Toward Data Center Digital Twins via Knowledge-based Model Calibration and Reduction

RUIHANG WANG, DENENG XIA, ZHIWEI CAO, YONGGANG WEN, and RUI TAN, Nanyang Technological University, Singapore
XIN ZHOU, Jiangxi Science and Technology Normal University, China

Computational fluid dynamics (CFD) models have been widely used for prototyping data centers. Evolving them into high-fidelity and real-time digital twins is desirable for online operations of data centers. However, CFD models often have unsatisfactory accuracy and high computation overhead. Manually calibrating the CFD model parameters is tedious and labor-intensive. Existing automatic calibration approaches apply heuristics to search the model configurations. However, each search step requires a long-lasting process of repeatedly solving the CFD model, rendering them impractical especially for complex CFD models. This paper presents *Kalibre*, a knowledge-based neural surrogate approach that calibrates a CFD model by iterating four steps of i) training a neural surrogate model, ii) finding the optimal parameters through neural surrogate retraining, iii) configuring the found parameters back to the CFD model, and iv) validating the CFD model using sensor-measured data. Thus, the parameter search is offloaded to the lightweight neural surrogate. To speed up Kalibre's convergence, we incorporate prior knowledge in training data initialization and surrogate architecture design. With about ten hours computation on a 64-core processor, Kalibre achieves mean absolute errors (MAEs) of 0.57°C and 0.88°C in calibrating the CFD models of two production data halls hosting thousands of servers. To accelerate CFD-based simulation, we further propose *Kalibreduce* that incorporates the energy balance principle to reduce the order of the calibrated CFD model. Evaluation shows the model reduction only introduces 0.1°C to 0.27°C extra errors, while accelerating the CFD-based simulations by thousand times.

CCS Concepts: • **Applied computing** → **Data centers**; • **Computing methodologies** → *Modeling methodologies*; *Neural networks*.

Additional Key Words and Phrases: Data center, computational fluid dynamics, surrogate model, knowledge-based neural network, proper orthogonal decomposition

ACM Reference Format:
Ruihang Wang, Deneng Xia, Zhiwei Cao, Yonggang Wen, Rui Tan, and Xin Zhou. 2022. Toward Data Center Digital Twins via Knowledge-based Model Calibration and Reduction. *ACM Trans. Model. Comput. Simul.* 1, 1 (June 2022), 24 pages. https://doi.org/XXXXXXX.XXXXXXX

A preliminary version of this work appears in The 7th ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys) held virtually in November 2020. This research is supported in part by the National Natural Science Foundation, China, funded under the project (No. 62262026) and the project of Jiangxi Education Department (No. GJJ211111), in part by the National Research Foundation, Singapore, funded under its Energy Research Testbed and Industry Partnership Funding Initiative, part of the Energy Grid (EG) 2.0 programme and its Central Gap Fund (Award No. NRF2020NRF-CG001-027), in part by the Ministry of Education, Singapore, under its Academic Research Fund Tier 1 of RT14/22 and RG96/20, in part by the Nanyang Technologi- cal University, Singapore, under its 8962-Accelerating Creativity and Excellence grant call (No. NTU–ACE2020-01), respectively.
Authors' addresses: Ruihang Wang, ruihang001@ntu.edu.sg; Deneng Xia, xiad0001@ntu.edu.sg; Zhiwei Cao, zhiwei003@ntu.edu.sg; Yonggang Wen, ygwen@ntu.edu.sg; Rui Tan, tanrui@ntu.edu.sg, Nanyang Technological University, School of Computer Science and Engineering, 50 Nanyang Avenue, Singapore, 639798; Xin Zhou, zhouxin@jxstnu.edu.cn, Jiangxi Science and Technology Normal University, School of Information and Mechatronics Engineering, China.

1049-3301/2022/6-ART $15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

The scale of the data center (DC) industry has been rapidly growing in response to the ever-increasing cloud computing and storage demands [3]. Such growth bring substantial challenges for DC operations, particularly in avoiding operational risks and reducing energy costs of the mission-critical infrastructure. Currently, DCs are mostly operated in a reactive way by the data center infrastructure management (DCIM) system with feedback controllers [2, 42]. The DCIM provides the operators with deployed sensor measurements for proper responses in case of abnormalities and failures. However, traditional DCIM does not provide accurate prediction capabilities that are desired by proactive DC management. These capabilities enable the operators to perform various what-if analyses, such as whether the increase of certain temperature setpoints can improve the energy efficiency without causing server overheating.

We consider *predictive digital twins* for the desired capability extension [33]. The computational fluid dynamics (CFD) modeling is a primary technique to characterize the thermodynamics in data halls [27]. CFD model can estimate the air velocity and temperature distributions in a given space by solving the Navier-Stokes (NS) and energy balance equations [4]. It has been adopted in the offline optimization for DC energy cost reduction and thermal risk management [28]. However, the accuracy and simulation speed of CFD models in general do not meet the online analysis requirements for two reasons [30]. First, the assumptions or simplifications made in the offline phase may lead to online result distortions. Second, the solving process of the governing NS equations may need lengthy computing time from hours to days.

The accuracy of a CFD model is mainly determined by the accuracy and completeness of the given boundary conditions. A model with incomplete boundary conditions may diverge from the ground truth. For example, as reported in [34, 41], an uncalibrated CFD model can yield temperature prediction errors up to 5°C. The low accuracy impedes the use of CFD model for the desired operational adjustment to pursue energy efficiency without causing thermal risk. Unfortunately, obtaining the complete and accurate boundary conditions often faces substantial challenges due to 1) the large number of parameters in the boundary spaces and 2) the labor-intensive and error-prone manual calibration process for these parameters. For instance, each server in a data hall may have its own characteristics of the passing-through air flow rate due to its internal fan control logic. Such information is often not available in the server hardware's specification and can only be empirically estimated or manually collected via *in situ* measurement. As a result, the rough settings of the server air flow rates can significantly downgrade the CFD prediction capability. The existing heuristic approaches [10, 29] (e.g., evolution strategies, genetic algorithms, simulated annealing, etc.) can be applied to calibrate these boundary conditions. However, these approaches in general require many search iterations, e.g., hundreds as shown in §5, to find accurate settings for the system configuration parameters. In each iteration, a CFD model solving is performed with the candidate parameters. When the CFD is built for a large-scale data hall with millions of mesh cells, the iterative search process may incur unacceptable computation times. As such, the existing search-based approaches scale poorly with the granularity of the CFD model.

To advance automatic calibration, we propose Kalibre, a neural surrogate-assisted approach to calibrate data hall CFD models with increasing scales and complexities. Kalibre avoids directly solving the CFD model for parameter search with the help of a trainable neural net by iterating four key steps. First, the "coarse" surrogate is trained to align with the "fine" CFD model in the current system state locality by updating its internal weights based on CFD-generated data. Second, the trained surrogate is re-optimized by updating the system configuration, which is also a part of trainable variables of the neural net, to maximize the consistency between the surrogate's predictions and the ground-truth sensor measurements. Third, the updated system configuration

is set back to the CFD model for refining. Finally, the ground-truth sensor measurements are used to validate the refined CFD model. Therefore, Kalibre offloads the fine-grained parameter configuration search to the surrogate. Vis-à-vis the existing heuristic approaches that solve the CFD model every configuration search step, Kalibre solves the CFD model much less frequently for merely providing feedback to the surrogate.

The implementation of Kalibre faces two challenges. First, the training data for the neural surrogate is limited since generating such data using the CFD model is compute-intensive. Second, the design of the surrogate to capture the high-dimensional feature space of a data hall is challenging. Piecemeal solutions to address the above two challenges separately tend to contradictory, i.e., a deeper neural surrogate to better capture the complex feature space may require a large amount of CFD-generated training data. Without proper consideration, the computational cost of training an accurate surrogate might be higher than that of directly calibrating the CFD model. To address the challenges, we incorporate prior knowledge and sensor measurements to adaptively generate training data in each Kalibre's iteration. With this adaptive design, the surrogate update is guided toward searching better configurations in the locality rather than toward ensuring global optimality. Compared with the random training data sampling adopted by the vanilla approach, the adaptive sampling enabled by the introduction of prior knowledge improves the efficiency in using data generated by the CFD simulations. To approximate the temperatures at locations with sensors, we design a knowledge-based neural surrogate to capture the spatial thermal relations that a considered sensor measurement is mostly affected by the settings of the nearby facilities. We implement Kalibre and apply it to calibrate the CFD models of two production data halls sized hundreds of square meters that host thousands of servers. The calibrated CFD models achieve mean absolute errors (MAEs) of 0.57°C and 0.88°C in predicting the temperatures at tens of cold/hot aisle positions in each hall, respectively. In contrast, the heuristic configuration search and the vanilla neural net-based surrogate approach achieve MAEs of around 1.46∼2.2°C with the same computation time for calibration as Kalibre. We also invite a domain expert to manually fine-calibrate the two CFD models, yielding MAEs of 0.98°C and 1.16°C, respectively. As previous research [20] has shown that increasing the air temperature is a common practice to reduce cooling energy, the high prediction accuracy achieved by Kalibre is beneficial for data center energy optimization while ensuring thermal safety constraint. For example, according to the ASHERE standard [5], the server inlet temperature is not allowed to exceed 27°C to prevent overheating. Therefore, an accurate predictive model with can be used to explore a less conservative policy that achieves more energy saving.

Although the calibrated CFD models achieve high-fidelity temperature prediction, the high computation overhead still presents challenges for their online usage. During the online usage, the prediction should be affordable to low-end computing devices with short response time, such that the potential thermal alarms can be properly prevented ahead of time. A possible workaround is to adopt the Kalibre's neural surrogate model for real-time temperature prediction. However, the neural surrogate does not provide a full-fledged temperature field approximation. For example, it is incapable to predict the temperatures at locations without sensors. To address the high computation overhead, we extend Kalibre to Kalibreduce by integrating a model reduction technique developed based on the proper orthogonal decomposition (POD) [11]. The POD method aims to describe a full field profile with a linear combination of a set of spatial basis functions, i.e., the POD modes and corresponding coefficients. While the previous studies have investigated the POD for low-order data hall modelings [25, 32], they assume that the boundary conditions from the CFD are well calibrated. Thus, the POD prediction results are only compared with the original CFD predictions instead of the sensor-measured data. Based on the calibrated CFD models, we further evaluate the reduced POD's performance with sensor data. With the calibrated CFD models, the reduced POD

Table 1. Summary of the existing studies relevant to DC thermal modeling.

Modeling category	Ref.	Hall scale*	Model	Calibration	Reduction	Error (°C)	
						Original	Reduced
White box	[37]	Mid	CFD	–	FFD	–	RMSE: 0.7
	[9]	Small			HRM	–	MAE: <3
	[23]	Mid		Manual	–	Max: 1.9	–
	[7]	Small			Linear	RMSE: 0.7	RMSE: <1
	Ours	**Large**		**Automated**	**POD**	MAE: 0.57 MAE: 0.88	MAE: 0.84 MAE: 0.98
Black box	[19]	Mid	MLP	–	–	MSE: 0.87	–
	[43]	Small	LSTM			RMSE: 1.24	–
Grey box	[17]	Small	ThermoCast	–	–	MSE: 0.1	–

*Small scale: <200 m^2, Mid scale: 200~800 m^2, Large scale: > 800 m^2

models achieve comparable MAEs of 0.84°C and 0.98°C, respectively, while only taking 0.53 and 0.76 seconds in reconstructing the temperature field.

