Learn to chill - Intelligent Chiller Scheduling using Meta-learning and Deep Reinforcement Learning

Praveen Manoharan, Malini Pooni Venkat, Srinarayana Nagarathinam, Arunchandar Vasan
TCS Research, IIT-Madras Research Park, Chennai - 600113, India
srinarayana.nagarathinam@tcs.com

ABSTRACT

Centralized chiller plants with multiple chillers are typically over-provisioned. Therefore, intelligent scheduling is required for the supply (operating chillers) to efficiently meet the demand (actual cooling load of buildings). Traditional cooling-load based control (CLC) may result in poor part-loaded efficiency. Recent data-driven approaches to chiller control either unrealistically assume perfect knowledge of individual chiller power at various leaving chilled water temperatures (LWTs) or control all chillers with same LWT.

We complement existing work with *iChill*, an end-to-end learning-based intelligent chiller power prediction and scheduling strategy. First, given a dataset of chillers of varying capacities, each of which operates at a fixed LWT and varying loads, *iChill meta-learns* a model for power prediction. Specifically, for an unseen target chiller, the meta-learned model is re-trained with known LWT to predict power at unseen LWT. Second, given the configuration of a chiller plant and a cooling load profile, *iChill* learns to schedule individual chillers by jointly deciding the ON/OFF status and LWT; using deep reinforcement learning (DRL).

We train and evaluate *iChill* in a simulated environment with real-world data from a chiller plant of 22 chillers. Specifically, we compare *iChill*'s (1) meta-learned power model with regular transfer learning; and (2) DRL scheduling with multiple baselines including CLC and an oracle model-based predictive control (MPC) strategy with perfect knowledge. We find that *iChill*'s (1) meta-learning improves over transfer learning by up to 15.5%; and (2) DRL scheduling saves 11.5% energy over CLC and is comparable with oracle MPC (12% over CLC). Finally, off-line pre-training of *iChill*'s DRL on the meta-learned chiller models reduces the need for real-world training experimentation by 11x from 3 years to 96 days.

CCS CONCEPTS

• **Theory of computation** → **Markov decision processes**; • **Computing methodologies** → **Simulation evaluation**; • **Applied computing** → **Physics**.

KEYWORDS

meta-learning, deep reinforcement learning, chiller plant, control, simulation

BuildSys'21, November 17–18, 2021, Coimbra, Portugal

ACM ISBN 978-1-4503-9114-6/21/11…$15.00
https://doi.org/10.1145/3486611.3486649

ACM Reference Format:
Praveen Manoharan, Malini Pooni Venkat, Srinarayana Nagarathinam, Arunchandar Vasan. 2021. Learn to chill - Intelligent Chiller Scheduling using Meta-learning and Deep Reinforcement Learning. In *Proceedings of The 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys), Nov 17–18, 2021, Coimbra, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3486611.3486649

1 INTRODUCTION

Problem statement: Campus HVAC systems consolidate and reject heat from multiple buildings in a centralized plant consisting of multiple chillers. A central chiller plant along with pumps typically accounts for about 75% of overall HVAC energy [7] with the rest consumed by air-handling units (AHUs) in buildings. Therefore optimizing the chiller plant power consumption is important. A central chiller plant potentially improves energy efficiency because efficiency of cooling typically improves with increasing capacity of chillers [36]. However, in practice, the offered cooling load of a campus is lower than the design capacity of the chiller plant for almost 99% of operations [1], leading to part-load induced inefficiency. Therefore, the chillers in a centralized chiller plant should be *scheduled intelligently* to dynamically handle the varying cooling load efficiently.

Existing approaches: Intuitively, intelligent chiller scheduling should match the chiller plant's cooling operational capacity (the supply) to the varying consolidated cooling load (the demand) as closely and efficiently as possible. To do this, the supply side control is done by 1) reactive PID-control by measuring system state; or 2) model-based predictive control (MPC) by forecasting the system state evolution using a data-driven or domain-driven model. The control knobs are switching individual chillers of the plant ON or OFF; and changing the leaving chilled water temperature (LWT) of individual chillers. The choice of LWTs determines the load, efficiency, and therefore the power of the individual chillers in a non-linear way [16, 21]; and so that of the chiller plant.

Cooling-load based control (CLC) [17], the most common reactive approach, reactively switches chillers ON and OFF based on the load. Because CLC does not consider individual chiller efficiency for varying loads, it is sub-optimal [14]. Physics-based models [18] for chiller efficiency improve performance, but are very sensitive to calibration errors [8]. More recent data-driven models for scheduling [4] predict chiller efficiency with time-varying cooling loads [29, 39, 40]. These studies, however, either simplistically choose the *same* LWT for *all* chillers in the plant or unrealistically assume that chiller efficiency is readily known at various LWTs for every chiller in the plant. Besides, model-based approaches are very sensitive to model errors [22]. They also require power or performance prediction models for various chiller sub-systems (pumps,

compressors, cooling towers, etc.) and obtaining data for calibration can be cumbersome, particularly for large facilities. Model-based approaches would also require periodic re-calibration of the models to account for chiller mechanical degradation.

LWT as a lever: In practice, the LWT can indeed be set differently at individual chillers as an additional degree of freedom to optimise. As we show later, using individual LWT can save up to 8% additional energy. However, the efficiency of an individual chiller is typically unavailable at various LWTs because the LWT is typically fixed after some initial tuning by the domain expert to ensure ease of operations. Facility managers are also unlikely to experiment with multiple LWT values (even with all chillers having the same LWT) because overall energy consumption may not always decrease with increasing LWT due to increase in secondary-loop pump energy consumption. Besides, a single LWT may not be optimal across all operating chillers particularly when the chillers are of different make, type, and size. In sum, a key requirement of any intelligent chiller scheduling is a model-free approach that can realistically predict and exploit the impact of LWT on chiller power consumption *without assuming a pre-calibrated explicit model* or *extensive experimentation* on existing chillers by facility managers.

Model-free approach: An end-to-end model-free approach for intelligent chiller scheduling should essentially learn to control the system efficiently from minimal data. In doing so, the approach needs to explicitly or implicitly learn to address two sub-problems: (1) prediction of power consumption of individual chillers at various LWTs; and (2) a model-free controller that uses the power predictions to schedule the chillers. Reinforcement learning (RL) is a model-free approach to control that has proven to be promising in various domains [24, 33]. However, using RL directly for chiller scheduling with LWT control (i.e., implicitly learning power prediction; and explicitly learning to control) would require extensive experimentation during the exploratory training, which is not feasible in practice. Therefore, we need a model-free approach with minimal real-world experimentation that can explicitly learn both power prediction and control.

Our approach: We address this gap with *iChill* – a model-free intelligent chiller scheduling approach. For the power prediction problem at unseen LWT, *iChill* uses a meta-learning technique. Specifically, *iChill* meta-learns (M^2L) over normal operating conditions of chillers of varying capacities to estimate the power consumption at unseen LWT (Figure 1). Let $\mathcal{M}(\theta_j)$ denote a machine-learned model for the power consumption of a chiller C^j with the typically known static features F_S^j (design capacity, COP, and the fixed LWT, T_W) and the dynamic feature F_D^j (actual load). Because our goal is to optimise over varying T_W, we need a model $\mathcal{M}(\theta)$ that, given F_S and F_D, predicts the power for a non-design T_W for any chiller with limited retraining. Commonly used machine learning (ML) usually trains for data-points for one chiller and tests for unseen data-points of the same chiller or at least the same type of chillers. Meta-learning generalises this across chillers of different capacities and types, and so minimizes the need for experimentation to train with reasonable accuracy in practice as we show in our work.

