

Decision-Focused Learning for Neural Network-Constrained HVAC Scheduling

Pietro Favaro, *Student Member, IEEE*, Jean-François Toubeau, *Member, IEEE*, François Vallée, *Member, IEEE*, and Yury Dvorkin, *Member, IEEE*,

Abstract—Heating, Ventilation, and Air Conditioning (HVAC) is a major electricity end-use with a substantial potential for providing grid services, such as demand response. Harnessing this flexibility requires accurate modeling of the thermal dynamics of buildings, a difficult task because nonlinear heat transfer and recurring daily cycles make historical data highly correlated and insufficient to generalize to new weather, occupancy, and control scenarios. This paper presents an HVAC management system formulated as a Mixed Integer Quadratic Program (MIQP), where Neural Network (NN) models of thermal dynamics are embedded as exact mixed-integer linear constraints. Unlike traditional training approaches that minimize prediction errors, we employ Decision-Focused Learning (DFL) to learn the NN parameters with the objective of directly improving the HVAC cost performance. However, the discrete nature of MIQP hinders DFL, as it leads to undefined and discontinuous gradients, thus impeding standard gradient-based training. We leverage Stochastic Smoothing (SS) to enable efficient gradient computation without the need to differentiate the MIQP. Experiments on a realistic five-zone building using a high-fidelity simulator demonstrate that the proposed SS-DFL approach outperforms conventional identify-then-optimize (i.e., the thermal dynamics model is identified on historical data then used in optimization) and relaxed DFL methods in both cost savings and grid service performance, highlighting its potential for scalable, grid-aware building control.

Index Terms—Building Energy Management, Differentiable Optimization, Decision-Focused Learning, Demand Response, Energy Management System, Stochastic Smoothing.

I. INTRODUCTION

BUILDINGS account for over 30% of global energy use [1], making them central to the low-carbon energy transition. Among their energy-intensive components, Heating, Ventilation, and Air Conditioning (HVAC) accounts for around 65% of total energy consumption in European households [2], yet it remains a largely untapped source of flexibility [3]–[5]. Controlling HVAC settings can reshape energy consumption patterns [6], improve efficiency [7], and even support grid stability [8], [9]. However, unlocking this potential requires understanding and controlling building thermal dynamics, i.e., an intricate problem where physics, data, and optimization must be seamlessly integrated into the decision-making pipeline.

A major opportunity to leverage HVAC flexibility is the participation in the day-ahead electricity market, the main platform for trading electricity between grid actors. In Europe, this market is cleared daily, setting electricity volumes and prices for each hour of the following day [10]. This timeline

aligns naturally with the slow thermal dynamics of buildings, where temperature adjustments from HVAC decisions and external conditions unfold gradually over time. Capturing these effects is crucial to maintain occupants’ comfort [7]. Thermal models for buildings generally fall into two main categories: physics-based and data-driven¹.

Physics-based (i.e., white-box) models rely on detailed information about a building’s physical characteristics, including construction materials, insulation properties, and ventilation dynamics. This category includes high-fidelity simulators such as EnergyPlus [12]. While these models provide highly detailed representations of buildings [13], some parameters, such as the materials resistance and capacitance, are inherently uncertain due to factors like aging and variability in the installation process, which can lead to significant errors [14]. Moreover, it requires expertise in building thermal modeling, which may not be readily available at each building site. But, there also exist simpler models: aggregated (or lumped) Resistance-Capacitance (RC) models. These models are linear and based on circuit analogy between heat transfer and electricity. The parameters of these models can be computed based on the construction materials [15], or by data-driven regression [16].

In contrast, data-driven (i.e., black-box) models rely on statistical or Machine Learning (ML) techniques trained on historical weather and energy consumption data. These models range from simple linear regressions [17] to highly complex nonlinear architectures, such as Neural Network (NN) models [18], [19]. In [20], physics-informed NNs were employed for control-oriented predictions. However, these NN approaches did not consider day-ahead scheduling, which introduces fundamentally different requirements. Table I summarizes the thermal modeling approaches.

NNs can capture complex patterns with a limited manual modeling effort, and will thus be used in this paper. To fully exploit the capabilities of NN models for HVAC control, it is essential to unify modeling and control tasks within a single framework. Such an integrated approach would allow HVAC systems to make energy-efficient planning decisions while maintaining occupants’ comfort. A first promising avenue is to model the control policy by a NN that outputs control decisions, allowing it to internalize the building dynamics. Traditionally, such NNs in real-time control are trained using reinforcement learning (RL) [21], which directly optimizes decisions through interaction with the environment. Alterna-

P. Favaro, J.F. Toubeau, F. Vallée are with the University of Mons, Belgium
Y. Dvorkin is with Johns Hopkins University, Baltimore, USA

¹A comprehensive review of the literature is provided in [11], Section III.

TABLE I: Comparison of thermal dynamics modeling approaches.

	Model	Complexity	Data Requirement	Ref.
White box	High-fidelity simulator	Very high	Geometry, materials, weather	[12]–[14]
	Lumped RC (known R/C)	Low	R and C	[15]
Gray box	Lumped RC (parameter fitting)	Moderate	I/O measurements for R/C	[16]
	Physics-informed NN	High	Partial physics + data	[20]
Black box	NN	Moderate	Historical input–output data	[18], [19]
	Regression models	Low	Historical system data	[17]

tively, self-supervised learning can be employed, but, similarly to RL, it often lacks guarantees on solution quality, especially in the presence of hard physical and operational constraints. For example, the method in [22] ensures feasibility but not optimality and cannot be applied to discrete problems, while extensions to mixed-integer nonlinear programming fail to guarantee either feasibility or optimality [23]. Self-supervised methods such as Primal-Dual Learning (PDL) [24] and deep Lagrangian dual networks [25] improve feasibility by integrating dual information of the underlying optimization. However, PDL relies on iterative dual updates and may struggle with nonconvex or stochastic problems, whereas the approach in [25], which predicts solutions to the AC Optimal Power Flow problem, is designed around problem-specific structures, which prevents generalization to settings with uncertainty or combinatorial decision variables. A second promising avenue is provided by NN-constrained optimization, where the learned thermal model, represented by a NN, is embedded directly into mathematical optimization as a set of constraints. This approach merges the expressiveness of NN with the stability and decision quality of constrained optimization. It has been shown that, by using a feedforward NN with Rectified Linear Unit (ReLU), the constraints can be exactly encoded as a set of Mixed-Integer Linear (MIL) equations [26].

In this paper, we formulate the day-ahead HVAC Management System (MS) as an NN-constrained optimization problem. The NN, which models the building’s thermal dynamics, is encoded precisely as a set of mixed-integer linear constraints within the optimization problem, enabling the optimization model to incorporate learned dynamics while ensuring decision quality and feasibility.

A. Literature Review

Although NNs offer strong modeling capabilities, their inherent approximation errors can propagate into suboptimal energy management decisions. Traditional ML models are trained to minimize statistical metrics such as Mean Squared Error (MSE), thus overlooking how the learned model affects downstream control decisions. In contrast, Decision-Focused Learning (DFL) aligns ML training with optimization objectives to improve decision quality [27]. DFL typically relies on gradient-based training methods. The core challenge in this approach is two-fold: (i) computing the sensitivity of

the optimization problem’s solution with respect to its input parameters, and (ii) ensuring that the resulting gradients are meaningful and informative for learning [28].

A foundational contribution in DFL addressed the differentiation of unconstrained problems. In [29], the authors differentiated an unconstrained Quadratic Problem (QP) to train a ML model that predicts the uncertain parameters of the QP. Donti et al. extended the framework to constrained QP [30] by relaxing the constraints in the objective function to compute the gradient. Agrawal et al. [31] enable differentiation through constrained convex programs using self-homogeneous embeddings of conic problems, allowing learning of controller parameters in the objective under known, differentiable dynamics and observable system behavior. Constrained convex optimization can also be differentiated via implicit differentiation of its Karush-Kuhn-Tucker conditions [32]. This framework has been applied to learn the parameters of building thermal dynamics models. In [33], Chen et al. propose a differentiable Model Predictive Control (MPC) policy that jointly learns the thermal dynamics and optimizes HVAC operations. Even though pioneering, their approach relies on linear thermal models that yield convex optimization problems, limiting its applicability to more realistic, non-convex building dynamics. Cui et al. extended this approach to include a forecaster upstream of the convex optimization problem [34]. They jointly learn the linear thermal model and a forecaster for its residuals. Other applications include forecasting electricity prices [35] and optimizing the allocation of flexibility between transmission and distribution grid assets [36].

In this paper, to extend DFL to the scheduling of buildings with a NN modeling the thermal dynamics, we need to differentiate through a Mixed-Integer Problem (MIP), which is particularly difficult due to the presence of (discrete) integer decision variables. As a result, the gradient of the MIP solution with respect to the MIP parameters is typically either zero or undefined, making it incompatible with standard gradient-based learning methods. One possible approach is to replace the MIP with its continuous relaxation, allowing the use of the existing DFL methods [31]. While relaxing a MIP to its continuous counterpart reduces computational complexity, it alters the NN representation, i.e., from capturing a nonlinear hypersurface to approximating it as a polytope, compromising modeling fidelity. Another approach generates cuts to obtain a linear program that has the same solution as the original mixed-integer formulation [37]. However, in addition to the implementation difficulty, this method is extremely burdensome, since a new set of cuts must be generated for each training instance.

To tackle these challenges and thus enable DFL for the NN-constrained HVAC planning problem, we propose to leverage Stochastic Smoothing (SS). This technique introduces random perturbations to the uncertain parameters. Early approaches use this perturbation to smooth the “max” operator limiting its applicability to problems in which the unknown parameters are in the linear objective function [38]. We go further by extracting informative gradients from the task-specific loss using the score function of the REINFORCE algorithm [39]. Unlike the aforementioned methods, SS does not make any

assumption about the optimization problem class, the position of the uncertain parameters (whether in the objective function and/or the constraints), and the specific form of the loss function. Therefore, SS obtains a gradient from ex-post signals directly such as the true realized cost or constraint deviation, without requiring differentiability. SS ensures high versatility of the proposed framework and is highly suitable for combinatorial optimization involving a non-differentiable dynamical system.