In summary, this paper develops a systematic framework to evolve the data hall CFD models into high-fidelity and real-time digital twins. We incorporate prior knowledge to address the CFD accuracy and speed problems through surrogate-assisted model calibration and POD-based model reduction, respectively. The contributions of this paper are summarized as follows:

- We formulate the model calibration and reduction problems and propose a systematic solution to solve the problems.
- We develop a surrogate-assisted approach that incorporates prior knowledge to solve the model calibration problem. The calibration is implemented with less human effort compared with manual baseline and fewer CFD simulations compared with search-based algorithm.
- Based on the calibrated CFD models, we further reduce the order of the calibrated model using the POD method and energy balance principle to accelerate the simulation speed.
- We conduct extensive evaluations for two industry-grade data halls hosing thousands of servers. The calibrated CFD models achieve MAE of 0.57°C and 0.88°C, respectively. The reduced-order models achieve comparable performance with MAE of 0.84°C and 0.98°C, respectively, while only taking 0.53 and 0.76 seconds in reconstructing the temperature field.

Paper organization: The rest of this paper is structured as follows. §2 reviews the related work. §3 formulates the calibration and reduction problems. §4 presents our proposed approach. §5 evaluates the calibration and reduction performances. §6 discusses several issues. §7 concludes this paper.

2 RELATED WORK

This section reviews the relevant studies in DC thermal modeling, CFD model calibration, CFD model reduction, and knowledge-based methods. Table 1 categorizes the existing thermal models, model calibration, model reduction and their temperature prediction errors, respectively. In what follows, we discuss the details of these existing studies.

■ **DC thermal modeling.** A variety of modeling techniques have been proposed for thermal management in data halls. They can be broadly categorized into white box [7, 9, 23, 37, 40], black box [19, 43] and grey box [17] methods. The CFD models are representative white box models, in that they capture the thermodynamic laws followed by the physical processes. However, the CFD models are computationally expensive due to their internal recursive execution. To reduce computation overhead, the reduced-order models are often used as alternatives. For example, the fast fluid dynamics (FFD) is proposed to accelerate the solving process in [37] and the heat

recirculation matrix (HRM) is fitted to predict server inlet/outlet temperatures in [9]. Another alternative is to use black-box data-driven models to learn a thermal map in the data hall. For example, the Weatherman system [19] predicts the steady-state temperatures of certain server blocks using a neural net consisting of two hidden layers. In [43], a long short-term memory (LSTM) network is designed for predicting server CPU temperature. Although these data-driven models are fast and suitable for real-time prediction, they often perform poorly in the cases that are not covered by the training data. For instance, these models cannot well capture the thermal processes in case of cooling system failures, because the training data for such failure scenarios is generally lacking. Grey box models integrate physical laws and sensor data for temperature forecasting. For instance, in [17], the grey box model named ThermoCast is proposed based on simplified thermodynamics and fitted with historical data. However, such grey box method often relies on specific assumptions of the system dynamics, and may not be transferable to other data halls.

■ **CFD model calibration.** A variety of modeling techniques have been proposed for thermal management in data halls. They can be broadly categorized into white-box [7, 9, 23, 37, 40], black-box [19, 43] and grey-box [17] methods. The CFD models are representative white-box models, in that they capture the thermodynamic laws followed by the physical processes. To ensure fidelity, the CFD models are often manually calibrated by human experts through trial-and-error process. For example, the CFD models in [7, 23] is manually fine-calibrated by a human expert. As such, the manual approach is labor-intensive and only suitable for small-size testbed. The heuristic search methods [10, 29] can be adopted for automatic calibration, but they often require many iterations. As the mesh complexity increases for the modeled data hall, the CFD model's solving time may increase from hours to days. Surrogate-assisted calibration [15] speeds up the parametric search of those compute-intensive and non-differentiable models. It builds a lightweight surrogate of the original model and then uses the surrogate to guide the parameter search. The surrogate design is application-specific [21, 22, 26, 41]. For example, response surface methodology based on radial basis function is studied for CFD model [26]. Among these studies, data-driven surrogates are advantageous in fast forwarding. However, the design of surrogate-assisted optimization faces a general challenge in balancing the surrogate fidelity and the computation overhead of generating training data for surrogate via executing the original compute-intensive model. A possible solution is to improve the local approximation of the data-driven surrogate via proper training data selection. Unfortunately, few studies are dedicated to investigating this in the context of CFD modeling for large-scale data halls.

■ **CFD model reduction.** To accelerate the thermal simulation of a data hall, several approaches have been proposed to reduce the computational complexity of the CFD model. They can be classified into partial-reserved and full-reserved approaches. The partial-reserved approaches simulate the effects of certain parameters on temperature only at certain discrete points, such as the server inlets/outlets [9] or the cold/hot aisles [40]. To maintain the spatial resolution, the full-reserved reductions are desirable for a complete temperature field approximation. The POD-based method is a representative full-reserved approach. It approximates the temperature field with a set of orthogonal base functions and corresponding coefficients. Existing studies [25, 32] have shown that the POD-based methods exhibit good approximation performance for the original CFD models built for data halls in small scale. However, they assume the boundary conditions of the CFD models are calibrated and only evaluate the POD's accuracy with the CFD simulated results. As such, evaluation of POD's performance against real sensor measurements has not been systematically investigated.

■ **Knowledge-based methods.** Knowledge-based modeling incorporates empirical methods or first principles to improve model approximation with less data. For neural nets, the knowledge can be any extra information about the modeled function beyond the function's inputs/outputs used

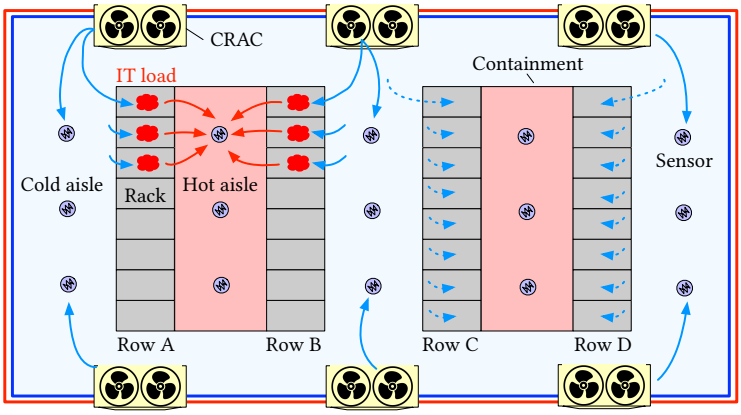

Fig. 1. The layout of a typical data hall. Sensors are installed at the cold and hot aisles for cooling evaluation. Sensor measurements are mostly affected by the nearby CRACs.

as training samples [6]. Several studies have shown that the knowledge-based neural nets exhibit better extrapolation capabilities while requiring less training data, compared with vanilla neural nets. In [35], the neural net is trained by learning a loss function capturing a physical constraint expressed in closed form. This method is also applied in neural surrogate modeling for fluid flows without using any simulator-generated data [36]. For POD-based reduction, the knowledge can be used to develop equations for solving the POD coefficients with new boundary conditions. The knowledge-based methods include the Galerkin projection [25] and the heat flux matching process [32]. In this paper, we will adopt the principle of energy balance to build a linear equation system at the locally specified regions to solve the POD coefficients.

Our prior work [40] proposed the knowledge-based neural surrogate calibration method and evaluated its effectiveness on two CFD models built for industry-grade data halls. In this paper, we further reduce the order of the calibrated CFD models to accelerate their simulation speed and evaluate the performance of the reduced-order models using physical sensor measurements. The reduced-order model is developed based on the POD technique and can be efficiently solved by adopting the energy balance principle of the modeled data hall.

3 BACKGROUND AND PROBLEM FORMULATION

In this section, we first introduce the related background. Then, we formulate the model calibration and reduction problem, respectively.

3.1 Background

Fig. 1 illustrates the layout of a typical data hall, where *racks* hosting *servers* are assigned into multiple *rows* that separate *aisles*. These aisles alternate between cold and hot aisles. The *computer room air conditioning* units (CRACs) supply cold air, denoted by T_{supply}, to the servers through the cold aisles and draw hot air, denoted by T_{return}, from the hot aisles. To reduce air recirculation, containments are often constructed for the hot aisles. To evaluate the thermal condition in a data hall, the inlet and outlet temperatures of servers are often used as the key thermal variables. Therefore, temperature sensors are deployed in the cold and hot aisles to monitor such thermal variables. The inlet temperature of a server, denoted by T_{in} is often required to be in the range of 15°C to 27°C [5]. The outlet temperature, denoted by T_{out}, is related to the heat generated by the server and the passing-through air flow rate. We assume that the energy dissipated from the

Table 2. Summary of notations.

Sym.	Definition	Sym.	Definition
$\|\cdot\|_2$	ℓ_2-norm	$\mathbf{T}_c$	vector of CRAC setpoints
$\langle\cdot,\cdot\rangle$	inner product	$\mathbf{V}$	vector of CRAC volume air flow rates
$\otimes$	element-wise product	$\mathbf{P}$	vector of server powers
l	number of CRACs	α	vector of server volume air flow rates
m	number of servers	$\mathbf{e}$	one-hot vector of cold/hot aisle sensors
n	number of sensors	$\mathbf{T}_s$	vector of sensor measurements
T_{in}, T_{out}	server inlet/outlet temperature	$\widetilde{\mathbf{T}}$	vector of CFD results at sensor locations
T_{supply}, T_{return}	CRAC supply/return temperature	$\widetilde{\mathbf{T}}_s$	vector of full field CFD results
$\mathcal{L}_1$, $\mathcal{L}_2$	loss functions	$\widehat{\mathbf{T}}_s$	vector of neural surrogate predictions
$\boldsymbol{\beta}$	vector of POD coefficients	$\mathbf{W}^{cs}$	adjacency matrix of CRAC to sensor
$\boldsymbol{\Phi}$	vector of POD modes	$\mathbf{W}^{ss}$	adjacency matrix of server to sensor

server in the forms of electromagnetic radiation and mechanical movements is negligible compared with that in the form of heat. Thus, in a steady state, the temperature difference between a server inlet/outlet and a CRAC supply/return can be derived from the energy balance principle [13] as:

$$\begin{aligned} T_{out} - T_{in} &= \frac{P}{c_p \rho \alpha}, && \cdots \text{server side} \\ T_{return} - T_{supply} &= \frac{\sum_i P_i}{c_p \rho V}, && \cdots \text{CRAC side} \end{aligned} \tag{1}$$

where P is the power usage of a server, $\sum_i P_i$ is the total power usage of servers within a CRAC air loop, α is the server passing-through volume flow rate, V is the CRAC supply volume flow rate, c_p and ρ is the heat capacity and density of air, respectively.

The servers in general have different characteristics in passing the cooling air through them. The characteristic highly depends on the server form factor and the control logics of the server's internal fans. Owing to the distinct characteristics, the servers in a data hall often have different passing-through air flow rates. The server air flow rates are part of the CFD boundary configurations that greatly affect the predicted temperature distributions of the data hall. Therefore, to achieve high CFD accuracy, the server air flow rates should be correctly configured before CFD simulation. Unfortunately, they are often unknown and hard to obtain. The current manual in situ measurement using an air volume flow rate meter for each server is labor intensive, especially for a large-scale data hall that hosts many types of servers. As a result, the server air flow rates are often empirically estimated by human expert. For a CFD model with many (e.g., thousands) servers, the rough settings of the server air flow rates could significantly downgrade the temperature prediction capability of the CFD model. The low accuracy will impede the use of CFD model for the desired fine-grained operational adjustment to pursue energy efficiency without causing thermal risk.

In this paper, we focus on devising an automatic approach to calibrate the server air flow rates configuration for data hall CFD models on a steady system state. The approach can be also extended to include other parameters (e.g., by-pass air flow rates and recirculated air flow rates) into calibration. The system state consists of the following measurements: the setpoints and fan speeds of CRAC units, server powers and the temperatures measured in the hot and cold aisles. With the calibrated server air flow rates, the CFD model will yield more accurate temperature prediction results. Although the CFD model can predict the temperature at any location, we focus on the locations that are deployed with temperature sensors and thus have ground-truth temperature measurements for accuracy evaluation. After model calibration, we will further investigate the POD-based model reduction technique to accelerate the CFD simulation speed. The reduced-order

model is expected to achieve real-time temperature prediction without losing spatial resolution and maintain satisfactory temperature prediction accuracy.