For **model-free control**, *iChill* uses deep reinforcement learning (DRL) [30] that has gained acceptance across multiple domains [24, 33]. DRL directly interacts with the environment even in the absence of any model to learn a near-optimal decision policy based on a received reward. In *iChill* (Figure 2), the scheduler is the DRL agent that takes control actions (chillers' ON/OFF status and LWT set-points) to minimise the energy consumption (reward) by interacting with the environment (building/cooling load profile). To minimize experimentation, the DRL agent leverages *iChill*'s M^2L model for extensive off-line learning. Post the off-line training, the DRL agent can learn on a real building with minimal re-training. We demonstrate in our experimental evaluation that M^2L cuts down the on-line training requirement significantly. Using DRL for model-free demand-side control of AHUs has received attention [30]. However, to the best of our knowledge, the use of meta-learning for chiller power prediction and chiller plant control using DRL has not been addressed in the literature. We complement existing literature with the following **specific contributions:**

- We demonstrate an M^2L technique to learn a target chiller's power consumption at unseen LWTs; using EnergyPlus's chiller performance dataset [9] that includes 96 chillers across a wide range of rated efficiencies; capacities; and LWTs. We compare M^2L with regular transfer learning (TL). Specifically, we study the effects of (1) a limited number of chillers for meta-learning (by considering subsets of the 96 chillers); (2) (temporally) sparse data for every chiller in the dataset (a few months instead of the entire year); and (3) similarity of target chiller with training chillers in terms of rated COP and rated capacity.
- We implement the DRL of *iChill* and evaluate it on a real-world campus served by a 22-chiller plant on a one-year cooling load profile. We evaluate *iChill* (M^2L+DRL) using the following baselines: (BL1) traditional CLC that switches on just-enough number of efficient chillers for each cooling load requirement without modulating the LWT; (BL2) an oracle MPC chiller scheduling that assumes perfect knowledge of the cooling load profile with both ON/OFF and LWT as decision variables; (BL3) same as BL2 but with a fixed LWT; (BL4) MPC that leverages M^2L for estimating the chiller power consumption at various LWTs in the prediction horizon; (BL5) *iChill*'s vanilla DRL without M^2L that trains on-line from scratch; and (BL6) *iChill* without online re-training. BL4 omits DRL and BL5 omits M^2L and thus they constitute an ablation study of *iChill*'s sub-components.

Our **key findings** include the following:

- M^2L of *iChill* generalises well and outperforms TL in all the cases considered. The average error for M^2L is in the range of 4.5% to 6.9% and for TL is in the range of 19% to 22%.
- Varying LWT can additionally save 8% of energy.
- *iChill* reduces training time significantly from 3 years to 96 days and the performance in terms of annual energy savings is close to that of the Oracle MPC (10% in comparison to 12% of BL2). These show that *iChill* is a real-world, scalable chiller scheduler that not only has very short real-world experimentation requirements but also gives close to ideal savings practically achievable.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 presents the mathematical formulation of the problem. Section 4 presents the solution strategy with algorithms. Section 5 describes the experimental setup. The results are discussed in Section 6. Finally, limitations and next steps are discussed in Section 7 followed by the conclusion in Section 8.

2 RELATED WORK

Related work can be broadly categorized into 1) Chiller modelling; 2) Chiller scheduling; and 3) Learning-based approaches that control the HVAC system for energy efficiency in buildings

Chiller Modelling: Data-driven techniques have become popular for modelling the performance of chillers. Recently, multi-task learning has been proposed in [40] to develop a separate model for chillers and showed improved performance over a one-for-all model. A clustered approach was used in [39] to develop a COP prediction model for each given chiller given a cooling load profile. In [29], a regression-based approach to model the performance of individual components of a chiller plant (condenser pump model, chilled water pump model, cooling tower model) was proposed. In [26, 31], a neural network-based on-line algorithm to predict chiller power was proposed.

Chiller Scheduling: Most works in this space predict the cooling load profile, which is further used in scheduling. [26] used deep-learning techniques for cooling load prediction and optimal chiller sequencing. [14] proposed an MPC method using the predicted cooling load and wet-bulb temperature profiles to control the chiller staging with condenser water temperature as the decision variable. An optimum load sharing strategy for chillers was developed in [35] that maximised plant COP. A time-constrained chiller sequencing method using a data-driven COP prediction model and joint priority ordering was proposed in [39]. [25] proposed a robust chiller loading strategy by accounting for the uncertainty in the cooling demand. [20] presented a stochastic chiller sequencing control that determined proper thresholds for switching ON/OFF of chillers in CLC-based control. [3] improved the overall chiller plant performance using a rule-based scheduling of appropriate air- and water-cooled chillers, conditioned on the load and outside conditions. This study, however, did not consider changing LWT.

Learning-based approaches: Recently, the classical problems in buildings such as energy prediction and HVAC control have been revisited using advanced ML-based techniques. Transfer learning has been successfully applied to model building thermal properties such as air temperature and humidity in [6, 19]. In [12, 13], data from similar buildings are used to predict the energy consumption of a new building. Both model-free and model-based DRL approaches have become popular for HVAC control problems. A multi-zone HVAC control was studied in [30]. Radiant heating system control was considered in [38]. Model-based DRL techniques for HVAC control were considered in [5, 37].

Research gaps and our focus: Existing chiller performance techniques consider either a fixed LWT or assume sufficient data is available across a wide range of LWTs. To the best of our knowledge, these techniques may not generalise when the data is inadequate. We leverage meta-learning to generalise the chiller power prediction. Meta-learning is largely unexplored for buildings in general and chillers, in particular. Further, most DRL methods for HVAC control in buildings focussed on the demand-side equipment, that is, modulating the comfort air temperature set-point or flowrate to minimise the energy and thermal discomfort. The optimal control of supply-side equipment (chiller plant and secondary pumps) using model-free techniques has received less attention. We use a multi-agent DRL technique to the optimal chiller scheduling problem. Note that, while DRL is model-free, it can still benefit from a model, where available, to train off-line that may help in reducing the interactions with the environment on-line. To the best of our understanding, the benefit of off-line training over training on-line from scratch is unreported in the previous studies. We quantify this benefit in the current work.

Table 1: Notation used

Symbol	Meaning
Control process	
t, Δt	Time-index and control time-step
τ	number of steps in prediction horizon
$\mathbf{T}^{\mathbf{j}}_{\mathbf{W}}$	Vector of LWT for chiller j in $[t,t+\tau\Delta t]$
$\mathbf{C}^{\mathbf{j}}$	Vector of ON/OFF status for chiller j in $[t,t+\tau\Delta t]$
E^T	Total energy consumed by chiller plant
E^j	Energy consumed by chiller j
E^P	Energy consumed by secondary pumps
Q^j, Q^j_{rated}	Cooling load and rated capacity of chiller j
$T_{\min}$, $T_{\max}$	Minimum and maximum LWT
T_R	Plant return water temperature
Q_L	Total cooling load offered to the plant
N	Number of chillers
Meta-learning	
Φ_A, Φ_B	Repository of train and target chillers
$\mathcal{M}$	Local learner (neural network)
$\mathcal{M}^2$	Meta (global) learner (neural network)
θ	Parameters of meta-learner
θ_i	Parameters of local learner i
α, β	Learning rates for meta and local learners
F^j_S, F^j_D	Static and dynamic data of chiller j
DRL	
S, $\mathbf{A}$	Global state vector, Joint action space
$N^i_{\text{ch,avail}}$	Number of chillers available in cluster i
C, $\mathcal{N}$	Number of chiller clusters, number of DRL agents
nEpisodes	Number of episodes
nEpochs	Number of epochs (time-steps per episode)
$\epsilon_{\min}$, r	Minimum ϵ value, global reward
γ, Λ	Discount factor and reply buffer
K	Target network update frequency
Q, $\bar{Q}$	Action-value and target networks

3 PROBLEM FORMULATION

Background: A centralised chiller plant (supply-side) serves multiple buildings (demand-side) that require cooling. The plant uses multiple chillers of varying capacity and performance. The chiller plant supplies chilled water to the buildings through a water distribution network. The cold water picks up heat from the air in the buildings in heat-exchangers and returns as warm water to the chiller plant, where it is again cooled in a repeating cycle. A higher cooling load in the buildings (typically, due to higher ambient temperature) reflects as a higher plant return water temperature. With increasing return water temperature, more chillers will be turned on by the controller (PID) to maintain the same LWT. All chillers in the plant typically use the same LWT that is fixed at the design stage. This is cooling load-based control (CLC). Because the size of the increase in the supply is in the steps of the chiller capacities being turned on, CLC, though easy to implement, can be sub-optimal due to lower part-load ratios in individual chillers [14].

