeting, pressure regulation, and equipment staging logic, thereby becoming a key setpoint capable of significantly affecting system-level performance.

D Relationship Plots of the Training Samples

Figures A1, A2, and A3 show the relationships between each candidate optimization variable and the segment-wise mean PUE in the training samples, corresponding to ps_stage_down, CDU_dp_nom, and $CTWP\_Nrel\_stageUp$, respectively.

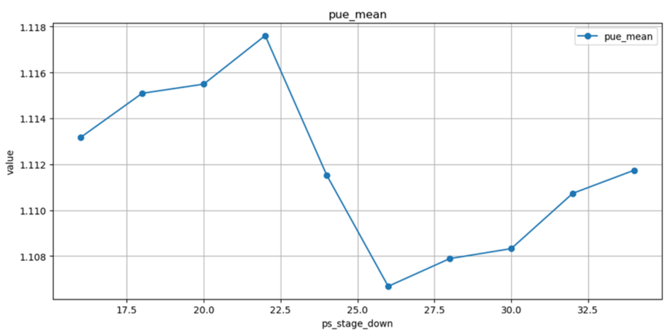


Figure A1. Relationship between ps_stage_down and Segment-wise Mean PUE in the Training Samples.

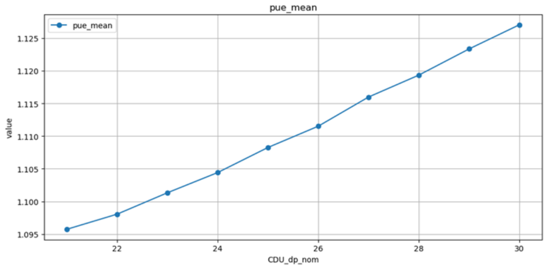


Figure A2. Relationship between CDU_dp_nom and Segment-wise Mean PUE in the Training Samples.

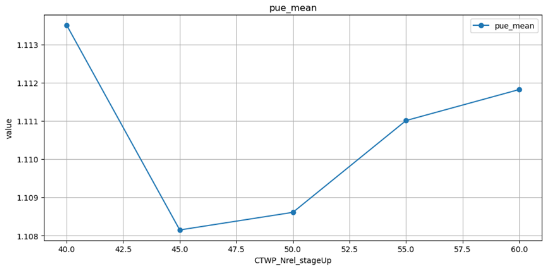


Figure A3. Relationship between $CTWP\_Nrel\_stageUp$ and Segment-wise Mean PUE in the Training Samples.

E Power Distributions of Six Synthetic Workloads

This appendix presents the power distribution profiles for the six synthetic workload scenarios generated in this study. To facilitate statistical analysis and improve visual clarity, jobs that recorded exactly zero energy consumption at the end of the statistical period were excluded.

As illustrated in Figure A4, the generated workloads encompass three time scales (24-h, 36-h, and 48-h) under both high-load and low-load conditions. Across all scenarios, the power distributions consistently exhibit a pronounced right-skewed and long-tail characteristic. This confirms that the improved workload generator successfully replicates the "power heterogeneity" typical of modern supercomputers—characterized by massive low-power test jobs interspersed with occasional, highly intensive scientific tasks.

Crucially, as the simulation duration extends and load intensity increases, the fundamental long-tail shape remains remarkably robust without unrealistic deviations. These profiles demonstrate that the synthetic workloads provide a rigorous and physically realistic testing ground for evaluating the Bayesian optimization strategies detailed in Appendix F.

F Post-hoc Thermal Risk Assessment

Tables A2 and A3 present the secondary- and primary-side return water temperature comparisons for $CTWP\_Nrel\_stageUp$, respectively. Tables A4 and A5 present the corresponding comparisons for CDU_dp_nom. Tables A6 and A7 present the comparisons for ps_stage_down.

Table A2. Comparison of maximum secondary-side return water temperature for *CTWP_Nrel_stageUp* under baseline and optimized strategies.