3.2 Problem Formulation

To formulate the model calibration and reduction problem, we first define the relevant parameters and variables. Unless particularly specified, the notations used in this paper are summarized in Table 2. We consider a data hall hosting l CRACs, m servers, and n temperature sensors deployed in the cold and hot aisles, respectively.

Definition 1 (Input)**.** The input data for solving a CFD model is a vector consisting of all boundary conditions. Formally, the input $\mathbf{x} = (\mathbf{T}_\mathrm{c}, \mathbf{V}, \mathbf{P}, \alpha)$, where $\mathbf{T}_\mathrm{c} = (T_{\mathrm{c}1}, T_{\mathrm{c}2}, \ldots, T_{\mathrm{c}l})$, $\mathbf{V} = (V_1, V_2, \ldots, V_l)$, $\mathbf{P} = (P_1, P_2, \ldots, P_m)$, and $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ are the vectors of CRAC setpoints, CRAC volume flow rates, server powers, and server air flow rates, respectively.

Definition 2 (Output)**.** The output of CFD is a steady-state temperature distribution field $\widetilde{\mathbf{T}}(\mathbf{x}) = (\tilde{T}_1, \tilde{T}_2, \ldots, \tilde{T}_N)$. For CFD model calibration, we focus on a subset of results within the field at n locations ($n \ll N$) installed with temperature sensors, which is denoted by $\widetilde{\mathbf{T}}_\mathrm{s} = (\tilde{T}_{\mathrm{s}1}, \tilde{T}_{\mathrm{s}2}, \ldots, \tilde{T}_{\mathrm{s}n})$.

Definition 3 (Measurement)**.** The measurement is a vector of real temperature values recorded by the physical sensors, which is denoted by $\mathbf{T}_\mathrm{s} = (T_{\mathrm{s}1}, T_{\mathrm{s}2}, \ldots, T_{\mathrm{s}n})$.

3.2.1 Model calibration problem. Let $\|\cdot\|_2$ denote the ℓ_2-norm of a vector. With the above definitions, the CFD model calibration aims to find the server air flow rate configurations that minimize the ℓ_2-norm of the error vector between the model outputs and the measurements:

$$\begin{aligned} &\alpha^* \triangleq \arg\min_{\alpha} \frac{1}{M} \sum_{i=1}^{M} \left\| \widetilde{\mathbf{T}}_\mathrm{s}(\mathbf{x}_i) - \mathbf{T}_{\mathrm{s}i} \right\|_2^2, \\ &\text{s.t. } \alpha_\mathrm{l} \le \alpha_i \le \alpha_\mathrm{u}, \ i = 1, 2, \ldots, m, \\ &\quad \sum_{i=0}^{m} \alpha_i \le \sum_{j=0}^{l} V_j, \end{aligned} \tag{2}$$

where M is number of the collected samples and α^* is the vector of calibrated air flow rates. Each element in α should be within an empirically estimated range $[\alpha_\mathrm{l}, \alpha_\mathrm{u}]$ and the total volume of server air flow rate should be less than the total CRAC supply. We assume that the servers of the same type have the same volume air flow rate.

We now use an example in a real production data hall to illustrate the discrepancy of an uncalibrated CFD model and the actual sensor measurements. We first show a summary of the working conditions of the data hall. Fig. 2(a) shows a sample distribution of the servers' power consumption at a time instant. Fig. 2(b) is the CRAC setpoints and the corresponding fan speed ratios. Fig. 2(c) shows the temperature values measured by a number of sensors and uncalibrated CFD predictions on the locations of these sensors. For the sensor measurements, the cold aisle temperatures range from 20°C to 24°C, which are related to the CRAC setpoints and fan speed ratios. The hot aisle temperatures range from 30°C to 36°C, which are affected by the generated heat from the servers. The air flow rate of each server is empirically determined for the raw CFD model. With these initial configurations, the CFD model has temperature prediction errors from 2°C to 6°C. Such large errors disqualify the raw CFD model as a high-fidelity digital twin.

3.2.2 Model reduction problem. The model reduction assumes the temperature field can be approximated by H orthogonal basis vectors $\mathbf{\Phi} = (\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_H)$ by $\widetilde{\mathbf{T}}(\mathbf{x}) = \sum_{i=1}^{H} \beta_i(\mathbf{x}) \boldsymbol{\phi}_i$, where $\beta_i, i = 1, 2, \ldots, H$ is the boundary-specific coefficients. Let $\langle \cdot, \cdot \rangle$ denote the inner product of two

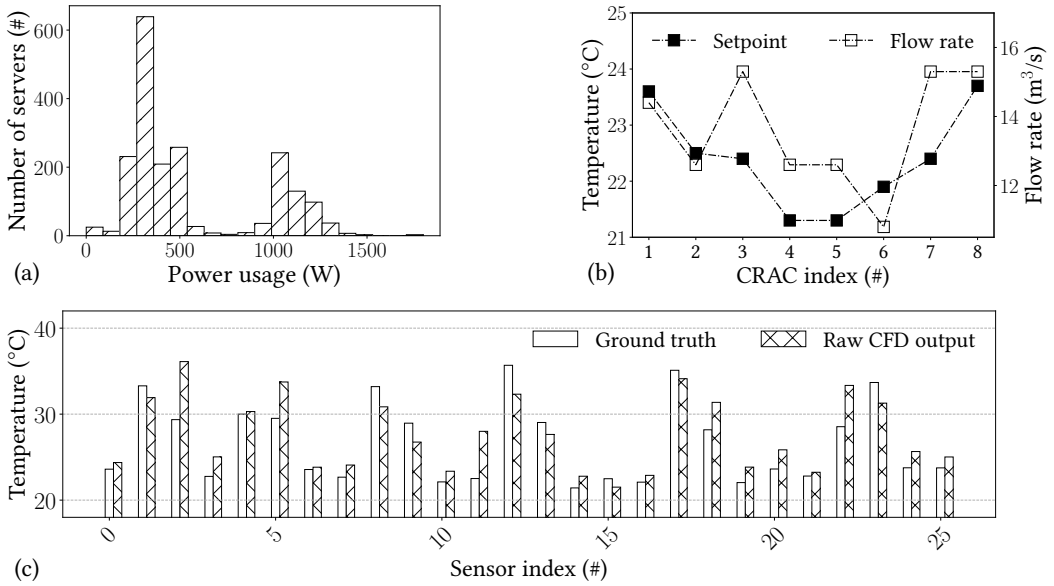

Fig. 2. Collected data from a data hall at a time instant. (a) Server power consumption distribution, (b) CRAC setpoints and flow rates, (c) Sensor measurements and raw CFD temperature outputs at corresponding sensor locations.

vectors. With the above assumption and definitions, the model reduction first aims to find the H orthogonal basis vectors that minimize the ℓ_2-norm of the error between the CFD model output and its projection onto the vector bases as:

$$\begin{aligned} \boldsymbol{\Phi}^* &\triangleq \arg\min_{\boldsymbol{\Phi}} \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} \left\| \widetilde{\mathbf{T}}(\mathbf{x}_i) - \sum_{j=1}^{H} \langle \widetilde{\mathbf{T}}(\mathbf{x}_i), \boldsymbol{\phi}_j \rangle \boldsymbol{\phi}_j \right\|_2^2, \\ \text{s.t. } \boldsymbol{\phi}_i^\intercal \boldsymbol{\phi}_j &= \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}, \ i, j = 1, 2, \ldots, H, \end{aligned} \tag{3}$$

where $\tilde{M}$ is the number of CFD generated samples and $\boldsymbol{\Phi}^*$ is the set of optimal basis vectors. The constraint ensures that the basis vectors are orthogonal and normalized with a unit length of 1. The above problem is equivalent to maximizing the projection of the model output onto the vector bases. Thus, we can write the Lagrange function as: $L(\boldsymbol{\Phi}, \lambda) = \frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} (\left\| \sum_{j=1}^{H} \langle \widetilde{\mathbf{T}}(\mathbf{x}_i), \boldsymbol{\phi}_j \rangle \right\|_2^2) - \sum_{i,j=1}^{H} \lambda_{ij} (\left\| \boldsymbol{\phi}_j \right\|_2^2 - 1)$, where λ is the Lagrange multiplier. Then, the optimal solution of $\boldsymbol{\Phi}$ can be derived by using the Karush-Kuhn-Tucker (KKT) conditions as $\frac{1}{\tilde{M}} \sum_{i=1}^{\tilde{M}} \widetilde{\mathbf{T}}(\mathbf{x}_i) (\widetilde{\mathbf{T}}(\mathbf{x}_i)^\intercal \boldsymbol{\phi}_j) = \lambda_{jj} \boldsymbol{\phi}_j, j = 1, 2 \ldots, H$. The optimization problem is then converted to solving the eigenvectors of the correlation matrix $\mathbf{C} = \frac{1}{\tilde{M}} \widetilde{\mathbf{T}} \widetilde{\mathbf{T}}^\intercal \in \mathbb{R}^{N \times N}$. Once the optimal modes are determined, we need to solve the boundary-specific coefficients to reconstruct the temperature field. We next present the knowledge-based approach to solve the coefficients.

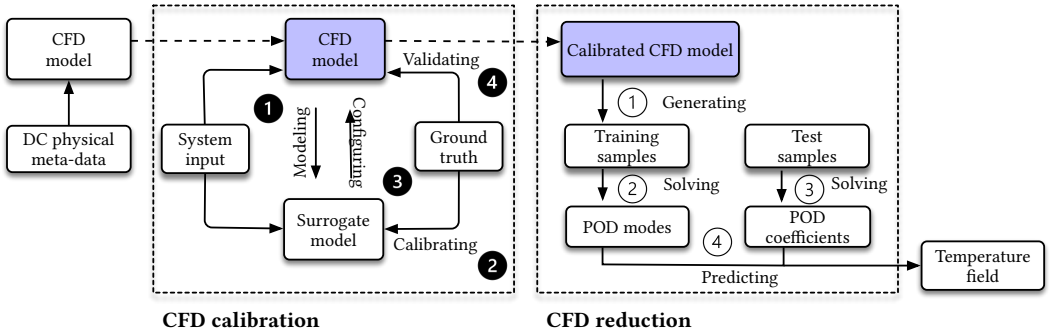

Fig. 3. System workflow that consists of two major blocks of CFD calibration and reduction. We first build the raw CFD model based on the meta-data of a physical DC. Then, we implement CFD model calibration based on real sensor data in four iterative steps. After the CFD model is calibrated, we further reduce the order of the calibrated model to accelerate its computation in four consecutive steps.

4 MODEL CALIBRATION AND REDUCTION APPROACH

In this section, we first present the overview of our approach. Then, we introduce the proposed knowledge-based model calibration and reduction methods, respectively.

4.1 Approach Overview

Fig. 3 illustrates the workflow of our proposed approach that consists of two major blocks of CFD calibration and reduction. We first build the CFD model using the meta-data of the target DC, such as the data hall geometric layout and the facility locations. Then, we conduct CFD model calibration based on real sensor measurements to improve its temperature prediction fidelity. The calibration is implemented in four iterative steps. After CFD calibration, we reduce the order of the calibrated model in four consecutive steps to achieve real-time temperature prediction without losing spatial resolutions. In what follows, we describe the details of the model calibration and reduction, respectively.