3.1 Chiller power prediction problem

Using varying LWT for individual chillers gives an extra degree of freedom to load the chillers more efficiently. Because the outputs of the chillers mix at the plant-level, the demand-side will still get chilled water at one uniform temperature that is, however, generated at better efficiencies in individual chillers. For example, to handle more load, a currently ON chiller can be loaded more to a lower chilled water temperature; or a newly turned ON chiller can be loaded at a higher chilled water temperature. To use this insight, we need to estimate the efficiencies of chillers at varying LWT. The key challenge is that this data is not available as only one LWT is typically used. Using the design data curves for varying LWT is unrealistic as the chiller performance invariably degrades over its lifetime of 15 to 20 years, leading to increased consumption of up to 25% [10]. Finally, there could be a trade-off between chiller plant energy and secondary pump energy if we vary the LWT at a plant level. We address the chiller efficiency estimation problem in Section 4.1 by *meta-learning over available operational data.*

3.2 Chiller scheduling problem

Objective function: Notation used is summarised notation from Table 1. Consider a chiller plant with N chillers of varying capacities. Our goal is to stage M chillers ($M \leq N$) with appropriate LWTs such that the total cooling energy (chiller plant + secondary pumps) is minimised subject to meeting the total cooling load of the buildings over the control horizon. Mathematically,

$$\min_{\mathbf{C},\mathbf{T_W}} \quad \sum_{t}^{t+\tau\cdot\Delta t} E_t^T, \tag{1}$$

where $\mathbf{C}$ and $\mathbf{T_W}$ are the vectors of chillers' ON-OFF status and LWT over the prediction horizon τ. After every time-step Δt, the optimization problem is solved again in a receding horizon manner. Here E_t^T is the total energy consumed by the chillers and pumps over $[t, t+\Delta t]$:

$$E_t^T = \left[\sum_{j=1}^{N}\left(C_t^j \cdot E_t^j\right)\right] + E_t^P, \tag{2}$$

where E_t^j is the energy consumed by chiller j and E_t^P the secondary pumps' energy over $[t, t+\Delta t]$. For a given plant-level return water temperature, deciding the LWT T_W^j of chiller j fixes its cooling load Q^j and energy consumption E^j.

Main constraints: At every control-step, the aggregate offered cooling load across all the buildings should be met across all the chillers scheduled at their specific LWTs. Specifically,

$$\sum_{j=1}^{N}\left[C^j \cdot Q^j\right] \geq Q_L, \tag{3}$$

and the cooling load of each chiller is less than or equal to its rated capacity, and the LWT is within an acceptable range,

$$\forall j, Q^j \quad \leq \quad Q^j_{\text{rated}}, \tag{4}$$

$$\forall j, T_W^j \quad \in \quad [T_{\min}, T_{\max}]\,. \tag{5}$$

Because our focus is on a fine-grained supply-side control, the demand-side comfort requirements are implicitly captured at an aggregate level. Modelling AHU-level comfort makes the problem computationally infeasible. However, we measure and report unmet cooling load hours (which is an indicator of occupant discomfort) in our experimental evaluation. Besides, in this study, we choose a control time-step of 1-hour. From our experience, the time-taken to ramp up chillers to full-load from low part-loads; the water distribution network inertia; and minimum ON (OFF) time requirements of chillers are usually within 10 to 30 minutes. Hence, for a control step of 1 hour, these effects become less important and are not accounted for.

Challenges: To solve the formulated problem, a few challenges need to be addressed. First, using a model-predictive control (MPC) strategy offers improved performance only if the model error is low [8]. Equations 1–5 assume a model for the chiller power consumption at different LWTs. Second, Equations 1–5 form a mixed-integer non-linear programming (MINLP) problem due to ON/OFF binary decisions coupled with non-linear continuous variables and functions; and therefore are hard to solve. Indeed, our evaluation of MPC with MINLP did not scale beyond a few chillers and hence we use a heuristic to address the issue. The model requirement for the chiller can be addressed by our meta-learning approach. For solving the scheduling problem at scale, we use *a multi-agent deep reinforcement learning-based approach* as described in Section 4.2.

4 SOLUTION STRATEGY

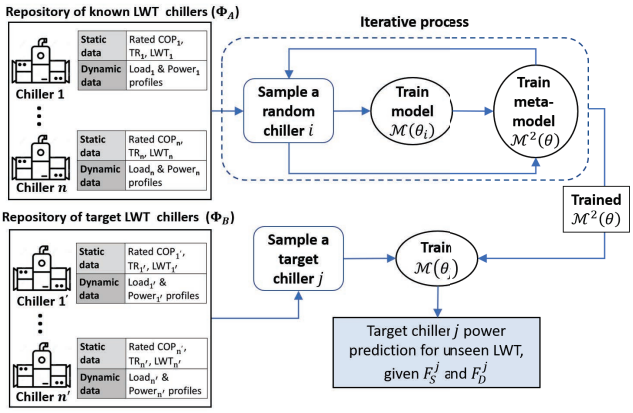

Figure 1: Meta-learning process (M^2L).

4.1 Meta-learning for chiller power prediction

Transfer learning builds a model for a known chiller of a specific type for known values of LWT and tries to generalise it to unknown values of LWT. Given our goal is to compute the efficiency of an arbitrary chiller at an arbitrary LWT, regular transfer learning does not suffice because 1) For a known chiller type, training data covers only **one** LWT; and 2) For an unseen chiller type (target chiller), there is no training data. Meta-learning [11] can help generalise known LWT for a few chiller types to unseen LWT of both known or unseen chiller types, while regular transfer learning can generalise for only known chiller types. The accuracy would depend upon the learning task and training data at hand. We adapt a state-of-the-art meta-learning algorithm for the problem of chiller efficiency estimation at unknown LWT. Specifically, we leverage the few-shot meta-learning approach proposed in [11]. Among the variants of

[11], we use the serial version that is known to be faster for nearly the same performance [23].

Overview: Figure 1 presents an overview of meta-learning for chillers. Chillers are grouped into two sets: Φ_A - training chillers with known power consumption with each chiller at a fixed LWT; and Φ_B - chillers for which the power consumption needs to be predicted at unseen LWTs. There are two types of learners: (a) local learners $\mathcal{M}(\theta_i)$, one for each chiller i in Φ_A; and (b) a meta-learner $\mathcal{M}^2(\theta)$ that learns to generalise to chillers in Φ_B. All $\mathcal{M}(\theta_i)$ and $\mathcal{M}^2(\theta)$ are neural networks with identical architecture.