B. Contributions

The main contributions of this paper can be summarized as:

- 1) We train a NN that models the building's thermal dynamics using DFL to optimize the decision quality rather than minimize a task-agnostic statistical metric. We formulate an HVAC day-ahead management as a Mixed-Integer Linear Program (MILP) in which the building's thermal dynamics are modeled by a NN and embedded directly as optimization constraints. To enable effective and robust gradient-based training of the NN, we (i) reformulate the MILP problem as an MIQP to avoid infeasibility during learning, (ii) solve the MIQP with random perturbations applied to unknown NN parameters to encourage exploration, (iii) estimate gradients using a score-function method that directly relates learning with decision quality.
- 2) We embed a piecewise linear NN in the HVAC management system to obtain an expressive and tractable nonlinear model of the building's thermal dynamics. We reformulate each ReLU activation as a set of Mixed-Integer Linear (MIL) constraints, providing solution quality guarantees via a Mixed-Integer Programming (MIP) optimality gap. This approach yields a trade-off between modeling accuracy of the dynamics and computational efficiency.
- 3) Because reformulating NNs with ReLU activations in MILP requires Big-M constants, we improve the standard formulation by adaptively tightening them. Specifically, instead of relying on fixed Big-M bounds, we dynamically adjust the feasible input intervals: when an input value is known in advance (i.e., it is a deterministic parameter), its interval is reduced to a single value. This adaptive strategy produces a tighter formulation than existing fixed Big-M methods [40], [41], leading to more efficient optimization without sacrificing correctness. The benefit of this approach is demonstrated through both theoretical analysis (Section II-D) and empirical validation (Section III-E).

Our framework bridges the gap between ML, constrained optimization, and real-world energy management, paving the way for more intelligent and efficient HVAC control strategies. We show the effectiveness of our approach on a realistic five-zone office building located in Denver, USA. Moreover, we compare the performance of multiple thermal models: various NN architectures and lumped RC model trained on historical data, or via traditional DFL methods.

C. Outline

In Section II, we describe the HVAC MS model, including the reformulation of NN as a set of mixed-integer linear constraints, and the derivation of the resulting MILP problem. In Section III, we introduce the case study and evaluate the performance of thermodynamic models of increasing complexity, from the RC linear model to the NN model. We compare the effectiveness of training these models in ITO and DFL fashions. In addition, we analyze the impact of the proposed tight formulation of ReLU, the standard deviation of the noise, and the number of samples on our DFL strategy. Finally, we summarize the key findings and outline the potential directions for future research in Section IV.

II. MODEL AND METHODS

A. HVAC Management System

The objective of day-ahead HVAC MS is to minimize the following day's operational cost on behalf of the building manager. This scheduling task is challenging, as it must account for nonlinear building thermodynamics, efficient energy use, and sufficient thermal comfort for the occupants. In general, the easiest way to control the HVAC system is through thermostats, which are widely available in buildings. Therefore, the decision variable of the scheduling problem is the matrix of indoor temperature setpoints for each time step and thermal zone². In the following model to optimize the day-ahead HVAC operation, we assume that the electricity cost λ_t^i is lower than the feed-in tariff λ_t^e for each time step.

The objective function (1) of the day-ahead HVAC MS is:

$$\min_{\tau^{\text{in}}} \underbrace{p^{\text{d}} \lambda^{\text{d}}}_{(i)} + \sum_{t=0}^T \left(\underbrace{p_t^{\text{i}} \lambda_t^{\text{i}} - p_t^{\text{e}} \lambda_t^{\text{e}}}_{(ii)} \right) \Delta t \quad (1)$$

where we minimize:

- (i) the peak demand cost by applying a charge λ^{d} to the daily power consumption peak p^{d} ;
- (ii) the electricity cost aggregated at the building level which is, at each time step, the difference between the cost of consuming grid electricity $\lambda_t^{\text{i}} p_t^{\text{i}}$ and the revenue from power injection into the grid $\lambda_t^{\text{e}} p_t^{\text{e}}$.

The optimization is constrained in (2) by the building thermal dynamics from time steps t to $t+1$. The relationship is modeled by a NN with parameters θ . The vector of zonal indoor temperatures τ_{t+1}^{in} is obtained based on the input from the previous time step t : the vector of zonal indoor temperatures $\tau_t^{\text{in}} \in \mathbb{R}^Z$ where Z is the number of zones, the vector of electrical power for zonal heating $p_t^{\text{h}} \in \mathbb{R}^Z$, the vector of electrical power for zonal cooling $p_t^{\text{c}} \in \mathbb{R}^Z$, and ambient temperature $\tau_t^{\text{amb}} \in \mathbb{R}$.

$$\tau_{t+1}^{\text{in}} = \text{NN}(\tau_t^{\text{in}}, p_t^{\text{h}}, p_t^{\text{c}}, \tau_t^{\text{amb}}; \theta) \quad \forall t \quad (2)$$

²A thermal zone in a building is a space or collection of spaces with similar space-conditioning requirements, typically sharing the same heating and cooling setpoint and controlled by a single thermostat or thermal control device.

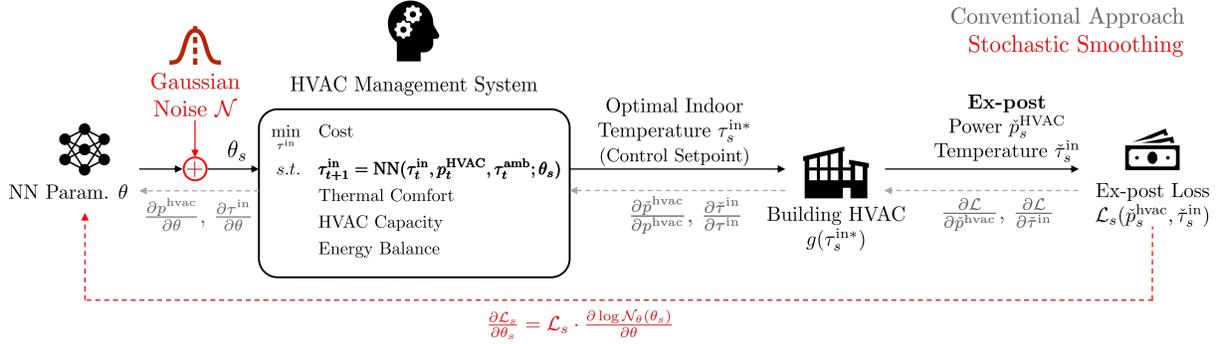


Fig. 1: Comparison of the proposed stochastic smoothing DFL pipeline (red) for HVAC management system with conventional DFL approaches (gray).

The initial state of the building, i.e., the set of initial zonal indoor temperatures, are given for all zones $z \in \mathcal{Z}$ by:

$$\tau_{0,z}^{\text{in}} = T_{0,z}^{\text{in}}. \quad (3)$$

In addition, zonal indoor temperature $\tau_{t,z}^{\text{in}}$ is restricted to lie within comfort limits $\underline{T}_{t,z}^{\text{in}}$ and $\overline{T}_{t,z}^{\text{in}}$.

$$\underline{T}_{t,z}^{\text{in}} \leq \tau_{t,z}^{\text{in}} \leq \overline{T}_{t,z}^{\text{in}} \quad (4)$$

Heating and cooling power consumptions are positive variables bounded by the HVAC capacities \overline{P}_z^c and \overline{P}_z^h :

$$0 \leq p_{t,z}^c \leq \overline{P}_z^c, \quad \forall t, z; \quad (5)$$

$$0 \leq p_{t,z}^h \leq \overline{P}_z^h, \quad \forall t, z. \quad (6)$$

We define the zonal HVAC electrical power $p_{t,z}^{\text{hvac}}$ as the sum of the powers for cooling and heating (7). The resulting energy balance is ensured by (8), in which the net power consumption of the building comprises the HVAC power consumption $p_{t,z}^{\text{hvac}}$ and the non-dispatchable load P_t^{nd} , minus the building's on-site power generation P_t^{gen} , e.g., from rooftop photovoltaic panels.

$$p_{t,z}^{\text{hvac}} = p_{t,z}^h + p_{t,z}^c \quad \forall t, z \quad (7)$$

$$p_t^i - p_t^e = \sum_{z=0}^Z p_{t,z}^{\text{hvac}} + P_t^{\text{nd}} - P_t^{\text{gen}} \quad \forall t \quad (8)$$

The peak power demand p^{d} is the maximum power consumption (or injection) from the grid over the day. The peak demand is greater than or equal to all power exchanges $p_t^i + p_t^e$ across the day (9). While conceptually equivalent to defining $p^{\text{d}} = \max\{p_t^i, p_t^e \mid t \in \mathcal{T}\}$, constraint (9) avoids introducing a bilevel structure into the optimization problem, preserving tractability.

$$p^{\text{d}} \geq p_t^i + p_t^e \quad \forall t \quad (9)$$

At all times, power imported from or exported to the grid must comply with the line capacity \overline{P}^1 of the connection between the building and the grid:

$$p_t^i \leq \overline{P}^1 \quad \forall t, \quad (10)$$

$$p_t^e \leq \overline{P}^1 \quad \forall t. \quad (11)$$

Finally, all power imports and exports must be positive:

$$p_t^i, p_t^e \geq 0 \quad \forall t. \quad (12)$$

B. Decision-Focused Learning

1) *Problem Formulation*: Our objective is to learn the NN parameters θ (i.e., weights and biases) of constraint (2). Instead of training θ in a task-agnostic manner on historical data, we aim to learn θ to directly improve HVAC management decisions as shown by Figure 1.