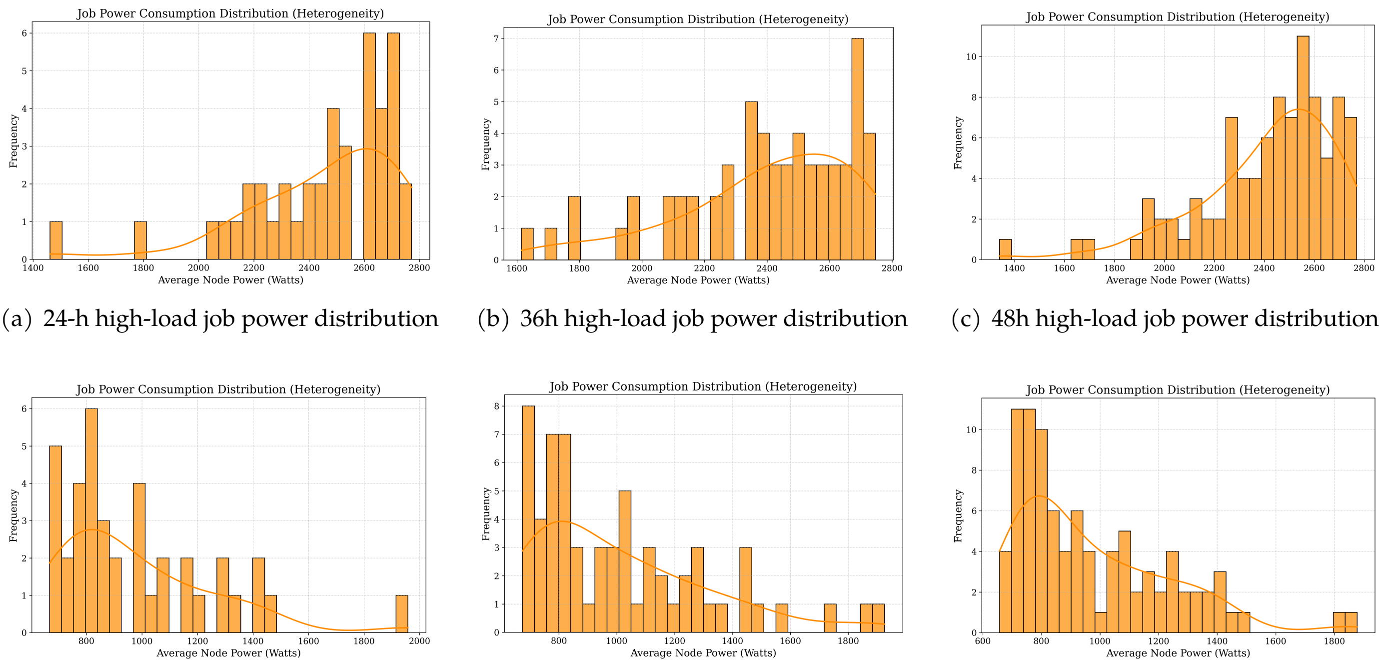
Case	Base.Max $T_{\mathrm{sec},r}$ (°C)	Opt.Max $T_{\mathrm{sec},r}$ (°C)	ΔMax $T_{\mathrm{sec},r}$ (°C)	Thres. (°C)
24-h_highjob	38.290	38.290	0.000	30.0
24-h_lowjob	34.183	34.615	0.432	30.0
36h_highjob	37.272	37.207	-0.065	30.0
36h_lowjob	36.593	36.601	0.008	30.0
48h_highjob	38.386	38.386	0.000	30.0
48h_lowjob	37.107	36.915	-0.192	30.0

Table A3. Comparison of maximum primary-side return water temperature for *CTWP_Nrel_stageUp* under baseline and optimized strategies.

Case	Base. Max $T_{\mathrm{prim},r}$ (°C)	Opt.Max $T_{\mathrm{prim},r}$ (°C)	Δ Max $T_{\mathrm{prim},r}$ (°C)	Thres. (°C)
24-h_highjob	38.778	38.778	0.000	35.0
24-h_lowjob	34.837	34.729	-0.108	35.0
36h_highjob	37.926	37.824	-0.102	35.0
36h_lowjob	36.498	36.650	0.152	35.0
48h_highjob	38.820	38.820	0.000	35.0
48h_lowjob	37.417	36.811	-0.606	35.0

Table A4. Comparison of maximum primary-side return water temperature for *CDU_dp_nom* under baseline and optimized strategies.

Case	Base. Max $T_{\mathrm{prim},r}$ (°C)	Opt.Max $T_{\mathrm{prim},r}$ (°C)	Δ Max $T_{\mathrm{prim},r}$ (°C)	Thres. (°C)
24-h_highjob	38.778	39.670	0.892	35.0
24-h_lowjob	34.837	38.776	3.939	35.0
36h_highjob	37.926	38.710	0.784	35.0
36h_lowjob	36.498	40.569	4.071	35.0
48h_highjob	38.820	39.369	0.549	35.0
48h_lowjob	37.417	37.987	0.570	35.0



(a) 24-h high-load job power distribution (b) 36h high-load job power distribution (c) 48h high-load job power distribution

(d) 24-h low-load job power distribution (e) 36h low-load job power distribution (f) 48h low-load job power distribution

Figure A4. Descriptive job-power distributions for the six synthetic workloads used to test Bayesian-optimization transfer across three durations and two power-intensity settings. Jobs with zero recorded energy at the end of the corresponding finite horizon are excluded. The scenario identifiers are retained for consistency with the optimization experiments; "highjob" and "lowjob" denote the imposed power-intensity settings and should not be interpreted as node-occupancy classes. Formal workload-generator validation is reported in Section 3.1.