The output of $\mathcal{M}(\theta_i)$ is the power of a chiller i at a given LWT. The inputs can be viewed as (1) static data: the rated COP, rated TR (tonnage), and LWT; and (2) dynamic data: the cooling load. Across all training examples indexed by t for a chiller i, note that Q^i_{rated}, $\text{COP}^{\text{i}}_{\text{rated}}$, and the LWT T^i_W are fixed. The actual cooling load Q^i_t and the actual power consumption P^i_t alone vary with t. During the training, first, a chiller i is sampled at random from Φ_A and $\mathcal{M}(\theta_i)$ is trained like in any learning. Specifically, $\mathcal{M}(\theta_i)$ is trained over a few iterations and in each iteration, a few points are drawn at random from chiller i dataset. Second, using the loss $L[\mathcal{M}(\theta_i)]$, the meta-learner is trained. The first and second steps are repeated for a few iterations. Third, the trained meta-learner is used in training the learner $\mathcal{M}(\theta_j)$ of a target chiller j from Φ_B. Finally, $\mathcal{M}(\theta_j)$ is used in predicting the power consumption of chiller j at unseen LWTs, given the static rated parameters (rated capacity and COP), F^j_S, and dynamic load, F^j_D.

Details: Algorithm 1 shows the details of meta-learning. Φ_A denotes the set of chillers from different locations with varying rated COPs and rated TRs. A chiller in Φ_A has training examples with one fixed LWT at various loads. However, across chillers of different or even the same type in Φ_A, the LWT varies. The meta-learner $\mathcal{M}^2(\theta)$ is initialised in Line 4 and updated at Line 14 for every iteration in the outer loop (Lines 5-15). For every outer loop iteration, a chiller is drawn at random from Φ_A and the local learner's parameters θ_i are initialised with the updated meta-learner's parameters θ (Line 7). Next, for every chiller i, the local learner $\mathcal{M}(\theta_i)$ is trained for a few iterations in Lines 8-12. Finally, meta-learner is updated in Line 14 using the loss that is estimated with the local learner $\mathcal{M}(\theta_i)$ over another D random examples from chiller i dataset. Therefore, in Line 14, the gradient of the loss function (in terms of θ_i) concerning the meta-learner θ is well defined. For developing a prediction model $\mathcal{M}(\theta_j)$ at unseen LWTs for a target chiller j in Φ_B, the meta-learner is iteratively adapted to j by sampling D examples from the available dataset in Lines 19–20. The available dataset for j consists of the power consumption examples at a fixed LWT and the post-retraining objective is to predict the power consumption at other unseen LWTs.

Algorithm 1: Meta-learning using multiple chillers.

Hyper-parameters:

1 α, β, β' // Learning rates of meta-learner and local learners

2 $ITERS_{outer}, ITERS_{inner}, ITERS_{retrain}$ // Number of updates for the meta-learner, local learner, re-training

Inputs:

3 Φ_A, Φ_B // Repository of training and target chillers

4 Randomly initialize meta-learner parameters θ

5 **for** $d = 1$ to $ITERS_{outer}$ **do**

6 Sample a chiller i from Φ_A

7 $\theta_i \leftarrow \theta$ // Initialize local learner with meta-learner

8 **for** $k = 1$ to $ITERS_{inner}$ **do**

9 Sample D examples from i

10 Evaluate gradients, $\nabla_{\theta_i} L[\mathcal{M}(\theta_i)]$ on D

11 Update learner, $\theta_i \leftarrow \theta_i - \beta \cdot \nabla_{\theta_i} L[\mathcal{M}(\theta_i)]$

12 **end for**

13 Sample another D examples from chiller i

14 Update meta-learner $\theta \leftarrow \theta - \alpha \cdot \nabla_{\theta} L[\mathcal{M}(\theta_i)]$

15 **end for**

16 **for each** j in Φ_B **do**

17 $\theta_j \leftarrow \theta$ // Initialize learner with meta-learner

18 **for** $k = 1$ to $ITERS_{retrain}$ **do**

19 Sample D examples from target chiller j

20 Update local learner $\theta_j \leftarrow \theta_j - \beta' \cdot \nabla_{\theta_j} L[\mathcal{M}(\theta_j)]$

21 **end for**

22 Store $\mathcal{M}(\theta_j)$

23 **end for**

4.2 Reinforcement Learning for scheduling

We present a model-free DRL approach to learn a scheduling policy that jointly decides the ON/OFF status and LWT for chillers. In RL, the agent examines a state of the system and learns to decide the optimal action for that state to maximize a reward it earns [28]. A naive approach for chiller scheduling would be to use one RL agent that decides ON/OFF and LWT for all chillers. The explosion in the action-space with increasing number of chillers makes this computationally infeasible [30]. Another approach is to decouple the multiple systems and obtain the optimal policy for a sub-system in isolation, and use transfer learning to scale the optimal policy to other similar systems with minimal or even no training [22]. However, in our setting, multiple chillers act together to meet the total cooling load and hence a chiller cannot be trained in isolation because local optimality may not imply joint optimality.

Scalable RL: We address scalability by clustering chillers. Specifically, each cluster is constrained to use the same LWT but each chiller in the cluster may independently be ON/OFF. For C clusters, we use multi-agent reinforcement learning using C+1 agents (neural networks). A main agent A_M decides the clusters to be staged; and an agent A_i for each cluster that decides the LWT for cluster C_i. The decisions of the C+1 agents (chiller clusters with their respective LWTs) are jointly implemented on the environment.

Multi-agent learning: The agents A_M and $\{A_i\}$ learn in a multi-agent cooperative setting [22, 27], where the rewards for all agents are identical, that is, $r^1 = r^2 ... = r^{C+1}$. All the agents select their respective actions based on the ϵ-greedy approach. Because A_M can turn on any subset of clusters, its action space has the size 2^C. Each agent A_i has 6 possible actions (the number of discrete steps for the LWT changes, 5–10°C in steps of 1°C). The size of joint actions of the set of $\mathbf{A}$ is thus $A^1 \times A^2 ... \times A^{C+1}$. Once the clusters are chosen and the LWT is fixed by the agent, the number of chillers within the cluster are chosen to match the load. Specifically, the environment may not require all chillers in the chosen cluster,

instead, the load is greedily allocated to the chillers in the chosen clusters in the decreasing order of their efficiencies at that LWT; till the total cooling load is met by the chosen chillers. Note that we account for the energy corresponding to dead load (minimum PLR) of the unused chillers in the cluster picked. We find that this helps in guiding the RL pick the right number of clusters, given a state.

State-space: We use the same global state vector S_t for all the agents at time t given by,

$$S_t = \left\{Q_L, t, T_R, N^1_{\text{ch,avail}} \ldots N^C_{\text{ch,avail}}\right\}_t, \tag{6}$$

where Q_L is the total cooling load presented to the plant, t the time-of-the-day, T_R the return water temperature, and $N^j_{\text{ch,avail}}$ the number of available or unused chillers to schedule in cluster j at time t.

Action-space: The joint action space of the RL is given by,

$$\mathbf{A}_t = \left\{[ON/OFF]^1 \ldots [ON/OFF]^C, T^1_W \ldots T^C_W\right\}_t, \tag{7}$$

Reward: The global reward is engineered in such a way to minimise the total plant power while meeting the cooling load. The instantaneous reward at time t+1 is given by,

$$r_{t+1} = \begin{cases} 1 - \frac{E^T_{t+1}}{E^{T_{\text{rated}}}}, & \text{if } \sum_j \{C^j \cdot Q^j\} \geq Q_L \\ \text{Penalty}, & \text{Otherwise} \end{cases} \tag{8}$$

where $E^{T_{\text{rated}}}$ is the rated energy of the chiller plant. Note that, by definition (Equation 2), E^T accounts for the operating chillers compressor energy and pumps energy. Further, a chiller's compressor energy is function of its load and LWT. The reward is normalised to the scale [0,1] when the cooling load presented to the chiller plant is met and is a penalty otherwise. The value of the penalty in Equation 8 was tuned to -10.