To formalize this DFL problem, we introduce r , a vector containing all stochastic parameters of the HVAC MS (e.g., weather conditions) excluding θ . We assume r follows the known training distribution \mathcal{R} . The goal is to minimize the task loss \mathcal{L} , which reflects the true operational goal of the HVAC MS. This naturally leads to a bilevel optimization problem, whose solution yields the optimal values of θ :

$$\min_{\theta} \mathbb{E}_{r \sim \mathcal{R}} [\mathcal{L}(\check{p}^{\text{hvac}}, \check{\tau}^{\text{in}})] \quad (13)$$

$$\text{s.t. } \check{p}^{\text{hvac}}, \check{\tau}^{\text{in}} = g(\tau^{\text{in}*}) \quad (14)$$

$$\min f(p^{\text{hvac}}, \tau^{\text{in}}) \quad (15)$$

$$\text{s.t. } (2) - (12) \quad (16)$$

The inner problem (15)-(16) is the HVAC MS, including the NN formulation, which is thus a MILP. The outer problem aims to find θ by minimizing the expected task loss over the distribution of input parameters r (13). Ideally, the task loss is the ex-post value of the decisions. Thus, it depends on the realized ex-post measurements of the HVAC power \check{p}^{hvac} , and indoor temperature $\check{\tau}^{\text{in}}$. These ex-post measurements are obtained by controlling the HVAC system with optimal indoor temperature profiles $\tau^{\text{in}*}$ (14). In (14), g is the mapping between $\tau^{\text{in}*}$ and the ex-post measures \check{p}^{hvac} and $\check{\tau}^{\text{in}}$. The function g models HVAC actuation, building response, and sensing.

2) *Gradient Computation*: Problem (13)-(16) is intractable and realistically large instances cannot be handled by off-the-shelf solvers. Therefore, we aim at learning θ by gradient descent. The backpropagation step of the conventional approach applies the chain rule, but for tractability this requires g to be differentiable. In practice, real systems, and even some building simulators, are not differentiable.

To address this issue, Stochastic Smoothing (SS) aims to learn directly from an ex-post loss signal, analogous to the reward in reinforcement learning (Figure 1). SS does not make any assumption about the form of the HVAC MS, the differentiability of g , and the structure of the ex-post loss. To

that end, instead of producing a point estimate of θ , SS models it as a distribution, e.g., a Gaussian $\mathcal{N}(\theta, \sigma)$. Because of the stochasticity in θ , the loss \mathcal{L} in (13) becomes an expectation:

$$\mathcal{L} = \mathbb{E}_{\theta_s \sim \mathcal{N}(\theta, \sigma)} [\mathcal{L}_s(\check{p}_s^{\text{hvac}}, \check{\tau}_s^{\text{in}})]. \quad (17)$$

By modeling probabilistic outputs, this approach avoids uninformative gradients that plague traditional deterministic methods. Even when the original gradient (e.g., from a LP) would be zero, taking the expectation over a smoothed distribution can yield informative, non-zero gradient signals. However, computing these gradients analytically is challenging.

To train the model, we thus employ score-function gradient estimation, a technique closely related to the REINFORCE algorithm from reinforcement learning. The approximation of the gradient becomes [39]:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \mathbb{E}_{\theta_s \sim \mathcal{N}(\theta, \sigma)} \left[\mathcal{L}_s \frac{\partial \log \mathcal{N}_\theta(\theta_s)}{\partial \theta} \right]. \quad (18)$$

In practice, the expectation of the loss \mathcal{L} is approximated using Monte Carlo with S samples [42]:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{1}{S} \sum_{s=1}^S \left[\mathcal{L}_s \frac{\partial \log \mathcal{N}_\theta(\theta_s)}{\partial \theta} \right]. \quad (19)$$

Although SS can accommodate any type of optimization problem, we reformulate the MILP as a Mixed-Integer Quadratic Program (MIQP) by relaxing the indoor temperature constraint (4) and incorporating it as a quadratic penalty in the objective function (20).

Reformulating the problem as a MIQP is not strictly required to obtain meaningful gradients, since SS is agnostic to the optimization structure, but it helps stabilize the learning process. By penalizing constraint violations in the objective (in the MIQP) rather than enforcing them strictly (in the original MILP), the MIQP approach prevents infeasibility during training, thereby ensuring smoother and more robust model updates. Moreover, the hard constraints can be enforced at test time, if desired.

The quadratic term (iii) penalizes deviation of zonal indoor temperature $\tau_{t,z}^{\text{in}}$ from the target temperature $T_{t,z}^{\text{tgt}}$, reflecting the loss of thermal comfort of the occupants. The weight $o_{t,z}$ should be designed to reflect the occupancy of the zone, i.e., higher occupancy implies a greater importance of thermal comfort and thus a stronger penalty.

$$\min_{\tau^{\text{in}}} p^{\text{d}} \lambda^{\text{d}} + \sum_{t=0}^T (p_t^{\text{i}} \lambda_t^{\text{i}} - p_t^{\text{e}} \lambda_t^{\text{e}} + \underbrace{\sum_{z=0}^Z o_{t,z} (\tau_{t,z}^{\text{in}} - T_{t,z}^{\text{tgt}})^2}_{(iii)}) \quad (20)$$

The resulting algorithm is detailed in Algorithm 1.

Algorithm 1 Algorithm for decision-focused learning of the HVAC management system.

```

1: Input:
2: Database of exogenous parameters  $R$ 
3: Initial NN parameters  $\theta_0$ 
4: for each epoch do
5:   for each parameter sample  $r$  in  $R$  do
6:     for each sample  $s$  do
7:       Day-Ahead Stage:
8:       Sample  $\theta_s \sim \mathcal{N}(\theta, \sigma)$ 
9:       Solve MIQP HVAC MS problem
10:      Simulator or Real Building:
11:      Thermostat control with  $\tau_s^{\text{in}*}$  as setpoints
12:      Get ex-post power  $\check{p}_s^{\text{hvac}}$  and temperature  $\check{\tau}_s^{\text{in}*}$ 
13:      Ex-Post Analysis:
14:      Compute loss function  $\mathcal{L}_s(\check{p}_s^{\text{hvac}*}, \check{\tau}_s^{\text{in}*})$ 
15:    end for
16:    Backward Pass:
17:    Compute gradient  $\frac{\partial \mathcal{L}}{\partial \theta} \approx \frac{1}{S} \sum_{s=1}^S \mathcal{L}_s \frac{\partial \log \mathcal{N}_\theta(\theta_s)}{\partial \theta}$ 
18:    Update NN parameters  $\theta_{n+1} \leftarrow \theta_n - \alpha \frac{\partial \mathcal{L}}{\partial \theta}$ 
19:  end for
20: end for

```

3) *Task Loss*: The task loss \mathcal{L} aims to evaluate the quality of the decisions made by the HVAC MS. The ideal task loss is the exact ex-post value of the decisions, which includes both the cost and the thermal comfort. Here, we align our loss \mathcal{L} with the HVAC MS objective (20). We name the loss *Expost+* and define it as:

$$\mathcal{L} = \underbrace{\sum_{t=0}^T (\check{p}_t^{\text{hvac}} \check{\lambda}_t + \sum_{z=0}^Z o_{t,z} (\Delta \check{\tau}_{t,z}^{\text{in}})^2)}_{\text{Ex-post cost}} + \underbrace{\text{MSE}(\check{C} - C)}_{\text{MSE Power Cost}}, \quad (21)$$

where $\Delta \check{\tau}_{t,z}^{\text{in}} = \check{\tau}_{t,z}^{\text{in}} - T_{t,z}^{\text{tgt}}$ and $\text{MSE}(\check{C} - C) = \frac{1}{T} \sum_{t=0}^T (\check{p}_t^{\text{hvac}} \check{\lambda}_t - p^{\text{hvac}} \lambda_t)^2$. The ex-post price $\check{\lambda}_t$ depends on the ex-post import and export powers \check{p}_t^{i} and \check{p}_t^{e} :

$$\check{\lambda}_t = \begin{cases} \lambda_t^{\text{i}} & \text{if } \check{p}_t^{\text{e}} = 0 \text{ and } \check{p}_t^{\text{i}} \neq \check{p}_t^{\text{d}}, \\ \lambda_t^{\text{e}} & \text{if } \check{p}_t^{\text{i}} = 0 \text{ and } \check{p}_t^{\text{e}} \neq \check{p}_t^{\text{d}}, \\ \lambda_t^{\text{i}} + \lambda_t^{\text{d}} & \text{if } \check{p}_t^{\text{i}} = \check{p}_t^{\text{d}}, \\ \lambda_t^{\text{e}} + \lambda_t^{\text{d}} & \text{if } \check{p}_t^{\text{e}} = \check{p}_t^{\text{d}}. \end{cases} \quad (22)$$

C. Neural Network MILP Formulation

Constraint (2) of the model presented in Section II-A represents the building thermal dynamics over a time interval. We propose learning building thermal dynamics in a data-driven fashion by leveraging piecewise linear NNs, which are obtained by designing NNs with only piecewise linear activation functions. Specifically, we use ReLU activation functions, which are made up of two linear pieces. ReLU speeds up training and reaches greater accuracy than other activation functions for deep [43] and sparse NNs [44], but ReLU-based NNs may be nonconvex [45]. The ReLU function is defined as:

$$y = \max(0, \hat{y}). \quad (23)$$

However, embedding (23) directly into the larger HVAC MS optimization makes the overall problem intractable, as it

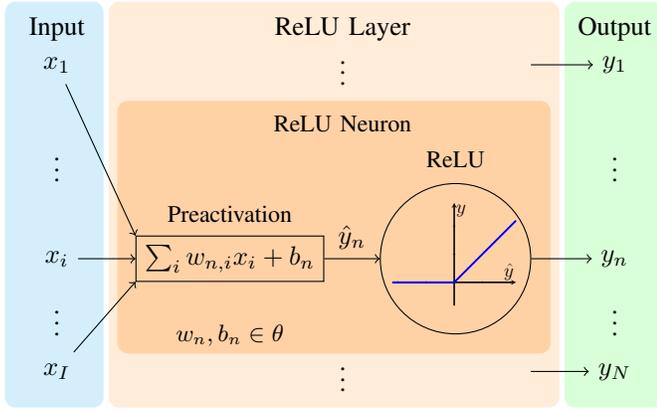


Fig. 2: Layer of neurons with Rectified Linear Unit (ReLU) activation function. The parameters are the weight w_i and the bias b .

requires solving a nested optimization at each ReLU. Consequently, each ReLU neuron $n \in \mathcal{N}$ is reformulated by a binary variable σ_n (24), two continuous variables y_n and \hat{y}_n (25), and a set of constraints. The complexity of the resulting optimization problem scales with the number of neurons N .

$$\sigma_n \in \{0, 1\} \quad (24)$$

$$y_n, \hat{y}_n \in \mathbb{R} \quad (25)$$

The first step, as shown in Figure 2, is to compute the preactivation function (26) of the neuron n based on the input x , the weights w_n , and the bias b_n . For I inputs, $x \in \mathbb{R}^I$, $w_n \in \mathbb{R}^I$, and $b_n \in \mathbb{R}$. The preactivation value \hat{y}_n is a scalar.

$$\hat{y}_n = \sum_{i=1}^I w_{n,i} x_{n,i} + b_n \quad \forall n \quad (26)$$

The second step is to choose a suitable formulation of the ReLU activation function, which is typically categorized into four main types: a disjunctive formulation [46], a Big-M formulation [47], a strong formulation [48], and a partition-based formulation [49]. Despite offering a strong LP relaxation of the ReLU, the strong formulation requires an infinite number of constraints or a significant number of additional auxiliary variables, which hinders its performances. Here, we adopt the Big-M formulation for its simplicity and excellent empirical performance [50]–[53].