■ **Model calibration approach.** Due to the high computational cost of CFD model, directly solving the optimization problem in Eq. (2) using the traditional search-based algorithms incur unacceptable computation overhead. To address this issue, we design a surrogate of the CFD model. Let $\widehat{\mathbf{T}}_s \in \mathbb{R}^{1\times n}$ denote the temperature output vector of the surrogate. Then, the problem in Eq. (2) is converted to a surrogate-assisted optimization that can be solved by iterating four consecutive steps as shown in the CFD calibration of Fig. 3. Specifically, in the first step marked by ❶, the surrogate model is trained to be locally aligned with the CFD model by minimizing the discrepancy between the surrogate's and the CFD's outputs as $\mathbf{W}^* \triangleq \arg\min_{\mathbf{W}} \frac{1}{M}\sum_{i=1}^{M}\left\|\widetilde{\mathbf{T}}_s(\mathbf{x}_i) - \widehat{\mathbf{T}}_{s_i}(\mathbf{W},\mathbf{x}_i)\right\|_2^2$, where $\mathbf{W}$ is a set of trainable weights of the surrogate and $\mathbf{W}^*$ is the result of the surrogate training. In the second step marked by ❷, with $\mathbf{W}^*$, the surrogate is re-optimized through re-training such that the discrepancy between the surrogate's output and the measurement is minimized as $\boldsymbol{\alpha}^* \triangleq \arg\min_{\boldsymbol{\alpha}} \frac{1}{M}\sum_{i=1}^{M}\left\|\widehat{\mathbf{T}}_{si}(\mathbf{W}^*,\mathbf{x}_i) - \mathbf{T}_{si}\right\|_2^2$. In the third step marked by ❸, the finded $\boldsymbol{\alpha}^*$ is configured into the CFD model for a steady-state simulation. In the forth step marked by ❹, the CFD is validated based on physical sensor measurements. If the surrogate approaches to the CFD model, the $\boldsymbol{\alpha}^*$ after the convergence of the four-step iterations will approach to the one given by Eq. (2). We will present the details of the surrogate modeling in §4.2.

■ **Model reduction approach.** After the CFD model is calibrated, it is desirable to reduce its order to accelerate the simulation speed. From §3.2.2, the model reduction first requires to solve the eigenvectors of the correlation matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$ derived from Eq. (3). However, directly solving $\mathbf{C}$ is computationally intensive when the mesh points N of a CFD model is large. For example, N can be larger than 10^7 for a large-scale data hall in §5.1.1. To address this challenge, we adopt the snapshot method [11] to build another correlation matrix $\widetilde{\mathbf{C}} = \frac{1}{\tilde{M}}\widetilde{\mathbf{T}}^\intercal\widetilde{\mathbf{T}} \in \mathbb{R}^{\tilde{M}\times\tilde{M}}$, where $\tilde{M}$ is the number of CFD generated samples, i.e., the snapshots. Then, the POD modes are derived by $\mathbf{\Phi} = \widetilde{\mathbf{T}}\mathbf{\Psi}$, where each column of $\mathbf{\Psi}$ is the eigenvector of $\widetilde{\mathbf{C}}$. Thus, the original $N \times N$ eigenvector problem is reduced to $\tilde{M} \times \tilde{M}$, where $\tilde{M} \ll \tilde{N}$. The model reduction consists of four steps as illustrated in the CFD reduction of Fig. 3. Specifically, in the first step marked by ①, we generate $\tilde{M}$ snapshots by running the CFD model with different boundary conditions. In the second step marked by ②, the POD modes are obtained by solving Eq. (3) using the snapshot method. In the third step marked by ③, with new boundary conditions as input, the POD coefficients are determined based on the energy balance principle. We will present the details of this step in §4.3. In the forth step marked by ④, the temperature field is obtained by the linear combination of the derived POD modes and corresponding coefficients.

4.2 Knowledge-based Neural Surrogate Calibration

4.2.1 Incorporated prior knowledge. As discussed in §1, to address the challenges of the surrogate's complexity versus the needed volume of CFD-generated training data, we incorporate prior knowledge for initial training data selection and surrogate architecture design. We now illustrate three pieces of incorporated knowledge. First, we model the server air flow rate as a linear function of the server power. Specifically, according to [8], the server's internal fan speed is related to the CPU utilization and the inlet temperature. These two factors jointly affect the server power usage. Therefore, we model the server air flow rate as a linear function of the server power usage, i.e., $\alpha = \overline{\alpha}P$, where $\overline{\alpha}$ is a constant in volume per second per Watt (m^3/s/W). Second, we initialize the server air flow rate based on the closest sensor measurements and the energy balance principle to generate the initial training data. Specifically, from Eq. (1), the i^{th} server's air flow rate can be initialized as $\alpha_i = \frac{P_i}{(\Delta T_i+\delta)c_p\rho}$, where δ is a uniform random variable and ΔT_i is approximated by the difference between the closest server sensor measurements at the hot and cold aisles, respectively. Third, we build a knowledge-based neural surrogate that can capture the physical layout and thermal relations among a number of key variables of the considered data hall. Specifically, we model a set of facilites (i.e., CRACs, servers, and sensors) in the considered hall as nodes and their connections as edges into a directed graph. The direction of an edge characterizes the thermal causality between the two end nodes of the edge. For example, an edge points from a CRAC node to a sensor node, because the supply air temperature of the CRAC affects the measured temperature of the sensor. The normalized reciprocal spatial distance between any two facilities will be used as the weight of the edge connecting the corresponding two nodes in the graph. This modeling approach follows the fact that the temperature measured by a sensor is mostly affected by the facilities in its neighborhood [16]. We will present the detailed architecture design in §4.2.2.

4.2.2 Neural surrogate architecture. The neural surrogate aims to approximate the complex thermophysics encompassed in the CFD model. In particular, its efficient training with a small amount of data generated from the CFD model is desirable, since the data generation requires intensive computation. Fig. 4 shows the proposed neural surrogate architecture. It consists of a *cooling block* and a *heating block*. The cooling block models the impact of the CRACs on the temperatures at all sensor locations; the heating block models the impact of the servers on the temperatures at the

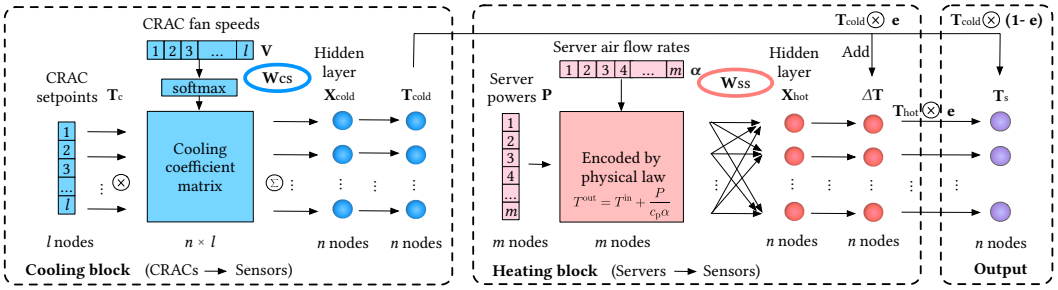

Fig. 4. The architecture of the knowledge-based neural surrogate for temperature prediction. The structure consists of a *cooling block* and a *heating block*. The weight between any two facilities is initialized using their normalized reciprocal spatial distance. Two linear hidden layers are used to predict the cold aisle temperatures and temperature rises induced by servers.

hot aisle sensor locations. Thus, the summation of the two blocks captures the effects from both CRACs and servers. The input of the model consists of the free variables of the data hall's steady state at a time instant, including CRAC temperature setpoints and fan speeds, and server powers. The server air flow rates are designated as trainable variables of the neural surrogate and initialized with rough estimates. Note that, as the neural surrogate is differentiable, the server air flow rates can be updated efficiently by backpropagation-based neural net training algorithms for the purpose of calibration. The output of the neural surrogate is a vector of n predicted temperatures at the sensor locations. In what follows, we present the designs of the cooling and heating blocks of the neural surrogate to capture prior knowledge of the thermal relations among the key variables. Lastly, we present the settings of the constants used by the neural surrogates, which are also based on the prior knowledge on the layout of the modeled data hall.

■ **Cooling block.** This block models the impact of the CRAC temperature setpoints $\mathbf{T}_c$ and fan speeds $\mathbf{V}$ on the temperatures at all sensor locations. First, we encode the two free variables (i.e., $\mathbf{T}_c$ and $\mathbf{V}$) into a hidden-layer variable for the k^{th} sensor as $X_k^{\text{cold}} = \sum_{i=1}^{l} T_{ci} \cdot c_{ik}$, where T_{ci} is the setpoint of the i^{th} CRAC and c_{ik} is a cooling coefficient characterizing the impact of the i^{th} CRAC on the k^{th} sensor. We design c_{ik} to be positively related to the CRAC fan speed. Specifically, we use softmax activation to compute the *cooling coefficient matrix* as $c_{ik} = \frac{e^{z_{ik}}}{\sum_{a=1}^{l} e^{z_{ak}}}$, where z_{ik} is an intermediate variable defined by $z_{ik} = V_i \cdot W_{ik}^{\text{cs}}$, V_i is the fan speed of the i^{th} CRAC, and W_{ik}^{cs} is a weight characterizing the thermal impact of the i^{th} CRAC on the k^{th} sensor. The CRAC-to-sensor matrix $\mathbf{W}^{\text{cs}} \in \mathbb{R}^{n \times l}$ consisting of W_{ik}^{cs} for $i = 1, \ldots, l$ and $k = 1, \ldots, n$ is an adjacency matrix. The weights in this matrix can be fixed or trainable. If they are fixed, their settings are important and will be discussed in §4.2.2. §5 will compare the performance of the neural surrogates with $\mathbf{W}^{\text{cs}}$ fixed or trainable. Lastly, we use a linear layer to project the hidden-layer variable to temperature as $T_k^{\text{cold}} = a_k X_k^{\text{cold}} + b_k$, where a_k and b_k are two trainable weights.

■ **Heating block.** This block models the impact of the servers on the temperatures at the hot aisle sensor locations. Based on this energy balance principle, we use server powers and air flow rates to predict the server-induced temperature increase ΔT_k at the k^{th} hot aisle sensor location. Specifically, $\Delta T_k = c_k X_k^{\text{hot}} + d_k$, where c_k and d_k are two trainable weights, and X_k^{hot} is a hidden-layer variable. The X_k^{hot} is defined by $X_k^{\text{hot}} = \sum_{j=1}^{m} \frac{P_j}{\alpha_j} \cdot W_{jk}^{\text{ss}}$, where P_j is the j^{th} server power, α_j is the j^{th} server air flow rate, and W_{ij}^{ss} is the weight characterizing the thermal impact of the j^{th} server on the k^{th} sensor. We define the server-to-sensor adjacency matrix $\mathbf{W}^{\text{ss}}$ consisting of W_{jk}^{ss} for $j = 1, \ldots, m$

and $k = 1, \ldots, n$. Similar to $\mathbf{W}^{\text{cs}}$, $\mathbf{W}^{\text{ss}}$ can be fixed or trainable. For the former case, its setting is discussed in §4.2.2. Note that the heating block outputs ΔT_k for all sensor locations (denoted by $\Delta\mathbf{T}$); but only the outputs at hot aisle sensor locations will be used when combing the results of the cooling and heating blocks. This design simplifies the vectorized implementation of the neural surrogate using PyTorch (cf. §5.1.4).