Algorithm 2: Chiller scheduling algorithm.

```
Initialise constants/hyper-parameters
for every agent N = C + 1, Initialise: replay buffer Λ;
  action-value network Q; target network Q̄
for every episode do
    Reset building environment and Linearly decay ϵ
    for every epoch do
        Implement the actions and, observe tuples of next
        state and rewards
        Store the tuples in replay buffer
        Select a mini-batch from the buffer for training
        Calculate targets and train the agents
        Using ϵ-greedy technique, the main agent A_M
        samples chiller clusters and each cluster agent A_i
        selects LWT
    end for
    The action-value network is copied to target network
    every K episodes
end for
```

RL implementation: We implement the DRL algorithm using a variant of the deep Q-network (DQN) [22, 30] in a multi-agent setting. Algorithm 2 presents the pseudo-code of our implementation.

4.3 Realising *iChill* in practice

Figure 2 presents the details of realising *iChill* framework in practice. Because RL can benefit from approximate models, we first train *iChill* off-line using chiller power models developed using the meta-learner (M^2L). We use EnergyPlus [9] to simulate the environment for the RL agent. In theory, we can build the entire facility

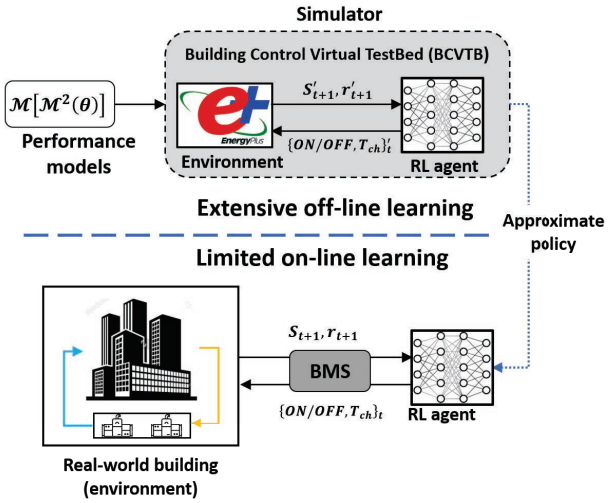

Figure 2: *iChill* framework.

in EnergyPlus by accounting for the building envelope properties, HVAC designs, etc. However, in practice, this can be cumbersome. Instead, we simplify the process by using the historical logs of the plant cooling load profile as input to the EnergyPlus. Further, the chiller power models (obtained using M^2L) serve as chiller performance curve inputs to the EnergyPlus. The RL agent acts on the EnergyPlus model through the co-simulation framework, building control virtual test-bed (BCVTB) [32], and receives an instantaneous reward (power consumption) for its actions while the state moves to the next cooling load sample. In addition to penalising the reward for any cooling load not met, we add the deficit cooling load to the next sample. This is done to mimic the dynamic behaviour of discomfort since the cooling load profile is given as an input to EnergyPlus. Once the RL agents have trained sufficiently on the simulator and learned an approximate policy, we use this to train the agent on-line through limited interactions with the real-world environment. As we will show later, *iChill* benefits from M^2L and requires less number of interactions with the real-world environment to learn a near-optimal chiller scheduling policy. During deployment, the trained DRL agent exchanges information with the building management system (BMS) to act on the chiller plant and for receiving the chiller energy consumption and cooling load.

5 EXPERIMENT SETUP

5.1 Chiller power prediction

We use the chiller performance curve repository of EnergyPlus [9]. The repository has 96 chillers with capacities in the range 134 TR to 1607 TR. Out of these, we randomly pick 12 target chillers of varying capacities for testing (Φ_B). The remaining 84 chillers (Φ_A) are used in training and are distributed equally across six climatic regions (14 chillers per region). We generate training data using

the Φ_A chillers as follows. We leverage the large office building model from the National Renewable Energy Laboratory (NREL) reference building archetypes. We obtain the cooling load profile with a frequency of 1 hour from a whole-year simulation of this office building across the six climatic regions. Each cooling load profile so generated is rescaled to each chiller in Φ_A such that the peak load of the profile is 0.9x of the chiller capacity (as is the design norm). For each of the 14×6 =84 combinations of chiller and cooling profile, we choose one LWT at random in the range of 5, 6, ..., 10°C; and collect training data-points by fixing the LWT for the entire year of the form (Rated capacity, Rated COP, LWT, Cooling load, Power). The choice of LWT depends on factors such as occupant thermal comfort and equipment safety. To this end, we chose the LWT range of 5-10°C that complies with ASHRAE's recommendation [2]. We aim to optimise LWT within this range.

Meta-learning details: A deep neural network (DNN) with 3 hidden layers is used. The network is randomly initialised with Xavier uniform and uses ReLU activation function for the hidden layers. The LWT, cooling load, and the rated values are the input features to the DNN. The DNN is trained using Algorithm 1. The parameters used for meta-learning training and retraining are given in Table 2.

Table 2: Hyper-parameters

Hyper-parameters	Outer Loop Training (M^2L)	Inner Loop Training (M^2L)	Training (TL)	Retraining (M^2L and TL)
Iterations	5000	5	5000	50
Learning rate	0.09	0.005	0.005	0.005
Minibatch size	1280	1280	1280	1280
Optimizer	SGD	Adam	Adam	Adam

Baseline (TL) details: As a baseline for the efficacy of meta-learning, we use transfer learning (TL). Specifically, we use the 84 chillers in the training set to pre-train a randomly initialized neural network. As is the case for regular TL, this pre-trained model is then fine-tuned for each target chiller using the target chiller's training data to develop the chiller models. We believe the design of TL in our work is a form of the "fine-tuned" domain adaptation generalisation technique [34]. We have tested various combinations of keeping low-level/high-level/all layers frozen and have presented the best results. The hyper-parameters for the baseline TL are also given in Table 2. For a fair comparison, TL and *iChill*'s M^2L use the same number of data samples for training with identical neural network architecture and hyper-parameters.

Additional experiments: In addition to TL, we use these scenarios to evaluate the robustness of meta-learning:

- **Varying number of training chillers:** Out of the 84 chillers available in the training set, different number of chillers are picked at random and used for training. Specifically, models were trained using 12 and 48 chillers in Φ_A.
- **Limited data availability for training chiller:** To simulate a chiller not being used in a season, instead of the whole year, each chiller has a reduced dataset spanning four to eight months of equivalent data.
- **Similarity of target chiller with training chillers:** For a given target chiller j, the training chillers are chosen to be within 25% of the target chiller's rated capacity and COP. Both TL and M^2L use this clustered train set Φ_A^j.

5.2 Chiller scheduling

We consider a real-world commercial campus in hot-humid climatic conditions that is cooled by a centralised chiller plant consisting of 22 chillers. The chiller plant also has a few backup chillers that are not considered in the present work. The rated specifications of the 22 chillers are shown in Table 3. We obtained a 4-year cooling load

Table 3: Chiller plant design parameters

Chiller index	WC 1	WC 2	WC 3	WC 4
Count	6	5	6	5
Rated capacity (TR)	532	1481	1413	389
Rated flowrate (GPM)	1276	3555	3391	934
Rated COP	6.5	6.9	7.1	7.4

profile at hourly granularity (through BTU meters) and the hourly energy consumption profile of all the chiller from building management system (BMS). All the chillers are water-cooled (WC) and operated at a LWT of 5 °C. In this case, the chillers naturally form four clusters as shown in Table 3. We consider the manufacturer's power curves as the ground-truth.