First, the Big-M formulation of the ReLU defines the convex hull of the function as follows:

$$y_n \geq 0, \quad (27)$$

$$y_n \geq \hat{y}_n, \quad (28)$$

where \hat{y}_n , the preactivation value, is the input of the ReLU function and y_n is the output. The constraints (29) and (30) are alternatively binding depending on the value of the binary σ_n . The parameters \hat{Y}_n^{\max} and \hat{Y}_n^{\min} are the Big-M constants. When $\sigma_n = 0$, (29) is binding, imposing y_n to be null with (27). Variable \hat{y}_n must be negative (28). In contrast, $\sigma_n = 1$

makes (30) binding, imposing $\hat{y}_n = y_n$ via (28). Variable \hat{y}_n must be positive (27).

$$y_n \leq \hat{Y}_n^{\max} \cdot \sigma_n \quad (29)$$

$$y_n \leq \hat{y}_n - \hat{Y}_n^{\min} \cdot (1 - \sigma_n) \quad (30)$$

D. Improved tightness of the NN reformulation

The Big-M constants \hat{Y}_n^{\min} and \hat{Y}_n^{\max} need to be accurately determined to produce a tight formulation of the ReLU. Three approaches exist in the literature. A first naive approach is to record the minimum and maximum values of \hat{y}_n during training. Although this method preserves the correlation between the NN inputs and remains simple, it is not well suited for NN-constrained optimization since physical constraints rather than the distribution of the NN training dataset should define the feasible domain [53]. More advanced methods formulate optimization problems to find the bounds \hat{Y}_n^{\min} and \hat{Y}_n^{\max} [51], [54]. Bound optimization is computationally cumbersome, making it impractical for online computations. The third approach is to calculate the bounds in all layers of the NN based solely on the input bounds using interval analysis [55]. Interval analysis enables efficient computation of bounds while ensuring physical consistency by appropriately setting the limits of the NN inputs. However, the correlation between inputs is lost.

In this work, the NN parameters θ are updated at each gradient descent during training via DFL (cf. Section II-B), such that the bounds \hat{Y}_n^{\min} and \hat{Y}_n^{\max} must be recomputed after each gradient descent step. To handle this efficiently, we use interval analysis, and further refine the method by setting the feasible interval of NN input to singleton (i.e., degenerate) interval if the input value is known prior to solving the optimization problem. This applies to all optimization parameters that are input to the NN.

Proposition 1 (Preactivation Range Reduction). *Consider a neuron—reformulated as mixed-integer linear constraints using Big-M constants—with preactivation $\hat{y} = w_x x + w_p p + b$, where $x \in \mathbb{R}^X$ are unknown input variables with weights $w^x \in \mathbb{R}^{1 \times X}$, $p \in \mathbb{R}^P$ are known parameters with weights $w^p \in \mathbb{R}^{1 \times P}$, and b is the neuron bias. Assume that x and p are bounded such that $x \in [X^{\min}, X^{\max}]^X$, and $p \in [P^{\min}, P^{\max}]^P$. Let $\Delta \hat{Y}$ denote the feasible range of the preactivation under these bounds. When the feasible range of p is reduced to a degenerated interval $p \in [P_0, P_0]^P$, the feasible range of the modified preactivation $\hat{y}' = w_x x + w_p P_0 + b$ satisfies:*

$$\frac{|\Delta \hat{Y}'|}{|\Delta \hat{Y}|} \approx 1 - \frac{\sum_p^P w_p}{\sum_x^X w_x + \sum_p^P w_p} \quad (31)$$

where $|\cdot|$ denotes the Lebesgue measure (i.e., length) of the interval.

Proof. Assuming we have a ReLU neuron with two inputs $x \in \mathbb{R}$ and $p \in \mathbb{R}$ that are respectively variable and parameter of the optimization problem, as illustrated by Figure 3, we can

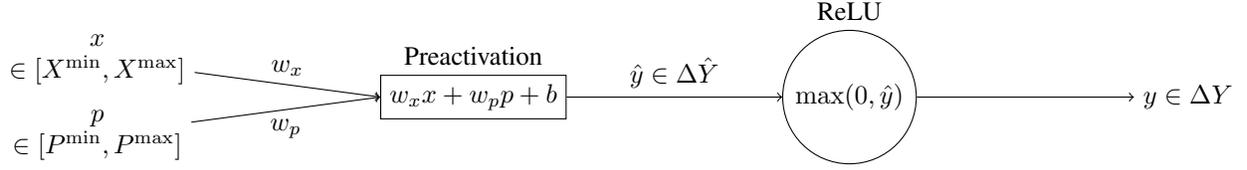


Fig. 3: Interval analysis of the bounds for a ReLU neuron with inputs p and x , where p is a parameter and x a variable in the overarching optimization.

demonstrate the tightening of the bounds by our approach. Let $\Delta(\cdot)$ denote the feasible interval of its argument. If

$$x \in \Delta X = [X^{\min}, X^{\max}], \quad (32)$$

$$p \in \Delta P = [P^{\min}, P^{\max}], \quad (33)$$

then, assuming without loss of generality that the associated weights $w_x \in \mathbb{R}$ and $w_p \in \mathbb{R}$ are positive³, interval analysis defines the bounds of the interval of \hat{y} as follows

$$\hat{Y}^{\min} = w_x X^{\min} + w_p P^{\min} + b, \quad (34)$$

$$\hat{Y}^{\max} = w_x X^{\max} + w_p P^{\max} + b. \quad (35)$$

Therefore, the interval length according to Lebesgue measure $|\cdot|$ on \hat{y} is

$$|\Delta \hat{Y}| = \hat{Y}^{\max} - \hat{Y}^{\min} \quad (36)$$

$$= w_x |\Delta X| + w_p |\Delta P|. \quad (37)$$

By degenerating the parameter interval (33) as $p \in [P_0, P_0]$, where P_0 is the parameter value, equation (37) becomes

$$|\Delta \hat{Y}'| = w_x |\Delta X|. \quad (38)$$

The ratio of the interval lengths is

$$\frac{|\Delta \hat{Y}'|}{|\Delta \hat{Y}|} = \frac{w_x |\Delta X|}{w_x |\Delta X| + w_p |\Delta P|}, \quad (39)$$

$$= 1 - \frac{w_p |\Delta P|}{w_x |\Delta X| + w_p |\Delta P|} \quad (40)$$

Assuming that the inputs have been normalized, which resulted in $|\Delta X| \approx |\Delta P|$, we have

$$\frac{|\Delta \hat{Y}'|}{|\Delta \hat{Y}|} \approx 1 - \frac{w_p}{w_x + w_p}. \quad (41)$$

Equation (41) can be extended to P parameter and X variable inputs as follows:

$$\frac{|\Delta \hat{Y}'|}{|\Delta \hat{Y}|} \approx 1 - \frac{\sum_p w_p}{\sum_x w_x + \sum_p w_p}, \quad (42)$$

which proves that the proposed method is at worst equivalent to the state-of-the-art since the weights have been assumed positive. The improvement in the bound tightness depends on the ratio of the sum of parameter input weights and the sum of all weights. \square

The bounds on y are

$$y \in [\max(0, \hat{Y}^{\min}), \max(0, \hat{Y}^{\max})]. \quad (43)$$

³If a weight is negative, the lower bound of the input associated to that weight must be used in (34) and the upper bound in (35).

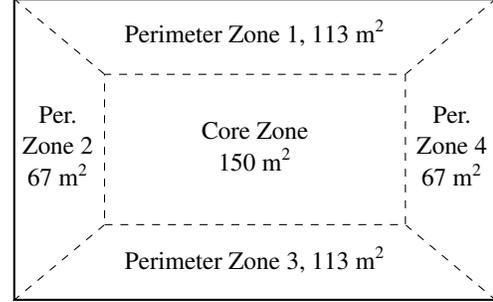


Fig. 4: Layout of the building floor.

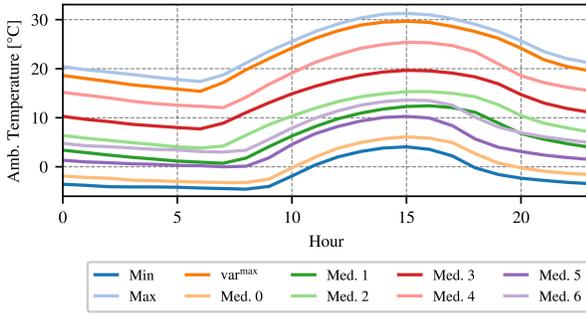
III. CASE STUDY

We analyze the effectiveness of the proposed method on a realistic office building comprising five zones. We start in Section III-A by describing the high-quality and publicly available building model, datasets, and tools used to design the case study, ensuring a fair comparison of results and reproducibility. Then, Section III-B presents the different benchmarks to compare the performance of our DFL-based training procedure. In Section III-C, we report the performance of all the models learned in a DFL-fashion. Finally, in Section III-D, we further compare our DFL approach to the corresponding Identify-Then-Optimize (ITO) baseline, where the building's thermal dynamics are learned using a standard statistical loss function independent of the optimization task. Our code is publicly available at https://github.com/PSMRB/dfh_hvac_management.