■ **Joining two blocks and adjacency matrices settings.** With hot-aisle containment and blanket, heat recirculation is negligible. Thus, the temperatures at cold aisle sensor locations are mainly affected by the CRACs; the temperatures at hot aisle sensor locations are jointly affected by the CRACs and servers. To combine the outputs of the cooling and heating blocks, we define a one-hot vector $\mathbf{e} \in \{0, 1\}^n$, where its element $e_k = 1$ or 0 represents that the k^{th} sensor location is in hot or cold aisle, respectively. Therefore, the final output of the neural surrogate, i.e., the temperatures at all sensor locations, can be expressed by $\widehat{\mathbf{T}}_{\text{s}} = \mathbf{T}_{\text{cold}} \otimes (\mathbf{1} - \mathbf{e}) + \mathbf{T}_{\text{cold}} \otimes \mathbf{e} + \Delta\mathbf{T} \otimes (\mathbf{1} - \mathbf{e})$, where $\otimes$ represents element-wise product, $\mathbf{T}_{\text{cold}} \otimes (\mathbf{1} - \mathbf{e})$ gives the temperatures at the cold aisle sensor locations, and $\mathbf{T}_{\text{cold}} \otimes \mathbf{e} + \Delta\mathbf{T} \otimes (\mathbf{1} - \mathbf{e})$ gives the temperatures at the hot aisle sensor locations.

If $\mathbf{W}^{\text{cs}}$ and $\mathbf{W}^{\text{ss}}$ are fixed, the weights of the neural surrogate are $\mathbf{W} = \{a_k, b_k, c_k, d_k | k = 1, \ldots, n\}$; otherwise, $\mathbf{W}$ additionally include $\mathbf{W}^{\text{cs}}$ and $\mathbf{W}^{\text{ss}}$. Each of their elements represents the thermal impact of a facility (CRAC or server) on a sensor location. Since the thermal impact decreases with spatial distance, in this paper, we set it to be a normalized reciprocal of the spatial distance between the facility and the sensor location. When it is lower than a threshold, it is forced to be zero, indicating that the corresponding thermal impact is negligible. Thus, to set these two matrices, the layout of the data hall and the sensor locations will be needed, which are available to the DC operator in general.

4.2.3 Four-step iterations for CFD calibration. Let $\widehat{T}_{sk}$, $\widetilde{T}_{sk}$, T_{sk} denote the surrogate-predicted temperature, CFD-predicted temperature, and the measured temperature at the location of the k^{th} sensor, respectively. Algorithm 1 shows the pseudocode of the four-step iterations. We now explain it in detail.

Neural surrogate training (Line 4-6): The training data is generated by solving the CFD model with collected system input (including CRAC temperature setpoints and fan speeds, server powers) and the initial $\boldsymbol{\alpha}$ or calibrated $\boldsymbol{\alpha}$ by the previous iterations to yield the predicted temperatures at sensor locations. The detailed training data generation is described in §5.2.1. Note that each element of $\boldsymbol{\alpha}$ should be within $[\alpha_l, \alpha_u]$. The system input, the $\boldsymbol{\alpha}$, and the predicted temperatures form a new training data sample that is added to the training dataset accumulated from the first iteration. With the training dataset, the neural surrogate is updated to minimize the errors between its predicted temperatures and the CFD-predicted temperatures of the training samples. Thus, the weights of the neural surrogate are updated using the gradient of the least squares loss function of $\mathcal{L}_1 = \frac{1}{M}\sum_{i=1}^{M} \left\| (\widehat{\mathbf{T}}_{si}(\mathbf{W}, \mathbf{x}_i) - \widetilde{\mathbf{T}}_{si}(\mathbf{x}_i)) \right\|_2^2$. As a result, the surrogate is trained to align with the CFD model. At the end of this step, $\mathbf{W}$ is frozen.

Surrogate-assisted calibration (Line 7-8): The surrogate is re-optimized to minimize the errors between its predicted temperatures and the measured temperatures by updating $\boldsymbol{\alpha}$. In this step, $\boldsymbol{\alpha}$ is set trainable. An empirical regularization term is added to penalize the loss function if the temperature difference between the hot and cold aisle, i.e., ΔT, is out of the empirical range $[\Delta T_{\text{l}}, \Delta T_{\text{u}}]$. The penalty term is expressed using the rectified linear units (ReLU) as $h(T) = \sum_{j=1}^{m} (\text{ReLU}(\Delta T_{\text{l}} - \Delta T) + \text{ReLU}(\Delta T - \Delta T_{\text{u}})) \times P_j$, where P_j is the j^{th} server power usage. The term means that, if the server power is higher, the penalty should be more significant. Thus, the second loss function is expressed by $\mathcal{L}_2 = \frac{1}{M}\sum_{i=1}^{M} \left\| \widehat{\mathbf{T}}_{si}(\mathbf{W}^*, \mathbf{x}_i) - \mathbf{T}_{si} \right\|_2^2 + \lambda h(\Delta T)$. In our experiments, we set ΔT_{l} and ΔT_{u} to be 5°C and 15°C, based on the operator's experience. To accelerate the

Algorithm 1 CFD model calibration.

Input: Measurements collected from a data hall, including CRAC setpoints $\mathbf{T}_{\text{c}}$, CRAC flow rates $\mathbf{V}$, server powers $\mathbf{P}$, and sensor measurements $\mathbf{T}_{\text{s}}$. Initial server air flow rates $\boldsymbol{\alpha}$. Adjacency matrix $\mathbf{W}^{\text{cs}}$ and $\mathbf{W}^{\text{ss}}$.

Output: Calibrated server air flow rates $\boldsymbol{\alpha}^*$.

1: Initialize α and adjacency matrix;
2: Assign initial configurations to the surrogate graph $\mathcal{G}$;
3: **for** $i = 1$: Max iteration **do**
4: Solve CFD model to obtain $\widetilde{\mathbf{T}}_{\text{s}}$;
5: Aggregate CFD solving results as training data;
6: Train surrogate by performing gradient descent that minimizes $\mathcal{L}_1$;
7: Search $\boldsymbol{\alpha}$ by performing differential evolution that minimizes $\mathcal{L}_2$;
8: Search $\boldsymbol{\alpha}$ by performing gradient descent that minimizes $\mathcal{L}_2$;
9: Configure $\boldsymbol{\alpha}$ to the CFD model;
10: **if** $\frac{1}{n}\sum_i^n |\widetilde{T}_{si} - T_{si}| < \epsilon$ **then**
11: $\epsilon \leftarrow \frac{1}{n}\sum_i^n |\widetilde{T}_{si} - T_{si}|$; $\quad \boldsymbol{\alpha}^* \leftarrow \boldsymbol{\alpha}$;
12: **end if**
13: **end for**

Algorithm 2 CFD model reduction

Input: Calibrated server air flow rates $\boldsymbol{\alpha}^*$ and other boundary conditions, including CRAC setpoints $\mathbf{T}_{\text{c}}$, CRAC flow rates $\mathbf{V}$ and server powers $\mathbf{P}$.

Output: A set of POD modes $\boldsymbol{\Phi}$, vector of coefficients $\boldsymbol{\beta}$ and reconstructed temperature field.

1: Generate training samples by running CFD simulation and calculate their mean average temperature $\widetilde{\mathbf{T}}_{\text{mean}}$;
2: Solve the eigenvector problem of Eq. (3) to derive $\boldsymbol{\Phi}$;
3: Observe new boundary conditions;
4: Construct the linear equation system of Eq. (4) via energy balance principle;
5: Solve the least square of Eq. (4) to derive $\boldsymbol{\beta}$;
6: Reconstruct the temperature field by $\widetilde{\mathbf{T}}_{\text{mean}} + \sum_{i=1}^{H} \beta_i \boldsymbol{\Phi}_i$

re-optimization, we implement a hybrid approach of combining differential evolution algorithm with gradient backpropagation to minimize the loss function $\mathcal{L}_2$. This hybrid approach has been shown effective in accelerating neural net training [39]. We will also evaluate its effectiveness for our specific problem in §5.2.1.

CFD configuration (Line 9): Once the $\boldsymbol{\alpha}$ is recommended by the surrogate, the updated value is configured back to solve the CFD model. The refined boundary conditions are then used for initializing the next calibration iteration.

CFD validation (Line 10-12): For each calibration iteration, the CFD model's accuracy is validated against the ground-truth sensor measurements. During the process, only better $\boldsymbol{\alpha}$ is recorded for final output candidate.

Through the adaptive sampling and iterative optimization of the two loss functions $\mathcal{L}_1$ and $\mathcal{L}_2$, the surrogate will be updated toward finding better $\boldsymbol{\alpha}$ that can improve the CFD model's accuracy. We will evaluate the effectiveness of this approach in §5.2.

4.3 Knowledge-based CFD Model Reduction

4.3.1 Determination of POD coefficients. As presented in §4.1, the POD modes can be derived by solving the eigenvector problem given a set of snapshots. To obtain the temperature field with a new set of boundary conditions, we need to find the corresponding POD coefficients such that the error between the reconstructed temperature and the original CFD output is minimized as $\boldsymbol{\beta}^* \triangleq \arg\min_{\boldsymbol{\beta}} \left\| \sum_{i=1}^{H} \beta_i(\mathbf{x}) \boldsymbol{\Phi}_i^* - \widetilde{\mathbf{T}}(\mathbf{x}) \right\|_2^2$. One approach to solve this problem is to project the derived POD modes onto the CFD governing equations by the Galerkin's method [25] and solve a set of algebraic equations for a steady state. In this paper, we focus on locally specified regions where the energy balance equations can be established, such as the server inlet/outlet and the CRAC supply/return surfaces. For a data hall equipped with multiple CRACs, we consider the global energy balance across these CRACs as $\sum_{k=1}^{l} c_{\mathrm{p}} \rho V_k (T^k_{\mathrm{return}} - T^k_{\mathrm{supply}}) = \sum_{j=1}^{m} P_j$. Then, the equation system in terms of specific boundary conditions is written as:

$$\begin{cases} \sum_{i=1}^{H} \beta_i(\mathbf{x})(\boldsymbol{\Phi}^{i1}_{\mathrm{out}} - \boldsymbol{\Phi}^{i1}_{\mathrm{in}}) = \frac{P_1}{c_{\mathrm{p}} \rho \alpha_1}, \\ \sum_{i=1}^{H} \beta_i(\mathbf{x})(\boldsymbol{\Phi}^{i2}_{\mathrm{out}} - \boldsymbol{\Phi}^{i2}_{\mathrm{in}}) = \frac{P_2}{c_{\mathrm{p}} \rho \alpha_2}, \\ \dots \\ \sum_{i=1}^{H} \beta_i(\mathbf{x})(\boldsymbol{\Phi}^{im}_{\mathrm{out}} - \boldsymbol{\Phi}^{im}_{\mathrm{in}}) = \frac{P_m}{c_{\mathrm{p}} \rho \alpha_m}, \\ \sum_{k=1}^{l} \sum_{i=1}^{H} V_k \beta_i(\mathbf{x})(\boldsymbol{\Phi}^{ik}_{\mathrm{return}} - \boldsymbol{\Phi}^{ik}_{\mathrm{supply}}) = \frac{\sum_{j=1}^{m} P_j}{c_{\mathrm{p}} \rho}, \end{cases} \tag{4}$$

where $\boldsymbol{\Phi}^{im}_{\mathrm{in}}$, $\boldsymbol{\Phi}^{im}_{\mathrm{out}}$ and $\boldsymbol{\Phi}^{ik}_{\mathrm{return}}$ are the values of the i^{th} POD mode close to the center of the m^{th} server inlet/outlet and the k^{th} CRAC return, respectively. If the summation of servers is greater equal than the number of POD modes, i.e., $m \geq H$, the equation system is overdetermined and can be solved using the least-square method. Thus, the computational overhead of solving a CFD model with the governing partial derivative equations is reduced to solving the least-square problem.

4.3.2 Four-step procedures for CFD reduction. After CFD calibration, we use the calibrated server air flow rate $\boldsymbol{\alpha}^*$ as the default boundary conditions to reduce the order of CFD model. Algorithm 2 shows the pseudocode of the CFD model reduction. We now explain it in detail.