We compare *iChill* (off-line and on-line training, Figure 2) with the following baselines.

BL1. **CLC:** The chillers are ranked based on their power consumption and the most (least) efficient chillers are staged ON (OFF) in sequence depending on the cooling load. The LWT is kept fixed at 5°C. BL1 reflects current practice.

BL2. **Oracle MPC:** The optimal control problem (Equations 1–5) is solved in a receding horizon setting assuming perfect knowledge of the cooling load profile and chiller power consumption at various LWTs. BL2 serves as an upper bound on the performance of *any* practically realizable control.

BL3. **Oracle MPC with fixed LWT:** This is same as BL2 but with a fixed LWT of 5°C. Only the chiller staging ON/OFF is used as decision variable. This baseline emphasises the need to vary LWT along with staging (BL2) to obtain additional savings.

BL4. **Approximate MPC:** This is same as BL2 except that instead of the perfect knowledge of the chiller power consumption information, we use the approximate chiller power models developed using the meta-learning technique. BL4 reflects limitations of MPC in practice when the actual system deviates with age or usage from the calibrated model.

BL5. **Fully on-line *iChill*:** Here, there is no off-line training. Instead, *iChill* learns the optimal control from scratch by directly interacting with the environment. BL5 helps us quantify how much on-line training can be avoided by using off-line training. The DRL algorithm is trained for three years and tested on one-year cooling profile.

BL6. ***iChill* trained off-line:** This involves training *iChill*'s DRL off-line using approximate chiller power models (M^2L). The approximate control policy learned off-line is exploited online without additional exploration, that is, without re-training.

DRL details: The state-space features are normalised to the range [0,1] using standard *min-max* scaling [30]. We use five agents: the main agent A_M for deciding the staging (ON/OFF status) of the

clusters; and four cluster-level agents $A_1 \cdots A_4$ for deciding the LWTs of each cluster. All the agents have the same network architecture: an input layer (seven nodes); and two hidden layers (20 nodes). For A_M, the output layer has 16 nodes (each representing one of 2^4 cluster choice combination). For A_i, the output layer has 6 nodes (5 to 10 ^{o}C in steps of 1 ^{o}C). We use Rectified Linear Unit (*ReLU*) as the activation function for the hidden layers and linear function for the output Q-value estimation. The weights and biases of the networks are initialized using *Xavier uniform initializer*. The hyper-parameters for the DRL are listed in Table 4.

Table 4: DRL hyper-parameters

Parameter	Value	Parameter	Value
(nEpochs, nEpisodes)	(7 days, 156)	mini-batch size	128
optimizer	RMSProp	$\epsilon_{\min}$	0.1
Learning rate	0.005	Λ^1,...,Λ^5	2000
γ	0.9	K	5

The *iChill* framework is trained off-line initially with M^2L and interacts with the environment on-line for 96 days. The baseline for fully on-line training (BL5) trains on-line for $3 \times 365 = 1095$ days. For testing, *iChill* is evaluated on an *unseen* one-year load profile. The control-step is taken as 1 hour. Choosing 1-hour control-step precludes the need to micro-model transient phenomena such as chiller physical and mechanical disturbances due to frequent chiller cycling [15, 26]; transition time to full load from low part-loads; and inertia in the water distribution network. Further, we considered a campus with a primary-secondary chilled water distribution system, that is, all the chillers are of constant flowrate type. A variable primary system may offer flowrate as an additional control knob.

Performance metrics: (a) For the power prediction problem, we use the mean absolute percentage error (MAPE) from the expected power. This is obtained from the ground truth of the manufacturer's power curves in the E+ repository. (b) For the chiller scheduling problem, we use the percentage of annual energy savings from the baseline BL1 and percentage of unmet cooling load hours over the testing year. In this case, we use the manufacturer's power curves of real-world chillers as ground truth.

6 RESULTS

6.1 Chiller power prediction

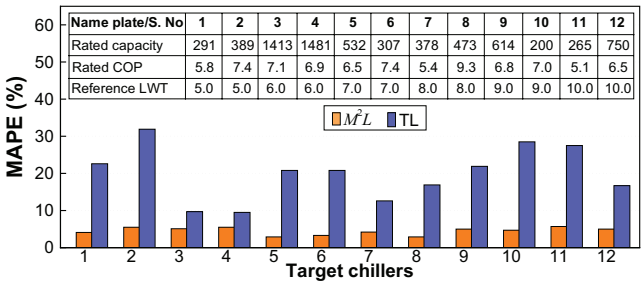

Figure 3: Comparison of power prediction error between M^2L and TL.

Figure 3 shows the MAPE in the power prediction for M^2L and TL for 12 target chillers in Φ_B at unseen LWTs in the range $5-10^oC$. The X-axis shows the chiller ID, and the Y-axis, the average error across the unseen LWT values in the chiller. The rated values and reference LWT (used for retraining the target chillers) are indicated in Figure 3. We observe that M^2L generalises well with an average error of 4.5% ± 0.3% (mean ± standard error), while TL gives an average error of 20% ± 2%.

Fewer training chillers: Instead of Φ_A=84, we experimented with two scenarios: Φ_A=12 (two chillers picked at random from each of the six climatic regions); and Φ_A=48 (eight chillers at random from each of the six climatic regions). All six climatic regions are used to maintain diversity in the LWTs. We find that: (a) for $\Phi_A = 12$, M^2L (TL) has an error of 6% ± 0.6% (21% ± 2%); and (b) for $\Phi_A = 48$, M^2L (TL) has an error of 4.6% ± 0.5% (19% ± 3%). We note that the error falls with increasing Φ_A for both, with M^2L consistently generalising better than TL.

Fewer samples for every training chillers: For the prediction errors between M^2L and TL with all training chillers having samples over a season of four to eight months instead of the whole year, M^2L (5% ± 0.5%) outperforms TL (22% ± 3%) for all the target chillers even with fewer training samples.

Training with only similar chillers: For every target chiller, training chillers are selectively chosen to be within ±25% of the capacity and the COP of the target chiller. This results in a different Φ_A for every target chiller, but with each target chiller's model being trained on only chillers of its type. In general, M^2L (6.9% ± 1.4%) still outperforms TL (21.8% ± 4.4%) in most cases. However, we found that when there are enough training chillers similar to a target chiller for a wide range of LWT, the performances of M^2L and TL are comparable (prediction error is ~ 4.5%). Further, both M^2L and TL perform poorly when the training set has only a few chillers and is also sparse in the LWT.

Note that we have assumed a diverse set of LWT across chillers under one climatic region, which may not always be available. To understand effect of limited LWT being available, we have designed the following two experiments: 1) For a target chiller, we trained a power model using chillers from the same climatic region as the target chiller with a few different LWT in the train set. This case represents reduced diversity in both chillers and LWT in the train set. M^2L gave an average error of 7.5% across the target chillers, while the TL gave a 20% error; 2) All train chillers within a climatic region use the same LWT. However, LWT varies across climatic regions to have some diversity. Here, there is reduced diversity in LWT alone and the train set consists of chillers from all climatic regions. M^2L gave an average error of 6%, while TL gave an 18% error. Even in these two experiments, M^2L outperforms TL. The slight increase in the errors for M^2L is expected because of the decreased diversity in LWT (and/or chillers). For brevity, we omit the figures for the additional experimentation cases.