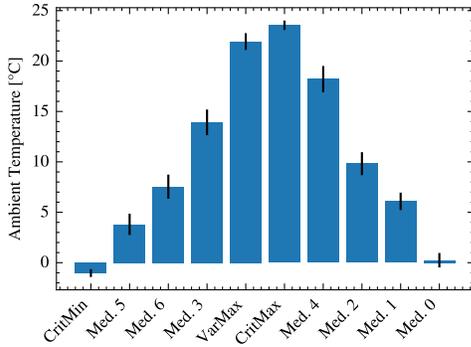
A. Data

The US Department of Energy has developed EnergyPlus, a high-fidelity physics-based simulator for building energy modeling, continuously updated since 2001 [12]. EnergyPlus includes publicly available building models [56]. We selected an office building with five actively controlled thermal zones. The zones are located over one floor of 511 m². Figure 4 provides an illustration of the floor layout. Each zone is equipped with its own air-to-air heat pump, such that the HVAC system of each zone is fully independent. The only alteration to the building model is the replacement of the gas-fired heating coil by an electric coil of equal power.

The building is assumed to be located in Denver, Colorado. To model building thermal dynamics, we utilize two typical meteorological datasets, each representing a full year of hourly weather data for Denver International Airport, derived from long-term observations. The first dataset is used to simulate one year of building operation using EnergyPlus's default heuristic control (i.e., without optimization), while the sec-



(a) Ambient temperature profiles of the medoids.



(b) Mean and standard deviation of the medoids ordered to form a smooth cycle.

Fig. 5: Cluster analysis.

and provides weather scenarios used to evaluate optimized scheduling strategies.

Optimizing then simulating daily schedules for all 365 days is computationally demanding. Moreover, there exist days with similar weather that bring very little extra information to the learning process. Therefore, we apply k -medoids clustering to group similar weather conditions from the second dataset. The k -medoids algorithm is chosen over k -means to prevent the generation of synthetic average data. As ambient temperature is the only meteorological input required for the model (2), we use it as the primary clustering criterion. Initially, we select three fixed medoids corresponding to the most extreme conditions: the coldest day, the hottest day, and the day with the highest temperature variability. We then determine seven additional medoids to partition the rest of the dataset. The resulting temperature profiles from the k -medoids clustering are illustrated in Figure 5a, with the cluster means and standard deviations summarized in Figure 5b. In total, only ten carefully selected days are used to train the model that should generalize over the whole year.

To reflect typical load patterns and incentivize off-peak usage, we adopt a time-of-use electricity tariff: 0.6 \$/kWh from 6 a.m. to 7 p.m., the rate is double the base rate at 0.6 \$/kWh, and 0.3 \$/kWh during off-peak hours.

B. Models and Benchmarks

Our goal is to learn the parameters θ of a piecewise linear NN, i.e., constraint (2), representing the thermal model of the building. We learn the parameters in a decision-focused

fashion using gradient descent. To that end, we coded in *Python* and used *PyTorch* for computing the backpropagation. All computations were performed on an Intel Skylake 16-core Xeon 6142 processors and 16GB of RAM. To perform the gradient descent, we use *Adam* optimizer with an initial learning rate at 10^{-3} (except for the RC model where 10^{-2} was found to be better). The learning rate decays exponentially with $\gamma = 0.98$. We limit the number of epochs (i.e., the number of times the 10 scenarios are run) to 100 for the NNs, 150 for the RC model, and set an early stopping with a patience of 15.

We pre-train the NNs in a supervised way using the first dataset (i.e., one year of building operation with default heuristic control) to capture the building's main thermal characteristics. The resulting pre-trained NNs are then trained in a DFL fashion to learn additional and more complex relationships. This is achieved using SS.

SS learns the parameters of the NN-based thermal model embedded within an MIQP by introducing controlled random perturbations, which smooth the loss landscape and allow for gradient-based optimization. The resulting gradient is computed using the REINFORCE algorithm which enables bypassing the differentiation of the HVAC MS and EnergyPlus. We take a single sample $S = 1$ for the Monte Carlo approximation following conventional practice in the literature [39] and allowing practical implementation with a real-life system where only one outcome can be observed. We investigated five Gaussian noises of various intensity with standard deviations of 0.01, 0.05, 0.1, $\mu/10$, $\mu/2$ to perturb the parameters. The value of σ can be fixed or variable.

Our proposed DFL-based approach using SS is compared to other state-of-the-art DFL-based methods. These include (i) a QP relaxation of the original optimization problem where the binaries are relaxed as continuous variables between 0 and 1 and (ii) a two-step approach that first fixes the integer variables, then solves the resulting continuous and differentiable QP problem. *Cvxpylayer* computes the gradient of QP problems that are solved with *ECOS* since it has the best performance for conic programs [35]. However, the MIQP is solved using *Gurobi*.

Because EnergyPlus is not differentiable, the QP relaxation and the two-step approach cannot directly optimize the ex-post value (21). Therefore, [57] introduced a *supervised DFL loss* that bypasses the need to differentiate EnergyPlus (or the real installation):

$$\mathcal{L}_{\text{DFL}} = \sum_{t=0}^T \frac{\tilde{\lambda}_t}{T} \left(|p_t^{\text{hvac}} - \tilde{p}_t^{\text{hvac}}| + \text{MAE}(p_{t,z}^{\text{hvac}}) \right), \quad (44)$$

where p_t^{hvac} is the total HVAC consumption of the building at time step t , and $\text{MAE}(p_{t,z}^{\text{hvac}}) = \frac{1}{Z} \sum_{z=0}^Z |p_{t,z}^{\text{hvac}} - \tilde{p}_{t,z}^{\text{hvac}}|$.

Regarding the NN architectures, we learn the parameters of three fully connected feedforward NNs with six inputs: the temperature in each zone (5 inputs) and the outdoor temperature. The inputs are normalized. The three architectures have one hidden layer with 2 (NN1), 5 (NN2), and 10 ReLU (NN3). The last layer of all architectures is a linear layer with five outputs: the zonal temperatures at the next time step. Each

combination of architecture and DFL method is run over five seeds, and the best outcome is reported.

Finally, we also compare the NNs to a lumped RC model, a linear model of the building based on the aggregated resistance and capacitance of each zone. For each zone, the RC model turns constraint (2) into:

$$\tau_{z,t+1}^{\text{in}} = \tau_{z,t}^{\text{in}} + \left(\frac{\eta_z^{\text{h}} p_{z,t}^{\text{h}} - \eta_z^{\text{c}} p_{z,t}^{\text{c}}}{C_z} + \frac{\tau_{z,t}^{\text{amb}} - \tau_{z,t}^{\text{in}}}{R_z C_z} \right) \Delta t. \quad (45)$$

Table II gathers the hyperparameters used in this case study. For more detail, you can visit the GitHub associated with the paper.

TABLE II: Hyperparameters.

Optimizer	Adam
Learning Rate	10^{-3} for NN 0.02 for RC
Exponential Decay	0.98
Max. Nb. Epoch	100 (150 for RC)
Patience	15
SS Noise (σ)	0.01, 0.05, 0.1, $\mu/10$, $\mu/10$
S	1
NN1	1 layer, 2 ReLU
NN2	1 layer, 5 ReLU
NN3	1 layer, 10 ReLU

C. DFL Results

Table III reports the results of the various DFL-based models. For the two smaller NN architectures (NN1 & NN2), the three DFL methods—i.e., SS, QP-relaxation (QP), and QP subproblem with fixed binaries (FB)—converge successfully. For the most complex architecture, NN3, only QP training is feasible within 24 hours. Indeed, the complexity of the HVAC MS increases with the number of neurons. Consequently, since the SS and FB require to solve the MIQP for each sample, the training time quickly becomes computationally intensive.

The first two rows, *Ex-post+* and Hierarchical Loss, are the metrics used for training. SS can directly minimize *Ex-post+* (i.e., ex-post power cost, thermal discomfort penalty, and power cost misestimation), whereas QP and FB methods minimize the error on the HVAC power through the hierarchical loss (44).

SS obtains the best ex-post value compared to QP, and FB. For NN1, SS ($\sigma = 0.01$) obtains the lowest *Ex-post+* at \$94. FB and QP follow at \$113 and \$130, respectively. All three NN1 models have an ex-post cost between \$41 and \$41.5. However, the model accuracy to predict the cost varies. SS is the most accurate model with a daily cost underestimation of \$3.7. QP and FB follow with an overestimation of \$4.69 and \$4.74, respectively. Regarding the temperature penalties, SS comes first with only \$31 of daily penalty compared to \$41 for FB, and \$57 for QP. The total computational time (training, validation, and test) ranges from 95 minutes for QP to 136 minutes for SS.

For NN2, the same ranking is observed as for NN1. SS ($\sigma = 0.01$) outperforms with an *Ex-post+* value at \$84. This is even better than \$94 for NN1. FB and QP follow with \$101 (\$113 for NN1) and \$182 (\$130 for NN1). The ex-post cost ranges from \$41.8 for SS to \$42.7 for FB. The cost

estimation error is \$1.95 for SS, but rises to \$6.17 for QP. The temperature penalty goes from \$35 for SS to \$59 and \$77 for QP and FB, respectively. In conclusion, SS offers for both architectures the best thermal comfort and with excellent estimate of the power cost along with very competitive ex-post cost and computational burden.

FB outperforms QP on almost all training and ex-post metrics. For NN1, FB has a better hierarchical loss (12.8) and *Ex-post+* (\$113) than QP (hierarchical loss at 14.3 and *Ex-post+* at \$130). Even though FB has a similar ex-post cost (\$41.0 versus \$41.1 for QP) and cost error (\$4.74 versus \$4.69 for QP), FB provides more thermal comfort (temperature penalty at \$41 versus \$57 for QP). FB and QP take advantage of NN2's stronger modeling abilities to improve their training loss (FB: 11.4 vs. 12.8; QP: 12.4 vs. 14.3)). However, FB *Ex-post+* falls to \$101 reflecting better ex-post metrics whereas it increases to \$182 for QP. These results show (i) the inherent misalignment between the hierarchical loss and the ex-post value; (ii) the poor quality of the gradient obtained by relaxing the problem.