Generate training samples (Line 1): We first generate a set of training samples based on the calibrated $\boldsymbol{\alpha}^*$ and other boundary conditions. These boundary conditions can be selected from historical measurements or proper experimental designs. We also calculate the mean average temperature across these training samples.

POD modes solving and selection (Line 2): With the generated CFD samples, we derive the POD modes $\boldsymbol{\Phi}$ by solving the problem in Eq. (3) using the snapshot method. We then select a finite number of derived modes with dominant energy percentage of the temperature field.

POD coefficients solving (Line 3-Line 5): To calculate the temperature field of a new case using the reduced-order model, we solve the corresponding POD coefficients based on the physics-based linear equation system in Eq. (4) by the least-square method.

Temperature prediction (Line 6): With the derived POD modes and the boundary-specific coefficients, the full temperature field can be reconstructed by the their linear combination for different number of selected POD modes. To increase the accuracy of the approximation, we use the summation of the mean average temperature over snapshots and their fluctuating values from the linear combination to reconstruct the temperature field as $\widetilde{\mathbf{T}}_{\mathrm{mean}} + \sum_{i=1}^{H} \beta_i(\mathbf{x}) \boldsymbol{\Phi}_i$.

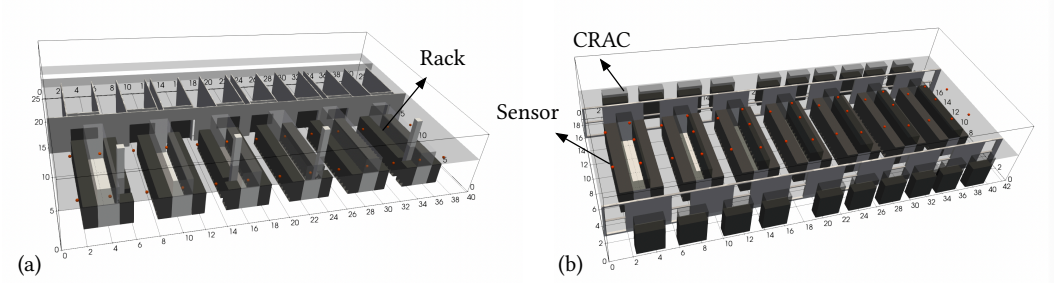

Fig. 5. Geometric layout of the target data halls. The red points are the positions installed with temperature sensors. (a) Hall A; (b) Hall B.

Table 3. Experimental settings.

Parameter	Setting	Parameter	Setting
server flow rate bound $[\alpha_l, \alpha_u]$ (m^3/s)	[0.005, 0.1]	noise bound for initialization $[\delta_l, \delta_u]$ (°C)	[6, 12]
CFD solving iterations	1000	surrogate training epoch	150
cases for calibration	10	cases for test	5
calibration iterations	10	differential search iterations	100
initial learning rate	0.1	crossover rate	0.6
learning rate decay	0.9	cases for POD training	15
population size	10	selected POD modes	10

5 PERFORMANCE EVALUATION

In this section, we apply the proposed Kalibre and Kalibreduce to CFD models built for two production data halls and present their evaluation results.

5.1 Experiment Methodology and Settings

5.1.1 Data halls and CFD models. Our targets are two production data halls (referred to as Hall A and Hall B) as shown in Fig. 5. Hall A is equipped with one-side CRACs and Hall B is equipped with bilateral CRACs. These two halls are in operation for e-commerce applications. Both of them occupy hundreds of square meters and host thousands of servers [1]. Their CFD models were built by a domain expert using OpenFOAM [12]. Specifically, first, the geometry of the data hall and its hosted facilities is built with Salome [31]. Then, OpenFOAM is used to discretize the fluid domain with hexahedral grid for solving. The numbers of the grids of the two CFD models are 7 and 10 million, respectively. After the grids are created, the NS and energy balance equations are solved in OpenFOAM to derive the air flow and temperature distribution, where the air is assumed to be incompressible and the turbulence is modeled using the k-epsilon method [18].

5.1.2 Evaluation metrics and experimental settings. To evaluate the model accuracy, we use the *mean absolute error* (MAE) to measure the differences of temperature prediction at locations installed with sensors. Specifically, $MAE = \frac{1}{M}\frac{1}{N}\sum_{i=1}^{M}\sum_{i=1}^{N}|y_i - \hat{y}_i|$, where N is the number of deployed sensors, M is the number of test cases, y_i and $\hat{y}_i$ are the i^{th} sensor measurement and the prediction made by the CFD, respectively. To evaluate the computation overhead, we compare the simulation time of the original and the reduced-order model on the same machine. Table 3 shows the experimental settings.

[1]The detailed configurations of their hosted facilities are omitted here due to confidentiality requirement.

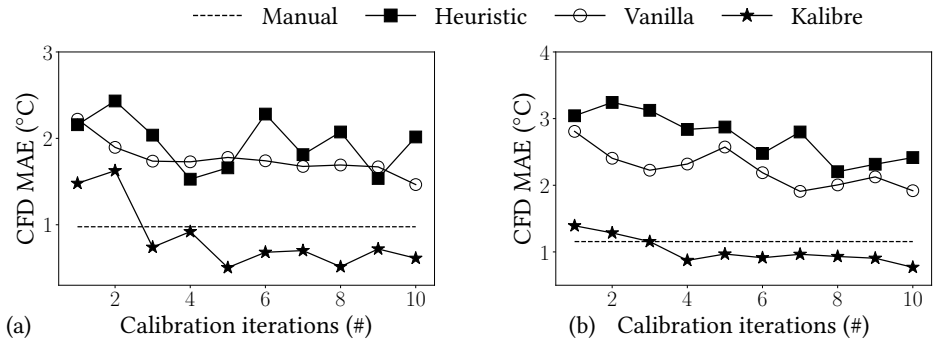

Fig. 6. CFD MAE over calibration iterations with different approaches. (a) Hall A; (b) Hall B.

These settings include the experimental settings, empirical bounds, the surrogate hyperparameters and the POD modes. They are selected based on advice from the domain expert or extensive experimental tests.

5.1.3 Compared approaches. We compare Kalibre in CFD model calibration with three baselines discussed in §1:

■ **Manual calibration** involves extensive tuning of the server air flow rates by a CFD expert with years of experience as conducted by [7]. Specifically, if the CFD-predicted temperature at a sensor location is higher than the ground-truth temperature, the expert empirically increases the flow rates of nearby servers and vice versa.

■ **Heuristic parameter search** uses the covariance matrix adaptation evolution strategy (CMA-ES) to search good $\boldsymbol{\alpha}$. It solves the CFD model every search iteration. CMA-ES is a gradient-free numerical optimization method. It applies the (1+1) strategy to generate one candidate solution per iteration. If the MAE of the new offspring is smaller, it becomes the parent. The mutation rate is set to $\sigma = 5$ and updated for each iteration by following the 1/5 successful evolution rule described in [29].

■ **Vanilla neural surrogate** uses a black-box neural surrogate with three fully-connected layers consisting of 518, 128, and 32 neurons, respectively. It also follows Kalibre's four-step iterations to perform the model calibration. The difference is that it does not incorporate the prior knowledge mentioned in §4.2.1 for initial training data selection.

We also consider two variants of Kalibreduce that incorporates different level of knowledge to reduce the CFD model:

■ **Local reduction** only considers the energy balance on individual server side. It constructs the linear equation system based on the temperature difference between each server's outlet and inlet.

■ **Global reduction** jointly considers the energy balance across multiple CRACs and servers. It constructs the linear equation system using both server-level and CRAC-level knowledge.

5.1.4 Implementation. We implement the model calibration and reduction with Python 3.8 and PyTorch 1.9 [24], where the latter is a library widely used for building machine learning applications. When we use PyTorch to build the neural surrogate's computational graph, the server air flow rates $\boldsymbol{\alpha}$ are set as a vector of trainable variables. This allows us to control their updating by choosing to freeze the gradients or not. We choose Adam [14] as the optimizer, which is a method for efficient stochastic optimization that only requires first-order gradients and little memory space. The linear equation system in Eq. (4) is solved by the least-square solver in SciPy [38]. The OpenFOAM can load the boundary conditions from a configuration file. In our implementation, the Python program

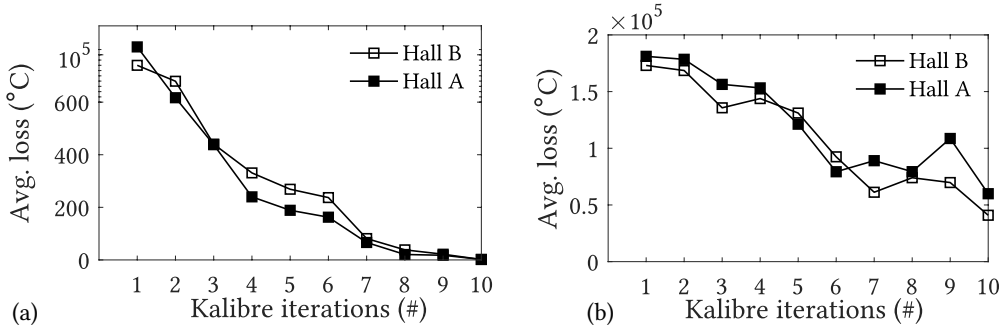

Fig. 7. Average loss over Kalibre's iterations. (a) With differential evolution; (b) Without differential evolution, note that the y-axis scale is not uniform.

writes the boundary conditions to a configuration file, invokes an OpenFOAM session to solve the CFD model, and collects results by parsing the OpenFOAM's output.

5.2 Evaluation of Kalibre

5.2.1 Convergence and effectiveness of Kalibre. In this set of experiments, we evaluate the convergence speed and the effectiveness of different calibration approaches. To initiate Kalibre's four-step iterations, we first solve the CFD model to generate five training data samples by setting each element of $\boldsymbol{\alpha}$ as described in §4.2.1. In the following four-step iterations, a new training data sample is generated by solving the CFD model configured with the latest $\boldsymbol{\alpha}^*$ found by the neural surrogate. This new training data sample is added to the training dataset. In this set of experiment, the calibration terminates after ten CFD simulations. Fig. 6 shows the MAEs of the two CFD models over the ten iterations. We observe that the MAE under Kalibre converges faster than both the heuristic and vanilla approaches. After around three to four iterations, the MAEs of the CFD models are lower than that calibrated by human expert. We also note that the CFD models under Kalibre achieve lower errors at the first calibration iteration. This is benefited from the incorporated knowledge to sample training data at the initialization phase. In the following iterations, Kalibre's surrogate is updated toward finding better configurations of $\boldsymbol{\alpha}^*$ from that initialization.

As presented in §4.2.3, Kalibre adopts a hybrid approach combining the gradient backpropagation widely used for neural net training and the differential evolution to find $\boldsymbol{\alpha}^*$. Fig. 7(a) shows the average loss over the four-step iterations. In the first iteration, the average loss is very large, reaching around 10^5. A closer examination shows that the regularization penalty of the loss function $\mathcal{L}_2$ is large in the very early iterations. However, in the subsequent iterations, the average loss sharply decreases and converges to zero in the tenth iteration. For comparison, we adopt a baseline of using the Adam optimizer only to find $\boldsymbol{\alpha}^*$. Fig. 7(b) shows the results of this baseline. We can see that the convergence is slow and the average loss remains large (about 0.5×10^5) after ten iterations. The results show that the differential evolution effectively accelerates the convergence of Kalibre.