Table A5. Comparison of maximum secondary-side return water temperature for *CDU_dp_nom* under baseline and optimized strategies.

Case	Base. Max $T_{\mathrm{sec},r}$ (°C)	Opt.Max $T_{\mathrm{sec},r}$ (°C)	ΔMax $T_{\mathrm{sec},r}$ (°C)	Thres. (°C)
24-h_highjob	38.290	38.978	0.688	30.0
24-h_lowjob	34.183	37.803	3.620	30.0
36h_highjob	37.272	37.890	0.618	30.0
36h_lowjob	36.593	39.525	2.932	30.0
48h_highjob	38.386	38.838	0.452	30.0
48h_lowjob	37.107	37.573	0.466	30.0

Table A7. Comparison of maximum primary-side return water temperature for *ps_stage_down* under baseline and optimized strategies.

Case	Base. Max $T_{\mathrm{prim},r}$ (°C)	Opt.Max $T_{\mathrm{prim},r}$ (°C)	ΔMax $T_{\mathrm{prim},r}$ (°C)	Thres. (°C)
24-h_highjob	38.778	38.838	0.060	35.0
24-h_lowjob	34.837	35.433	0.596	35.0
36h_highjob	37.926	37.167	-0.759	35.0
36h_lowjob	36.498	37.160	0.662	35.0
48h_highjob	38.820	38.518	-0.302	35.0
48h_lowjob	37.417	37.672	0.255	35.0

Table A6. Comparison of maximum secondary-side return water temperature for *ps_stage_down* under baseline and optimized strategies.

Case	Base. Max $T_{\mathrm{sec},r}$ (°C)	Opt.Max $T_{\mathrm{sec},r}$ (°C)	ΔMax $T_{\mathrm{sec},r}$ (°C)	Thres. (°C)
24-h_highjob	38.290	38.330	0.040	30.0
24-h_lowjob	34.183	35.413	1.230	30.0
36h_highjob	37.272	36.769	-0.503	30.0
36h_lowjob	36.593	37.160	0.567	30.0
48h_highjob	38.386	38.197	-0.189	30.0
48h_lowjob	37.107	37.618	0.511	30.0

Simian Gu received the B.S. degree in Information Science from École Centrale de Pékin, Beihang University, where she is currently pursuing the Master's degree. Her research interest focuses on the use of large language models as optimizers.

Yuxuan Wang received the B.S. degree in Applied Physics from the École Centrale de Pékin, Beihang University, Beijing, China, where he is currently pursuing the Master's degree. His current research interests include AI for Science.

Shunyao Zhu received the B.S. degree in Information Science from the École Centrale de Pékin, Beihang University, Beijing, China, where he is currently pursuing the Master's degree. His current research interests include reasoning large language models.

Lei Yu received his Ph.D. degree in Computer Science from École Centrale Paris in 2008. He also received his Master's degree from the University of Besançon and his Bachelor's degree from Xi'an Jiaotong University. He is currently an Associate Professor at the Sino-French Engineer School, Beihang University. His research interests include data center energy consumption optimization, collaborative perception, and the application of large language models. (Email: yulei@buaa.edu.cn)

Frédéric Magoulès (Fellow BCS, Fellow IMA) is a Professor in the Department of Mathematics and Informatics at Université Paris-Saclay, CentraleSupélec, France. He received the B.Sc. degree in engineering sciences and the M.Sc. degrees in applied mathematics and numerical analysis from Université Pierre et Marie Curie, where he also earned the Ph.D. degree in applied mathematics and the Habilitation à Diriger des Recherches (HDR). He previously held academic positions at Université Pierre et Marie Curie and Université Henri Poincaré before joining École Centrale des Arts et Manufactures (now CentraleSupélec). His research interests include high-performance computing, scientific machine learning, and computational science and engineering Prof. Magoulès has authored or coauthored over 250 refereed publications and several books. He has received multiple international awards, including distinctions from IBM, NVIDIA, and Microsoft, and is a Fellow of several international professional societies.