The poor gradient approximation generated by QP even undermines the performance of larger architectures such as NN3, causing them to underperform relative to smaller SS-trained architectures. Note that for NN3, the QP relaxation is also used for validation. Since the complexity of the HVAC MS grows with the number of neurons, SS and FB become too cumbersome. Indeed, FB and SS require to solve the MIQP at each sample. QP hierarchical loss is worse for NN3 than for NN2 going up from 12.4 to 13.2, while *Ex-post+* goes down from \$182 to \$125. The ex-post cost stands at \$43.2, slightly above the usual range. The cost overestimation (\$2.22) and the temperature penalty (\$58) become competitive. However, such a large NN with ten ReLU is long to train (almost four hours) and is still outperformed in terms of *Ex-post+* value by NN1 trained by SS. Moreover, solving the MIQP with NN3 at test time is about ten times longer than with NN1.

In contrast, since the RC model is linear, the resulting HVAC MS is inherently a continuous QP problem. Nevertheless, the RC model cannot capture the complexity of the thermal dynamics and exhibits the worst metrics among all models. The RC model is trained to minimize the hierarchical loss, which converges at 24.7 and *Ex-post+* at \$252. Regarding the ex-post metrics, the average daily ex-post cost of HVAC power is \$38.0, but it is largely underestimated at \$32.3. The average daily temperature penalty, reflecting the thermal discomfort, is \$150. Therefore, the ex-post cost is the lowest of all models because the RC model fails to provide comfortable temperatures. The total computational time for training, validation, and test is about seven hours.

In conclusion, SS outperforms DFL approaches that differentiate through the optimization problem (QP or FB), particularly when the underlying system or its simulator is non-differentiable. Unlike QP and FB, SS bypasses both the differentiation of the optimization layer and the system model, allowing the direct use of the true ex-post value (*Ex-post+*) as a training signal instead of relying on surrogate task losses. This results in superior gradient estimates and more informative training loss, enabling smaller models to outperform

TABLE III: Results of the decision-focused learning models: NN with one layer of two ReLU (NN1), five ReLU (NN2), ten ReLU (NN3), and one multi-zone RC model (RC). Three methods make the MIQP problem differentiable: fixing the binaries to their optimal value (FB), relaxing the integer constraint (QP), or applying a stochastic smoothing (SS).

	RC	SS	NN1 QP	FB	SS	NN2 QP	FB	NN3 QP
Ex-post+ (\$)	252	94	130	113	84	182	101	125
Hierarchical loss	24.7	20.4	14.3	12.8	22.2	12.4	11.4	13.2
MAE (kW)	0.6	0.52	0.55	0.55	0.5	0.54	0.54	0.54
MSE (kW ²)	0.44	0.35	0.37	0.39	0.34	0.39	0.39	0.39
Error mean (kW)	-0.07	-0.08	0.07	0.04	-0.01	0.02	0.01	0.03
Error std (kW)	0.39	0.42	0.37	0.36	0.45	0.37	0.39	0.36
Expected cost (\$)	32.3	37.9	45.8	45.8	39.9	47.4	43.1	45.4
Ex-post cost (\$)	38.0	41.5	41.1	41.0	41.8	41.2	42.7	43.2
Cost error (\$)	5.74	3.64	-4.69	-4.74	1.95	-6.17	-0.43	-2.22
Temp. Penalty(\$)	162	31	57	41	35	59	47	58
Nb. Epochs	150	71	36	54	43	25	94	100
Training time	03:30:24	01:08:34	00:48:59	00:56:49	04:17:05	00:49:33	07:19:17	04:25:18
Validation time	03:14:32	01:07:13	00:44:01	00:52:38	04:22:52	00:59:21	06:16:23	03:54:35 ⁴
Test time	00:00:46	00:00:49	00:01:51	00:00:57	00:07:59	00:01:57	00:01:05	00:10:13

⁴Validation conducted on the continuous relaxation (i.e., QP-relaxation) of the problem.

TABLE IV: Comparison of the performance of the Identify-Then-Optimize approach to decision-focused learning via Stochastic Smoothing (SS) for each model—NN with one layer of two ReLU neurons (NN1), five ReLU neurons (NN2), ten ReLU neurons (NN3), and one multi-zone RC model (RC).

	RC		NN1		NN2		NN3	
	ITO	DFL	ITO	SS	ITO	SS	ITO	QP
Ex-post+ (\$)	516	252	318	94	495	84	579	125
Hierarchical loss	54.2	24.7	35.2	20.4	43.7	22.2	39.4	13.2
MAE (kW)	0.59	0.6	0.5	0.52	0.51	0.5	0.53	0.54
MSE (kW ²)	0.45	0.44	0.34	0.35	0.36	0.34	0.38	0.39
Error mean (kW)	-0.27	-0.07	-0.26	-0.08	-0.35	-0.01	-0.35	0.03
Error std (kW)	0.55	0.39	0.38	0.42	0.38	0.45	0.35	0.36
Expected cost (\$)	21.3	32.3	26.8	37.9	21.7	39.9	20.5	45.4
Ex-post cost (\$)	41.5	38.0	41.6	41.5	42.2	41.8	42.2	43.2
Cost error (\$)	20.2	5.7	14.7	3.6	20.5	2.0	21.7	-2.2
Temp. Penalty(\$)	55	162	32	31	17	35	16	58
Nb. Epochs	0	150	0	71	0	43	0	100
Training time	-	03:30:24	-	01:08:34	-	04:17:05	-	04:25:18
Validation time	-	03:14:32	-	01:07:13	-	04:22:52	-	03:54:35
Test time	00:01:12	00:00:46	00:01:46	00:00:49	00:02:17	00:07:59	02:15:50	00:10:13

larger QP-trained counterparts while reducing computational overhead at inference.

D. ITO Results

Table IV compares the results of the DFL-trained models with their ITO counterpart. ITO models are trained over one year of historical data by Mean Square Error (MSE) minimization. The trained NN is then reformulated as a constraint of the HVAC MS. The ITO and DFL models are tested over the same ten days.

All models see a significant improvement with DFL compared to ITO. The three NN models see major improvements in terms of *Ex-post+* and hierarchical loss. *Ex-post+* falls from \$318 to \$94 for NN1, from \$495 to \$84 for NN2, and from \$579 to \$125 for NN3. The cost errors of the NN1 and NN2 ITO models are \$14.7 and \$20.5, compared to \$3.6 and \$2.0 for the SS models. The QP DFL with NN3 reduces the cost error from \$21.7 to \$2.2. Similarly, DFL of the RC parameters improves the hierarchical loss from 54.2 to 24.7 and *Ex-post+* from \$516 to \$252. It shows the possible improvement DFL can bring even for simple models as already noted in [57].

Because of DFL, the RC model can outperform more complex ITO models such as NNs. The *Ex-post+* value of the RC model is \$252, much lower than \$318 for NN1, \$495 for NN2, and \$579 for NN3.

When surveying only ITO models, NN1 is the best architecture, which shows that when ITO is used, a more complex architecture may not yield better decisions. NN1 reports an *Ex-post+* ITO value at \$318 far below the RC model (\$516), NN2 (\$495), and NN3 (\$579).

In conclusion, DFL significantly improves the quality of the ex-post metrics reflecting better and more realistic decisions compared to the conventional two-stage ITO approach. A simple DFL-trained model, such as the RC model, is to be preferred to ITO-trained NNs. Nevertheless, the improvement brought by DFL is limited by the modeling power of the architecture, as shown for RC model, and the quality of gradient approximation, as shown for NN3. Caution is warranted when resorting to ITO: increasing model complexity does not necessarily yield better decisions and may introduce additional computational burden.

E. Effect of Constraint Tightness on Solving Time

TABLE V: Comparison of the solving time for the optimization during test (i.e., MIQP formulation) with and without the proposed tight formulation.

	NN1			NN2			NN3	Avg.
	SS	QP	FB	SS	QP	FB		
SOTA	39.0	23.4	23.0	41.5	72.6	211	1443	264.4
Tight	32.0	23.2	19.8	38.2	66.0	180	1422	254.0
Gain (%)	17.9	0.9	13.9	8.0	9.1	14.7	1.5	9.4

In subsection II-D, we analytically showed how to tighten the MIL equations of a ReLU by dynamically adjusting the feasible interval of its inputs. In this case study, the NNs have 11 inputs, among which one is a parameter (i.e., the ambient temperature) and ten are decision variables (i.e., the five zonal indoor temperatures and the related HVAC power consumption). Applying (31) and assuming that the weights have the same magnitude, the expected theoretical improvement in tightness is approximately 9.1%.

As shown in Table V, the tight formulation consistently reduces solving time across all test cases, with an average gain of 9.4%. This validates the theoretical prediction from (31) and demonstrates the practical advantage of our approach in terms of computational efficiency. Importantly, this improvement is achieved without sacrificing model accuracy or solution quality.

F. Noise intensity analysis

In this section, we analyze the impact of the noise intensity on the performance of SS-DFL. Since we use a Gaussian noise, we modify σ to investigate five level of noise: 0.01, 0.05, 0.1, $\mu/10$, and $\mu/2$. We further enable two modes for σ : it can either remain constant or be learned. Note that for $\mu/10$ and $\mu/2$, σ is either initialized as a fraction of μ and then kept constant, or evolves with μ . We focus on two key metrics⁵—the $Ex-post+$ loss and the number of epochs necessary to converge—shown in Figure 6.

The analysis of noise impact on SS-DFL training performance reveals that the $Ex-post+$ loss increases with the noise. This is true for both NN1 and NN2, whether σ is constant or variable. Interestingly, setting σ as a fraction of μ (i.e., σ is a fraction of the pre-training value found for each parameter) does not yield better results. Allowing σ to be trainable (variable configuration) does not improve the $Ex-post+$ loss. Unlike μ , σ has no clear optimal value. It is a variable without physical reality introduced by SS. Therefore, making σ a parameter to learn complexifies training without a clear gain. Ultimately, it results in a loss of performance.

In both variable and constant configurations, the best performance is achieved with the smallest constant noise ($\sigma = 0.01$), and initializing all the σ 's at the same value is to be preferred to fraction of μ .