5.2.2 Performance of calibrated model. Then, we show the effectiveness of the model calibration. After ten calibration iterations, we choose the $\boldsymbol{\alpha}$ with the lowest error as default settings to evaluate the CFD models on test cases with different boundary conditions. Figs. 8(a) and 9(a) show the two halls' thermal planes computed based on the uncalibrated CFD models presented for one test case. We can see that the temperature distribution is uneven in both the cold and hot aisles. The uncalibrated CFD models' prediction errors are from 3°C to 6°C as shown in Fig. 2(c). Such large errors are due to the inaccurate estimation of the server air flow rates. Figs. 8(b) and 9 (b) show

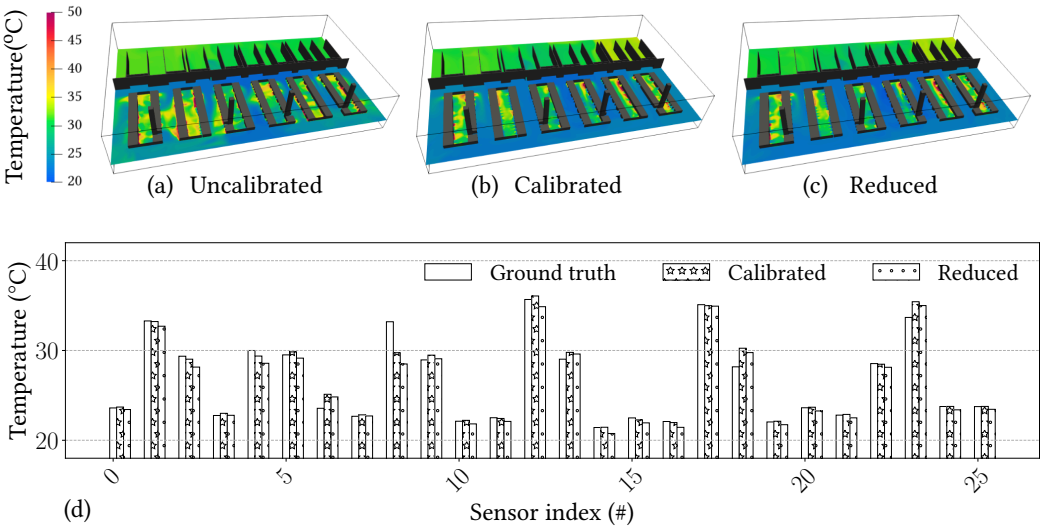

Fig. 8. Hall A temperature distribution. (a) Thermal plane produced by the original CFD model; (b) Thermal plane produced by the calibrated CFD model; (c) Thermal plane produced by reduced-order CFD model; (d) Temperature predictions versus ground-truth temperatures at the sensor locations.

the thermal planes computed based on the CFD models with the calibrated server air flow rates. Figs. 8(d) and 9(d) show the temperatures predicted by the two halls' calibrated CFD models at the sensor locations and the ground-truth values measured by the sensors. We can see that the temperature distributions become more uniform, compared with the results shown in Figs. 8(a) and 9(a). From Figs. 8(d) and 9(d), the temperatures predicted by the calibrated CFD models well match the ground-truth values, with MAEs of 0.51°C and 0.86°C for the two halls, respectively. The above results show the effectiveness of Kalibre for large-scale data halls.

5.2.3 Comparison with baseline approaches. We next compare the performance of Kalibre with other baselines mentioned in §5.1.3. Table 4 shows the MAEs of five independent cases achieved by different approaches after ten calibration iterations. With ten calibration iterations, Kalibre achieves lower MAEs compared with the baseline approaches for both halls. After about 10 calibration iterations, Kalibre achieves MAEs of 0.57°C and 0.88°C for the two halls, respectively. As shown in §5.2.1, the convergence speed of the vanilla neural surrogate is slow without better initialization. It thus requires more calibration iterations to generate more training data samples to well represent the CFD model. Thus, with ten calibration iterations, its calibrated CFDs still yield MAEs higher than manual approach. The heuristic parameter search cannot find good configurations with the same CFD iteration times. Its MAEs remain at high levels of 1.52°C and 2.20°C, respectively. Even after one hundred of CFD model solving processes, their MAEs still saturate at 1.42°C and 1.96°C for the two halls, respectively. Thus, we can see that the heuristic parameter search and vanilla neural surrogate are inefficient to find the optimal configuration under the same computation time. Although the manual calibration reduces the MAE to 0.98°C to 1.16°C, it is labor-intensive. In addition, its MAEs are higher than Kalibre's. The lower MAEs achieved by Kalibre further improves the fidelity of the CFD results. If such results are used to guide the DC operations, the risks caused by the errors can be further reduced. In sum, systematic approaches to improve the fidelity of data hall digital twin are always desirable.

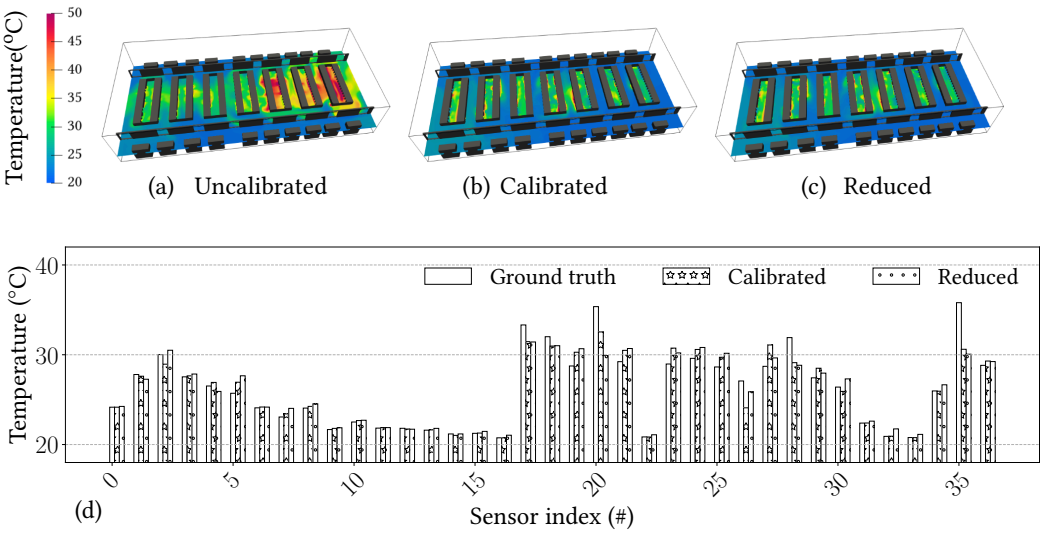

Fig. 9. Hall B temperature distribution. (a) Thermal plane produced by uncalibrated CFD model; (b) Thermal plane produced by calibrated CFD model; (c) Thermal plane produced by reduced-order CFD model; (d) Temperature predictions versus ground-truth temperatures at the sensor locations.

Table 4. MAE achieved over test cases after ten calibration iterations

Data hall	Calibration (°C)				Reduction (°C)	
	Manual	Heuristic	Vanilla	**Kalibre**	**Kalibreduce local**	**Kalibreduce global**
Hall A	0.98	1.52	1.46	**0.57**	0.89	**0.84**
Hall B	1.16	2.20	1.91	**0.88**	1.02	**0.98**

5.2.4 Compute time. Table 5 shows the online inference time for solving the CFD models when the number of used CPU cores varies. Note that the OpenFOAM software package supports parallel computing with multiple CPU cores on the same computer. The evaluation shows that a single CFD model solving takes up to several hours and the solving time decreases with the number of used CPU cores. However, the model solving speed (i.e., the reciprocal of the solving time) is sub-linear to the number of used CPU cores. This suggests that the CFD computation is not completely divisible and the communications among the paralleled units matter. Thus, even if the 64-core limit is lifted, the attempt to use more CPU cores across multiple computers may face performance bottlenecks due to the cross-computer communication overheads. With 64 CPU cores, the CFD model solving time is about one hour. This solving time still renders the heuristic search-based model calibration approaches impractical, since they generally need a large number of iterations (e.g., hundreds as shown shortly). Note that GPU acceleration has been introduced to another commercial CFD software package [1]. However, it brings 3.7x acceleration only, which does not change the impracticality of the heuristic search-based model calibration approaches. From the results in Table 4, Kalibre achieves sub-1°C MAEs with 10 calibration iterations. The compute time breakdown of each iteration is as follows: about 200 seconds for surrogate training, about 100 seconds for configuration search, about one hour for CFD model solving with 64 CPU cores.

Table 5. Average online inference time of the two CFD and reduced-order models with varying CPU cores.

Model		CFD model					Reduced-order model
CPU cores* (#)		4	8	16	32	64	1
Solving	Hall A	77400	38880	25200	10440	3240	0.53
time (s)	Hall B	75960	39240	23400	12960	4320	0.76

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

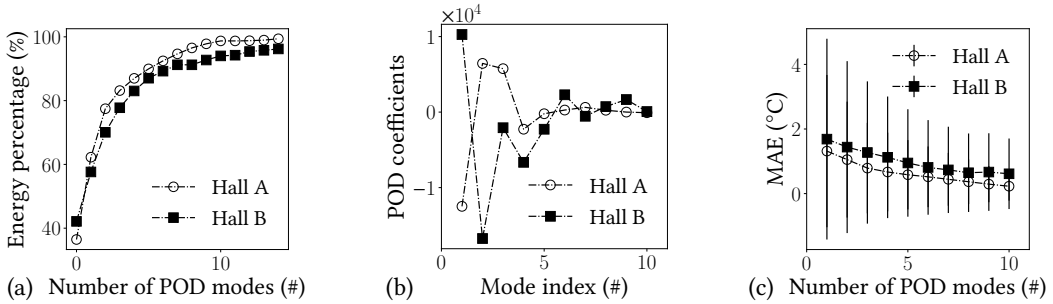

Fig. 10. Impact of POD mode numbers. (a) Cumulative energy percentage; (b) POD coefficients; (c) MAE of the full reconstructed temperature fields. The error bars are the standard deviation.

Kalibre's compute time for calibrating the two hall's CFD models in 10 iterations are about 12 and 9 hours, respectively.

5.3 Evaluation of Kalibreduce

5.3.1 Impact of the POD mode numbers. We first generate the training samples for solving the POD modes. Specifically, we generate 15 samples by running the calibrated CFD models with different boundary conditions from historical input measurements and the calibrated server air flow rates. These samples are then used to calculate the POD modes by solving the eigenvector problem of Eq. (3). The captured energy of each POD mode is proportional to the corresponding eigenvalue. Fig. 10(a) shows the POD captured cumulative energy percentage versus the number of modes for the two data hall models. From the figure, we observe that the first ten modes take 97.8% and 92.7% of the total energy, respectively, indicating the temperature field information is mainly captured by a first few dominant POD modes. In this study, we use the first ten modes to reconstruct the temperature field. Fig. 10(b) shows the POD coefficients $\boldsymbol{\beta}$ versus the mode index for two cases. We observe that the absolute values of the first coefficients are high for both two cases, reaching around 10^4. The values then decrease dramatically when the mode index increases, indicating the first few terms of $\sum_{i=1}^{H} \beta_i(\mathbf{x})\boldsymbol{\Phi}_i$ are dominant in reconstructing the temperature field. We also evaluate the accuracy of the reconstructed temperature field with the full CFD simulation. Fig. 10(c) shows the temperature MAE versus the number of modes for two cases. The error bar represents the standard deviation. From the figure, we observe that the error converges with the increase number of used POD modes. The standard deviation is high with only a few used modes. This is because the reconstruction only focuses on energy balance at local boundaries, i.e., it only aims to satisfy the linear equation system in Eq. (4). Therefore, the errors could be high for regions far away from those boundaries.