6.2 Chiller scheduling

Reasons for saving energy: We explain the out-performance of *iChill*'s DRL using Figure 4. The X-axis represents the time in weeks. The grey coloured line in Figure 4 is the cooling load offered to the chiller plant. This is the main driver for the entire system. The primary Y-axis represents the average LWT in ^{o}C across the plant (the average of individual chillers' LWT weighted by the design flowrate). The secondary Y-axis shows the number of active chillers

at any point - both raw and smoothed. We make the following observations. First, DRL learns to stage the chillers closely in sync with the offered cooling load. Second, when chillers are staged, DRL aims to load an existing chiller to the lowest LWT to ensure the part-loading is minimized; this explains why DRL touches the lower end of the LWT spectrum. Finally, we find that DRL maintains the (smoothed) number of active chillers consistently lower than BL1. Due to these factors, DRL achieves energy savings.

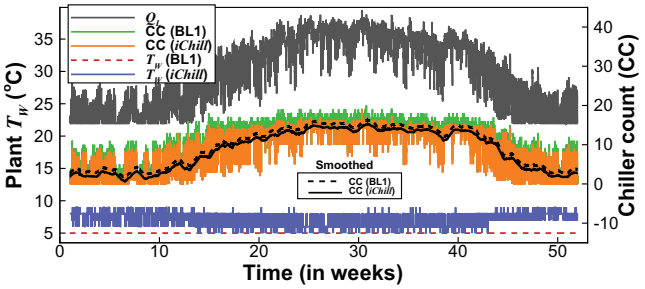

Figure 4: Control strategy of *iChill* and BL1.

Baseline comparison: Figure 5 quantifies the extent of *iChill*'s annual energy savings and unmet cooling hours by comparing with the baselines. BL1 represents the as-is CLC control. All savings reported in Figure 5 are reported as percentages relative to BL1; therefore we do not show BL1 explicitly. The X-axis shows the control strategy. The primary (secondary) Y-axis shows the percentage of annual energy saved (percentage hours of unmet cooling load) over the test year. The percentage hours of unmet cooling load is an indication of the end-user discomfort due to the scheduling.

The oracle MPC (BL2) performs the best with 12% over BL1 while meeting the cooling load offered at all times. Because BL2 has perfect knowledge of the cooling load and power models of the chillers; it shows the best performance for *any* control under idealized settings. Next, BL3 is the same as BL2 but with the effects

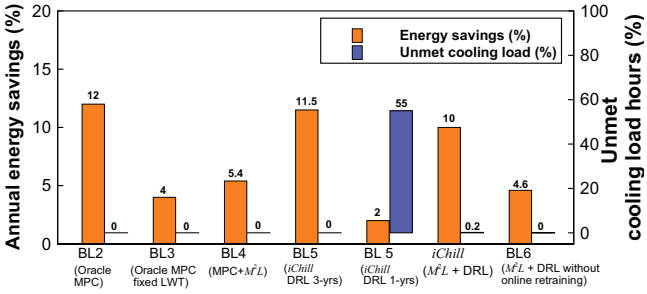

Figure 5: *iChill* vs baselines. All savings are reported as improvements over baseline BL1 (not shown).

of varying LWT removed. We find that the energy savings drop by 8% from BL2, which indicated that having LWT as an additional control knob helps.

The approximate MPC (BL4) uses the chiller power models obtained with M^2L. BL4 improves only 5.4% over BL1 in terms of the energy while meeting the cooling load at all times. This is because MPC performance is reduced by the inherent prediction error of 4-5% in the M^2L chiller model. These findings are consistent with [22]

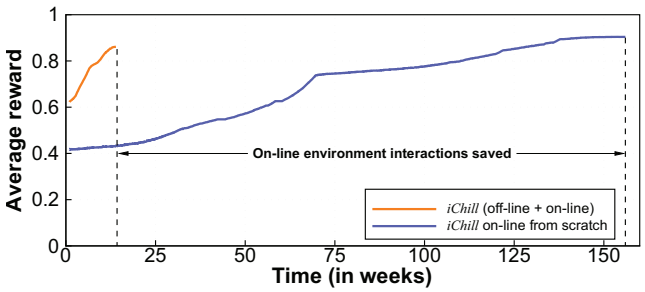

Figure 6: DRL convergence of *iChill* with and without off-line pre-training.

where a small error in the model caused MPC to perform poorly. Note that BL4 still meets the cooling load at all times (no unmet hours) since there are no errors in the cooling load profile given to the MPC but it does so with additional energy consumption.

The next baseline BL5 (*iChill*'s model-free DRL that converges after 3-years) gives a savings of 11.5% and meets the cooling load at all times. BL5's solution is comparable with oracle MPC (BL2), indicating that the solution is near-optimal. However, a disadvantage of BL5 is that it requires lengthy training to learn the optimal policy. Such prolonged experimentation may not be feasible in the real-world. Reducing the training period to one-year results in low energy savings of only 2% over BL1 and high percentage cooling unmet of 55%. This indicates that to maintain performance while reducing the on-line training period, off-line training is needed. Note that BL4 and BL5 constitute an ablation study of *iChill*. Specifically, BL4 represents *iChill* with DRL removed (MPC + M^2L) and BL5 is *iChill* with M^2L removed (fully on-line DRL). Note that *iChill* uses a combination of off-line training with M^2L and then limited online re-training. BL6 shows the need for real-world retraining. In BL6, the control policy learned off-line is exploited in the real-world system without additional exploration or re-training. Compared with *iChill* (limited online re-training), the savings drop by 5.4% for BL6 with negligible unmet cooling load hours; this indicates that limited online re-training helps to improve the control policy. The experiment design of BL6 is similar to BL4 in terms of no real-world re-training involved; and give similar performances.

To validate the hypothesis that off-line training helps, we trained *iChill* off-line till convergence using the chiller power models developed using M^2L. This approximate policy learned off-line is used to initialise the on-line training that continues till convergence. Figure 6 shows the learning convergence of *iChill* with and without the off-line training. The X-axis shows the epoch, and the Y-axis shows the average reward obtained by the agents. We find that off-line pre-training (with approximate M^2L model) reduces the convergence time of on-line training (with real-world chillers) by almost 11× (from 1095 days to 96 days). This is because DRL learns to work around the error in the predicted power of the M^2L model. The corresponding energy savings is about 10% over BL1, which is good compared to 12% for complete training. Facility managers may choose these 96 days to be weekends/public-holidays, without affecting the normal business operations. In sum, a combination of model-assisted off-line training and model-free on-line training enables *iChill* to learn a near-optimal chiller scheduling policy while reducing experimentation in a real-world environment.

7 LIMITATIONS

We considered only water-cooled chillers where the condenser water temperature is tightly controlled through cooling towers. This allowed us to omit the ambient temperature as a part of the feature vector for the chiller power consumption model. However, ambient wet-bulb temp. should be considered in the state vector when condenser water temperature varies; and dry-bulb temperature needs to be considered for air-cooled condensers. Because DRL handles the cooling load that varies with ambient temperature, we believe this variation in the condenser temperature could be handled with more extensive training. A smaller control time-step may help improve the optimal control solution. However, this would require additional information such as time-taken to ramp up to full-load from low part-loads; water distribution network inertia; and minimum ON (OFF) time of a chiller. Further, we did not model the heat gain in the distribution network. Besides, chillers may be required to be run for a minimum number of hours to ensure their mechanical integrity. Last, a chiller's performance degrades over time. We did not account for these in the current work. We believe that many of these requirements can be handled through RL reformulation and additional re-training.