There is no significant effect of σ on the training convergence speed, as measured by the number of epochs required. Additionally, whether σ is kept constant or treated as a trainable parameter does not appear to influence convergence

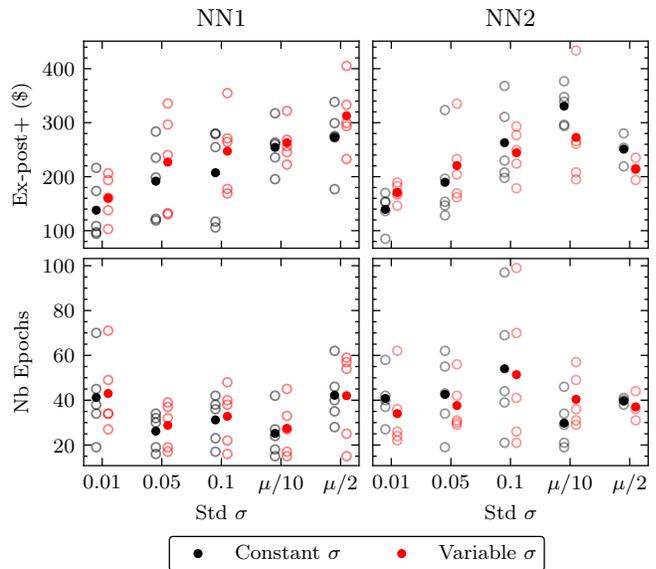


Fig. 6: Analysis of $Ex-post+$ and the number of epochs for various definition of σ . The mean of the five trainings is in bold. The $Ex-post+$ loss tends to increase with higher noises while no clear trend is observe on the number of epochs indicating little effect on the convergence rate.

rates. This behavior is consistent across both NN1 and NN2 architectures.

G. Number of Samples Analysis

We complement our results by analyzing the impact of the sample size parameter S on performance, noting that all previous experiments were conducted with $S = 1$. The motivation for $S = 1$ was threefold. First, it reflects practical deployment conditions in real buildings, where only a single true realization of the building's response is observable at each time step due to the absence of simulators. Second, the score-function (REINFORCE) gradient estimator remains unbiased regardless of S . Increasing S decreases the variance of the gradient estimate but increases computational cost per training instance, leading to longer training times. Third, this choice aligns with the reinforcement learning literature (and the advice of the SS-DFL authors [39]), where policy gradient methods typically use single-sample estimates at each update to balance computational efficiency and estimator variance. Therefore, fixing $S = 1$ provides a practical and theoretically sound approach for our decision-focused learning framework.

In Figure 7, we can see the $Ex-post+$ loss, the training time and the number of epochs for $S = 1, 2, 5, 10^6$. All trainings were performed with σ set to its optimal value (i.e., constant $\sigma = 0.01$). The $Ex-post+$ loss tends to increase with more samples for both NN1 and NN2. We believe this is because having more samples reduces the exploration during training. As expected, the number of epochs necessary for the training to converge decreases with S , reflecting the higher quality of the gradient estimation. This decrease is extremely steady for NN1, but less for NN2. Despite the number of epochs going down with S , the training time goes up as expected.

⁵The complete tables are available in the repository.

⁶The complete tables are available in the repository.

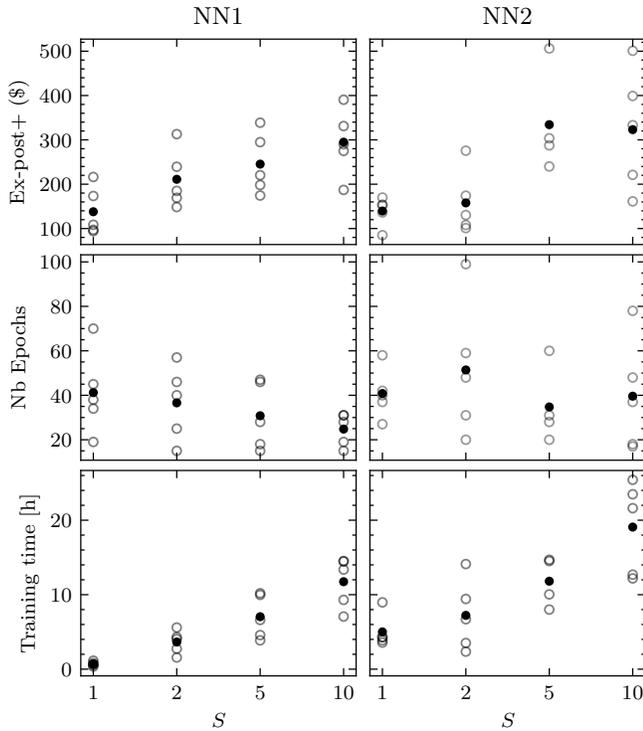


Fig. 7: Analysis of $Ex\text{-}post+$, the number of epochs, and the training time depending on the number of samples S . The mean of the five trainings is in bold. The $Ex\text{-}post+$ loss tends to increase with more samples. Despite the number of epochs going down with S , the training time goes up.

In conclusion, $S = 1$ leads to the best decision quality (i.e., minimal $Ex\text{-}post+$ loss) and training time.

IV. CONCLUSION

We presented an HVAC MS where the building thermal dynamics are modeled using NN. We started by improving the formulation of NN as constraints in optimization problem. Then, we learned the parameters of the NN using DFL. In order to ensure meaningful gradient and training robustness, the HVAC MS is formulated as an MIQP where the thermal comfort is a quadratic penalty in the objective function rather than a hard constraint. Since MIQP are discrete by nature, we employed SS to produce an informative gradient without having to differentiate the MIQP and the building (or its simulator).

We tested our approach on a realistic five-zone building. Results show that SS outperforms the conventional two-stage approach where NN are trained on historical data and then embedded into the optimization. We also showed that DFL with SS is to be preferred to more naive approaches that relax or fix the binaries of the mixed integer problem.

Further research is required to strengthen both scalability and robustness. On the modeling side, developing optimization formulations for entire neural networks, rather than individual neurons, and generating feasible distributions over large sets of constraint parameters would enhance stochastic smoothing and enable chance-constrained programming to address model uncertainty. On the computational side, software advances, such as parallelization, warm-starting optimization with its

closest neighbor, and multi-fidelity simulation, will further reduce computation times and support large-scale deployment.

REFERENCES

- [1] "Buildings - Energy System," Jul. 2023. [Online]. Available: <https://www.iea.org/energy-system/buildings>
- [2] "Energy consumption in households," Jun. 2024. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Energy_consumption_in_households
- [3] D. Papadaskalopoulos, G. Strbac, P. Mancarella, M. Aunedi, and V. Stanojevic, "Decentralized Participation of Flexible Demand in Electricity Markets—Part II: Application With Electric Vehicles and Heat Pump Systems," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 3667–3674, Nov. 2013. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6515366>
- [4] K. Kouzelis, Z. H. Tan, B. Bak-Jensen, J. R. Pillai, and E. Ritchie, "Estimation of Residential Heat Pump Consumption for Flexibility Market Applications," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1852–1864, Jul. 2015. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7079500>
- [5] E. Mocanu, P. H. Nguyen, and M. Gibescu, "Energy disaggregation for real-time building flexibility detection," in *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE, 2016, pp. 1–5.
- [6] G. Tian, Q. Z. Sun, and W. Wang, "Real-time flexibility quantification of a building HVAC system for peak demand reduction," *IEEE Transactions on Power Systems*, vol. 37, no. 5, pp. 3862–3874, 2021.
- [7] M. Yousefi, A. Hajizadeh, M. N. Soltani, and B. Hredzak, "Predictive Home Energy Management System With Photovoltaic Array, Heat Pump, and Plug-In Electric Vehicle," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 430–440, Jan. 2021.
- [8] Y.-J. Kim, E. Fuentes, and L. K. Norford, "Experimental Study of Grid Frequency Regulation Ancillary Service of a Variable Speed Heat Pump," *IEEE Transactions on Power Systems*, vol. 31, no. 4, pp. 3090–3099, Jul. 2016. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/7268771>
- [9] D. Wang, S. Parkinson, W. Miao, H. Jia, C. Crawford, and N. Djilali, "Online voltage security assessment considering comfort-constrained demand response control of distributed heat pump systems," *Smart Grids*, vol. 96, pp. 104–114, Aug. 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306261911007999>
- [10] "The current electricity market design in Europe," Jan. 2015. [Online]. Available: <https://set.kuleuven.be/ei/factsheets>
- [11] J. Drgoňa, J. Arroyo, I. Cupeiro Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter, D. L. Vrabie, and L. Helsen, "All you need to know about model predictive control for buildings," *Annual Reviews in Control*, vol. 50, pp. 190–232, Jan. 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1367578820300584>
- [12] D. Crawley, L. Lawrie, F. Winkelmann, W. Buhl, Y. Huang, C. Pedersen, R. Strand, R. Liesen, D. Fisher, M. Witte, and J. Glazer, "EnergyPlus: Creating a New-Generation Building Energy Simulation Program," *Energy and Buildings*, vol. 33, pp. 319–331, Apr. 2001.
- [13] Z. Afroz, Gm. Shafiqullah, T. Urmee, and G. Higgins, "Modeling techniques used in building HVAC control systems: A review," *Renewable & Sustainable Energy Reviews*, vol. 83, pp. 64–84, Dec. 2017.
- [14] D. D'Agostino, R. Landolfi, M. Nicoletta, and F. Minichiello, "Experimental Study on the Performance Decay of Thermal Insulation and Related Influence on Heating Energy Consumption in Buildings," *Sustainability*, vol. 14, no. 5, p. 2947, Jan. 2022. [Online]. Available: <https://www.mdpi.com/2071-1050/14/5/2947>
- [15] K. J. Kircher and K. Max Zhang, "On the lumped capacitance approximation accuracy in RC network building models," *Energy and Buildings*, vol. 108, pp. 454–462, Dec. 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378778815302930>
- [16] F. Belić, Ž. Hocenski, and D. Sliškočić, "Thermal modeling of buildings with RC method and parameter estimation," in *2016 International Conference on Smart Systems and Technologies (SST)*, Oct. 2016, pp. 19–25, journal Abbreviation: 2016 International Conference on Smart Systems and Technologies (SST).
- [17] G. Qiang, T. Zhe, D. Yan, and Z. Neng, "An improved office building cooling load prediction model based on multivariable linear regression," *Energy and Buildings*, vol. 107, pp. 445–455, Nov. 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378778815302255>
- [18] Y.-J. Kim, "A Supervised-Learning-Based Strategy for Optimal Demand Response of an HVAC System in a Multi-Zone Office Building," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4212–4226, Sep. 2020.