5.3.2 Performance of reduced-order model. Next, we show the effectiveness of the model reduction based on the derived POD modes and corresponding coefficients. From Fig. 10(c), with ten used modes, we observe that the reduced-order models achieve MAE of 0.23°C and 0.61°C and standard deviation of 0.45°C and 1.09°C compared with the full CFD simulated temperature field. Figs. 8(c) and 9(c) show the thermal planes reconstructed by the reduced-order models. The thermal planes show close matches to the thermal planes of Figs. 8(b) and 9(b), indicating the POD-based reduction can well approximate the CFD generated temperature fields. Figs. 8(d) and 9(d) show the temperature predictions of the reduced-order models at locations installed with sensors. Table 4 shows the MAEs of the reduced-order models. Compared with the ground-truth values, the predictions with global energy balance produce MAEs of 0.84°C and 0.98°C for the two data halls, respectively. Although the MAEs of the reduced-order models are slightly higher than the calibrated CFD models, the simulation only takes 0.53 and 0.76 seconds, respectively, for reconstructing the temperature fields as shown in Table 5, which is thousand times faster than solving the original CFD models.

6 DISCUSSIONS

We now discuss several worth-noting issues. First, the primary purpose of the Kalibre's surrogate is to improve the efficiency of parameter search. The surrogate does not provide a full-fledged approximation of the CFD model. For instance, it does not model the temperatures at the locations without sensors, which are modeled by the CFD model in contrast. Thus, only the calibrated CFD model or the reduced-order model shall be used as a digital twin for the run-time temperature evaluation of the modeled data hall. Second, the surrogate's architecture described in this paper is for data halls installed with hot-aisle containments and server blanking panels. Thus, heat recirculation is not considered. To address the data halls without hot-aisle containments, heat recirculation and temperature mixing effects should be added to the neural surrogate's design. Third, due to the computation overhead, the training data used to solve the POD modes are generated using boundary conditions from only a subset of historical measurements. To extrapolate the reduced-order model to other cases, it is important to generate extra CFD samples under various boundary configurations. Forth, this paper mainly focuses on temperature prediction. For other types of prediction, the proposed approach can be extended to address their calibration and reduction problems with proper surrogate designs. For instance, if air flow rate sensors are deployed, Kalibre and Kalibreduce can be extended to calibrate the CFD for predicting air velocity distribution.

7 CONCLUSION

This paper first presents Kalibre, an automatic surrogate-based approach to calibrating data hall CFD models. The design of Kalibre's neural surrogate integrates prior knowledge, including the energy balance principle and thermal relations among the key variables of the physical infrastructure. The incorporated knowledge improves the approximation capability of the neural surrogate to the CFD model in the locality for calibration. We demonstrate its effectiveness on two CFD models built for two production data halls that host thousands of servers. The CFD models calibrated by Kalibre achieve temperature prediction MAEs of 0.57°C and 0.88°C, respectively. Compared with manual calibration, Kalibre's improvement of up to 0.3~0.4°C is significant in CFD modeling due to the sharply increased difficulty in improving accuracy when the errors are already low (i.e., at around 1°C). With the calibrated CFD models, this paper further presents Kalibreduce to accelerate the simulation speed of the CFD models. Kalibreduce adopts the POD techniques and energy balance principle to reduce the order of the calibrated CFD models. The reduced CFD models achieve satisfactory MAEs of 0.84°C and 0.98°C, respectively, while accelerating CFD-based simulations for temperature prediction by thousand times. Kalibre and Kalibreduce shed lights in transforming the CFD models built for large-scale data halls into high-fidelity and low-overhead digital twins.

REFERENCES

[1] [n.d.]. GPU-accelerated Ansys Fluent. https://www.nvidia.com/en-sg/data-center/gpu-accelerated-applications/ansys-fluent/

[2] 2019. The evolving data center management maturity model, A Quick Update. https://journal.uptimeinstitute.com/the-evolving-data-center-management-maturity-model-a-quick-update/

[3] 2021. Hyperscale Data Centres Global Market Report 2021: COVID-19 Growth and Change to 2030. https://www.researchandmarkets.com/reports/5446153/

[4] John David Anderson and J Wendt. 1995. *Computational fluid dynamics*. Vol. 206. Springer.

[5] ASHRAE. 2011. 2011 Thermal guidelines for data processing environments–expanded data center classes and usage guidance.

[6] John W Bandler, Radoslaw M Biernacki, Shao Hua Chen, Piotr A Grobelny, and Ronald H Hemmers. 1994. Space mapping technique for electromagnetic optimization. *IEEE Transactions on Microwave Theory and Techniques* 42, 12 (1994), 2536–2544.

[7] Jinzhu Chen, Rui Tan, Yu Wang, Guoliang Xing, Xiaorui Wang, Xiaodong Wang, Bill Punch, and Dirk Colbry. 2012. A high-fidelity temperature distribution forecasting system for data centers. In *2012 IEEE 33rd Real-Time Systems Symposium (RTSS)*. 215–224.

[8] Jinzhu Chen, Rui Tan, Guoliang Xing, and Xiaorui Wang. 2014. PTEC: A system for predictive thermal and energy control in data centers. In *2014 IEEE 35th Real-Time Systems Symposium (RTSS)*. IEEE, 218–227.

[9] Sandeep KS Gupta, Ayan Banerjee, Zahra Abbasi, Georgios Varsamopoulos, Michael Jonas, Joshua Ferguson, Rose Robin Gilbert, and Tridib Mukherjee. 2014. Gdcsim: A simulator for green data center design and analysis. *ACM Transactions on Modeling and Computer Simulation* 24, 1 (2014), 1–27.

[10] John H Holland. 1992. Genetic algorithms. *Scientific American* 267, 1 (1992), 66–73.

[11] Philip Holmes, John L Lumley, Gahl Berkooz, and Clarence W Rowley. 2012. *Turbulence, coherent structures, dynamical systems and symmetry*. Cambridge university press.

[12] Hrvoje Jasak, Aleksandar Jemcov, Zeljko Tukovic, et al. 2007. OpenFOAM: A C++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, Vol. 1000. IUC Dubrovnik Croatia, 1–20.

[13] Michael Jonas, Rose Robin Gilbert, Joshua Ferguson, Georgios Varsamopoulos, and Sandeep KS Gupta. 2012. A transient model for data center thermal prediction. In *2012 International Green Computing Conference (IGCC)*. IEEE, 1–10.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Slawomir Koziel and Leifur Leifsson. 2013. Surrogate-based aerodynamic shape optimization by variable-resolution models. *American Institute of Aeronautics and Astronautics (AIAA) Journal* 51, 1 (2013), 94–106.

[16] Nevena Lazic, Craig Boutilier, Tyler Lu, Eehern Wong, Binz Roy, MK Ryu, and Greg Imwalle. 2018. Data center cooling using model-predictive control. In *Advances in Neural Information Processing Systems*. 3814–3823.

[17] Lei Li, Chieh-Jan Mike Liang, Jie Liu, Suman Nath, Andreas Terzis, and Christos Faloutsos. 2011. Thermocast: A cyber-physical forecasting model for datacenters. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1370–1378.

[18] Bijan Mohammadi and Olivier Pironneau. 1993. Analysis of the k-epsilon turbulence model. (1993).

[19] Justin Moore, Jeffrey S Chase, and Parthasarathy Ranganathan. 2006. Weatherman: Automated, online and predictive thermal mapping and management for data centers. In *2006 IEEE international Conference on Autonomic Computing*.

[20] David Moss and John H Bean. 2009. Energy impact of increased server inlet temperature. *APC white paper* 138 (2009).

[21] Weicong Na, Feng Feng, Chao Zhang, and Qi-Jun Zhang. 2016. A unified automated parametric modeling algorithm using knowledge-based neural network and l_1 optimization. *IEEE Transactions on Microwave Theory and Techniques* 65, 3 (2016), 729–745.

[22] Shreshth Nagpal, Caitlin Mueller, Arfa Aijazi, and Christoph F Reinhart. 2019. A methodology for auto-calibrating urban building energy models using surrogate modeling techniques. *Journal of Building Performance Simulation* 12, 1 (2019), 1–16.

[23] Zachary M Pardey, James W VanGilder, Christopher M Healey, and David W Plamondon. 2015. Creating a calibrated CFD model of a midsize data center. In *International Electronic Packaging Technical Conference and Exhibition*, Vol. 56888. American Society of Mechanical Engineers, V001T09A029.

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems* 32 (2019).

[25] Long Phan and Cheng-Xian Lin. 2017. Reduced order modeling of a data center model with multi-Parameters. *Energy and Buildings* 136 (2017), 86–99.

[26] Long Phan and Cheng-Xian Lin. 2019. CFD-based response surface methodology for rapid thermal simulation and optimal design of data centers. *Advances in Building Energy Research* (2019), 1–23.
[27] Amir Radmehr, Brendan Noll, John Fitzpatrick, and Kailash Karki. 2013. CFD modeling of an existing raised-floor data center. In *29th IEEE Semiconductor Thermal Measurement and Management Symposium*. 39–44.
[28] Yongyi Ran, Han Hu, Xin Zhou, and Yonggang Wen. 2019. DeepEE: Joint optimization of job scheduling and cooling control for data center energy efficiency using deep reinforcement learning. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. 645–655.
[29] Ingo Rechenberg. 1973. Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution, frommann–holzboog.
[30] Matt Renner and Mark Seymour. 2015. Data Center Operational CFD Predictive Models: Are They Accurate Enough to Be Useful and Reliable? *ASHRAE Transactions* 121 (2015), 1K.
[31] Andre Ribes and Christian Caremoli. 2007. Salome platform component model for numerical simulation. In *31st annual international computer software and applications Conference (COMPSAC 2007)*, Vol. 2. IEEE, 553–564.
[32] Emad Samadiani and Yogendra Joshi. 2010. Proper orthogonal decomposition for reduced order thermal modeling of air cooled data centers. *Journal of heat transfer* 132, 7 (2010).
[33] Mike Shafto, Mike Conroy, Rich Doyle, Ed Glaessgen, Chris Kemp, Jacqueline LeMoigne, and Lui Wang. 2012. Modeling, simulation, information technology & processing roadmap. *National Aeronautics and Space Administration* 32, 2012 (2012), 1–38.
[34] Umesh Singh, Amarendra Singh, S Parvez, and Anand Sivasubramaniam. 2010. CFD-based operational thermal efficiency improvement of a production data center. In *SustainIT*.
[35] Russell Stewart and Stefano Ermon. 2017. Label-free supervision of neural networks with physics and domain knowledge. In *31st AAAI Conference on Artificial Intelligence*.
[36] Luning Sun, Han Gao, Shaowu Pan, and Jian-Xun Wang. 2020. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering* 361 (2020), 112732.
[37] Wei Tian, Jim VanGilder, Michael Condor, Xu Han, and Wangda Zuo. 2019. An accurate fast fluid dynamics model for data center applications. In *2019 18th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm)*. IEEE, 1275–1281.
[38] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods* 17, 3 (2020), 261–272.
[39] Lin Wang, Yi Zeng, and Tao Chen. 2015. Back propagation neural network with adaptive differential evolution algorithm for time series forecasting. *Expert Systems with Applications* 42, 2 (2015), 855–863.
[40] Ruihang Wang, Xin Zhou, Linsen Dong, Yonggang Wen, Rui Tan, Li Chen, Guan Wang, and Feng Zeng. 2020. Kalibre: Knowledge-based Neural Surrogate Model Calibration for Data Center Digital Twins. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 200–209.
[41] Paul M Watson and Kuldip C Gupta. 1997. Design and optimization of CPW circuits using EM-ANN models for CPW components. *IEEE Transactions on Microwave Theory and Techniques* 45, 12 (1997), 2515–2523.
[42] Montri Wiboonrat. 2014. Data center infrastructure management WLAN networks for monitoring and controlling systems. In *The International Conference on Information Networking 2014 (ICOIN)*. IEEE, 226–231.
[43] Deliang Yi, Xin Zhou, Yonggang Wen, and Rui Tan. 2019. Toward efficient compute-intensive job allocation for green data centers: A deep reinforcement learning approach. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 634–644.