8 CONCLUSION

We considered the problem of scheduling chillers and setting their LWT. Using meta-learning, we developed a model for the power consumption of target chillers at unseen LWTs. Using the meta-learned model, we trained a DRL algorithm off-line using a simulator. We found that meta-learning performs better than transfer learning for power prediction. DRL achieves MPC-like performance with meta-learned models. Further, pre-training DRL makes it more robust than MPC to modelling errors; and significantly reduces the on-line training. Directions for future work include: 1) explore advanced domain adaptation techniques to predict chiller power with limited data; 2) meta-learning a generalised scheduling policy; and 3) full-system RL modelling including condenser side and building AHU level details.

REFERENCES

[1] US energy information administration, residential and commercial energy consumption survey, 2015.

[2] *ASHRAE Handbook - HVAC applications.* ASHRAE, 2019.

[3] S. Alonso, A. Moran, M. Prada, P. Reguera, J. Fuertes, and M. DomAnguez. A data-driven approach for enhancing the efficiency in chiller plants: A hospital case study. *Energies*, 12:827, 03 2019.

[4] A. Beghi, L. Cecchinato, and M. Rampazzo. A multi-phase genetic algorithm for the efficient management of multi-chiller systems. *Energy Conversion and Management*, 52:1650–1661, 03 2011.

[5] B. Chen, Z. Cai, and M. Bergés. Gnu-rl: A precocial reinforcement learning solution for building HVAC control using a differentiable mpc policy. In *Proceedings of the 6th ACM BuildSys*, pages 316–325. ACM, 2019.

[6] Y. Chen, Y. Zheng, and H. Samuelson. Fast adaptation of thermal dynamics model for predictive control of HVAC and natural ventilation using transfer learning with deep neural networks. In *2020 American Control Conference (ACC)*, pages 2345–2350, 2020.

[7] E. Cheng. Dynamic chiller plant optimization. ATAL Building Services Engineering Ltd, 2018.

[8] J. Cigler, J. Siroky, M. Korda, and C. Jones. On the selection of the most appropriate mpc problem formulation for buildings. In *11th REHVA World Congress*, 2013.

[9] D. B. Crawley, C. O. Pedersen, L. K. Lawrie, and F. C. Winkelmann. Energyplus: Energy simulation program. *ASHRAE Journal*, 42:49–56, 2000.

[10] B. Dong, Z. O'Neill, D. Luo, and T. Bailey. Development and calibration of an online energy model for campus buildings. *Energy and Buildings*, 76, 2014.

[11] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1126–1135, 2017.

[12] Y. Gao, Y. Ruan, C. Fang, and S. Yin. Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data. *Energy and Buildings*, 223:110156, 2020.

[13] A. Hooshmand and R. Sharma. Energy predictive models with limited data using transfer learning. In *Proceedings 10th ACM e-Energy*, page 12–16, 2019.

[14] S. Huang, W. Zuo, and M. Sohn. A new method for the optimal chiller sequencing control. In *Proceedings of 14th IBPSA*, pages 316–323, 2015.

[15] S. Hussain, R. Yuen, and G. Huang. Degree of freedom based set-point reset scheme for HVAC real-time optimization. *Energy and Buildings*, 128, 07 2016.

[16] M. Hydeman, K. Jr, and A. Dexter. Tools and techniques to calibrate electric chiller component models. *ASHRAE Transactions*, 108:733–741, 01 2002.

[17] F. Jabari, M. Mohammadpourfard, and B. Mohammadi-ivatloo. Energy efficient hourly scheduling of multi-chiller systems using imperialistic competitive algorithm. *Computers & Electrical Engineering*, 82:106550, 2020.

[18] W. Jiang and T. Reddy. Reevaluation of the gordon-ng performance models for water-cooled chillers. In *ASHRAE Transactions*, volume 109 PART 2, pages 272–287, 2003.

[19] Z. Jiang and Y. M. Lee. Deep transfer learning for thermal dynamics modeling in smart buildings. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2033–2037, 2019.

[20] Z. Li, G. Huang, and Y. Sun. Stochastic chiller sequencing control. *Energy and Buildings*, 84:203 – 213, 2014.

[21] D. Monfet and R. Zmeureanu. Calibration of a central cooling plant model using manufacturer's data and measured input parameters and comparison with measured performance. *Journal of Building Performance Simulation*, 6(2).

[22] S. Nagarathinam, V. Menon, A. Vasan, and A. Sivasubramaniam. MARCO - multi-agent reinforcement learning based control of building HVAC systems. In *Proceedings 11th ACM e-Energy*, page 57–67, 2020.

[23] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *ArXiv*, abs/1803.02999, 2018.

[24] P. Peng, Y. Wen, Y. Yang, Q. Yuan, Z. Tang, H. Long, and J. Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games, 2017.

[25] M. Saeedi, M. Moradi, M. Hosseini, A. Emamifar, and N. Ghadimi. Robust optimization based optimal chiller loading under cooling demand uncertainty. *Applied Thermal Engineering*, 148:1081–1091, 02 2019.

[26] E. Sala-Cardoso, M. Delgado-Prieto, K. Kampouropoulos, and L. Romeral. Predictive chiller operation: A data-driven loading and scheduling approach. *Energy and Buildings*, 208:109639, 2020.

[27] F. L. D. Silva, R. Glatt, and A. H. R. Costa. Moo-mdp: An object-oriented representation for cooperative multiagent reinforcement learning. *IEEE Transactions on Cybernetics*, 49:567–579, 2019.

[28] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction.* A Bradford Book, Cambridge, MA, USA, 2018.

[29] H. D. Vu, K. S. Chai, B. Keating, N. Tursynbek, B. Xu, K. Yang, X. Yang, and Z. Zhang. Data driven chiller plant energy optimization with domain knowledge. In *Proceedings ACM CIKM*, page 1309–1317, 2017.

[30] T. Wei, Y. Wang, and Q. Zhu. Deep reinforcement learning for building HVAC control. In *Proceedings 54th ACM DAC*, 2017.

[31] X. Wei and G. Xu. Modeling and optimization of a chiller plant. *Energy*, 73:898–907, 08 2014.

[32] M. Wetter. Co-simulation of building energy and control systems with the building controls virtual test bed. *Journal of Building Performance Simulation*, 4(3):185–203, 2011.

[33] C. Wu, A. R. Kreidieh, K. Parvate, E. Vinitsky, and A. M. Bayen. Flow: A modular learning framework for mixed autonomy traffic. *IEEE Transactions on Robotics*, pages 1–17, 2021.

[34] W. Xu, J. He, and Y. Shu. Transfer learning and deep domain adaptation, 2020.

[35] F. Yu and K. Chan. Optimum load sharing strategy for multiple-chiller systems serving air-conditioned buildings. *Building and Environment*, 42(4), 2007.

[36] F. W. Yu and K. Chan. Optimization of water-cooled chiller system with load-based speed control. *Applied Energy*, 85:931–950, 10 2008.

[37] C. Zhang, S. R. Kuppannagari, R. Kannan, and V. K. Prasanna. Building HVAC scheduling using reinforcement learning via neural network based model approximation. In *Proceedings 6th ACM BuildSys*, pages 287–296, 2019.

[38] Z. Zhang, A. Chong, Y. Pan, C. Zhang, S. Lu, and K. P. Lam. A deep reinforcement learning approach to using whole building energy model for HVAC optimal control. In *2018 Building Performance Analysis Conference and SimBuild*, 2018.

[39] Z. Zheng, Q. Chen, C. Fan, N. Guan, A. Vishwanath, D. Wang, and F. Liu. Data driven chiller sequencing for reducing HVAC electricity consumption in commercial buildings. In *Proceedings of the 9th ACM e-Energy*, page 236–248, 2018.

[40] Z. Zheng, D. Xie, J. Pu, and F. Wang. Melody: Adaptive task definition of COP prediction with metadata for HVAC control and electricity saving. In *Proceedings 11th ACM e-Energy*, page 47–56, 2020.