- [19] S. A. Nabavi, N. H. Motlagh, M. A. Zaidan, A. Aslani, and B. Zakeri, "Deep Learning in Energy Modeling: Application in Smart Buildings With Distributed Energy Generation," *IEEE Access*, vol. 9, pp. 125 439–125 461, 2021.
- [20] J. Drgoňa, A. R. Tuor, V. Chandan, and D. L. Vrabie, "Physics-constrained deep learning of multi-zone building thermal dynamics," *Energy and Buildings*, vol. 243, p. 110992, Jul. 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378778821002760>
- [21] D. Jang, L. Yan, L. Spangher, and C. J. Spanos, "Active Reinforcement Learning for Robust Building Control," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 20, pp. 22 150–22 158, Mar. 2024, section: AAAI Technical Track on AI for Social Impact Track. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/30219>
- [22] P. L. Donti, D. Rolnick, and J. Z. Kolter, "DC3: A learning method for optimization with hard constraints," Apr. 2021. [Online]. Available: <http://arxiv.org/abs/2104.12225>
- [23] B. Tang, E. B. Khalil, and J. Drgoňa, "Learning to Optimize for Mixed-Integer Non-linear Programming with Feasibility Guarantees," May 2025, arXiv:2410.11061 [cs]. [Online]. Available: <http://arxiv.org/abs/2410.11061>
- [24] S. Park and P. V. Hentenryck, "Self-Supervised Primal-Dual Learning for Constrained Optimization," Nov. 2022, arXiv:2208.09046 [cs]. [Online]. Available: <http://arxiv.org/abs/2208.09046>
- [25] F. Fioretto, T. W. K. Mak, and P. V. Hentenryck, "Predicting AC Optimal Power Flows: Combining Deep Learning and Lagrangian Dual Methods," Dec. 2019, arXiv:1909.10461 [eess, math]. [Online]. Available: <http://arxiv.org/abs/1909.10461>
- [26] I. Murzakanov, A. Venzke, G. S. Misyris, and S. Chatzivasileiadis, "Neural Networks for Encoding Dynamic Security-Constrained Optimal Power Flow," *arXiv:2003.07939 [cs, eess, math]*, Oct. 2021. [Online]. Available: <http://arxiv.org/abs/2003.07939>
- [27] R. Li, H. Zhang, M. Sun, F. Teng, C. Wan, S. Pineda, and G. Kariniotakis, "Decision-Oriented Learning for Future Power System Decision-Making under Uncertainty," Apr. 2024. [Online]. Available: <http://arxiv.org/abs/2401.03680>
- [28] J. Mandi, J. Kotary, S. Berden, M. Mulamba, V. Bucarey, T. Guns, and F. Fioretto, "Decision-Focused Learning: Foundations, State of the Art, Benchmark and Future Opportunities," Aug. 2023. [Online]. Available: <http://arxiv.org/abs/2307.13565>
- [29] Y.-H. Kao, B. V. Roy, and X. Yan, "Directed Regression," in *Neural Information Processing Systems*, Dec. 2009. [Online]. Available: <https://www.semanticscholar.org/paper/Directed-Regression-Kao-Roy/76efadb1274df410d7060ca0b184c4dba354e96c>
- [30] P. L. Donti, B. Amos, and J. Z. Kolter, "Task-based end-to-end model learning in stochastic optimization," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., Dec. 2017, pp. 5490–5500.
- [31] A. Agrawal, S. Barratt, S. Boyd, and B. Stellato, "Learning Convex Optimization Control Policies," in *Proceedings of the 2nd Conference on Learning for Dynamics and Control*. PMLR, Jul. 2020, pp. 361–373. [Online]. Available: <https://proceedings.mlr.press/v120/agrawal20a.html>
- [32] B. Amos and J. Z. Kolter, "OptNet: Differentiable Optimization as a Layer in Neural Networks," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, Jul. 2017, pp. 136–145, ISSN: 2640-3498. [Online]. Available: <https://proceedings.mlr.press/v70/amos17a.html>
- [33] B. Chen, Z. Cai, and M. Bergés, "Gnu-RL: A Precocious Reinforcement Learning Solution for Building HVAC Control Using a Differentiable MPC Policy," in *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, ser. BuildSys '19. New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 316–325. [Online]. Available: <https://dl.acm.org/doi/10.1145/3360322.3360849>
- [34] X. Cui, J.-F. Toubeau, F. Vallée, and Y. Wang, "Decision-Oriented Modeling of Thermal Dynamics Within Buildings," *IEEE Transactions on Smart Grid*, vol. 16, no. 1, pp. 369–382, Jan. 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10638763/>
- [35] D. Wahdany, C. Schmitt, and J. L. Cremer, "More than accuracy: End-to-end wind power forecasting that optimises the energy system," *Electric Power Systems Research*, vol. 221, p. 109384, Aug. 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0378779623002730>
- [36] L. Ortmann, F. Böhm, F. Klein-Helmkamp, A. Ulbig, S. Bolognani, and F. Dörfler, "Tuning and Testing an Online Feedback Optimization Controller to Provide Curative Distribution Grid Flexibility," Mar. 2024. [Online]. Available: <http://arxiv.org/abs/2403.01782>
- [37] A. Ferber, B. Wilder, B. Dilkina, and M. Tambe, "MIPaaL: Mixed Integer Program as a Layer," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 02, pp. 1504–1511, Apr. 2020. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/5509>
- [38] B. Tang and E. B. Khalil, "PyEPO: A PyTorch-based End-to-End Predict-then-Optimize Library for Linear and Integer Programming," Apr. 2023, arXiv:2206.14234 [math]. [Online]. Available: <http://arxiv.org/abs/2206.14234>
- [39] M. Silvestri, S. Berden, J. Mandi, A. I. Mahmutoğulları, B. Amos, T. Guns, and M. Lombardi, "Score Function Gradient Estimation to Widen the Applicability of Decision-Focused Learning," Jun. 2024. [Online]. Available: <http://arxiv.org/abs/2307.05213>
- [40] F. Ceccon, J. Jalving, J. Haddad, A. Thebelt, C. Tsay, C. D. Laird, and R. Misener, "OMLT: Optimization & Machine Learning Toolkit," Nov. 2022. [Online]. Available: <http://arxiv.org/abs/2202.02414>
- [41] S. Zhang, J. S. Campos, C. Feldmann, F. Sandfort, M. Mathea, and R. Misener, "Augmenting optimization-based molecular design with graph neural networks," *Computers & Chemical Engineering*, vol. 186, p. 108684, Jul. 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0098135424001029>
- [42] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte carlo gradient estimation in machine learning," *Journal of Machine Learning Research*, vol. 21, no. 132, pp. 1–62, 2020.
- [43] C. Baldassi, E. M. Malatesta, and R. Zecchina, "Properties of the Geometry of Solutions and Capacity of Multilayer Neural Networks with Rectified Linear Unit Activations," *Physical review letters*, vol. 123, no. 17, pp. 1–170 602, 2019.
- [44] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *Journal of machine learning research*, vol. 15, pp. 315–323, 2011.
- [45] R. Balestrieri and R. G. Baraniuk, "Mad Max: Affine Spline Insights Into Deep Learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 704–727, May 2021.
- [46] E. Balas, "Disjunctive programming," *Annals of discrete mathematics*, vol. 5, pp. 3–51, 1979.
- [47] R. R. Bunel, I. Turkaslan, P. Torr, P. Kohli, and P. K. Mudigonda, "A Unified View of Piecewise Linear Neural Network Verification," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2018/hash/be53d253d6bc3258a8160556dda3e9b2-Abstract.html
- [48] R. Anderson, J. Huchette, W. Ma, C. Tjandraatmadja, and J. P. Vielma, "Strong mixed-integer programming formulations for trained neural networks," *Mathematical Programming*, vol. 183, no. 1, pp. 3–39, Sep. 2020. [Online]. Available: <https://doi.org/10.1007/s10107-020-01474-5>
- [49] C. Tsay, J. Kronqvist, A. Thebelt, and R. Misener, "Partition-Based Formulations for Mixed-Integer Optimization of Trained ReLU Neural Networks," in *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc., 2021, pp. 3068–3080. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2021/hash/17f98ddf040204eda0af36a108cbdea4-Abstract.html
- [50] A. Alcántara and C. Ruiz, "A neural network-based distributional constraint learning methodology for mixed-integer stochastic optimization," *Expert Systems with Applications*, vol. 232, p. 120895, Dec. 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0957417423013970>
- [51] D. Kenefake, R. Kakaodkar, M. Ali, and E. N. Pistikopoulos, "A novel neural network bounds-tightening procedure for multiparametric programming and control," *Computer Aided Chemical Engineering*, vol. 52, pp. 1841–1846, Jan. 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780443152740502924>
- [52] J. Zhou, Y. T. Chu, J. Ren, W. Shen, and C. He, "Integrating machine learning and mathematical programming for efficient optimization of operating conditions in organic Rankine cycle (ORC) based combined systems," *Energy*, vol. 281, p. 128218, Oct. 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360544223016122>
- [53] P. Favaro, M. Dolányi, F. Vallée, and J.-F. Toubeau, "Neural network informed day-ahead scheduling of pumped hydro energy storage," *Energy*, vol. 289, p. 129999, Feb. 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360544223033935>
- [54] V. Tjeng, K. Xiao, and R. Tedrake, "Evaluating Robustness of Neural Networks with Mixed Integer Programming," Feb. 2019. [Online]. Available: <http://arxiv.org/abs/1711.07356>

- [55] R. E. Moore, R. B. Kearfott, and M. J. Cloud, *Introduction to Interval Analysis*. Society for Industrial and Applied Mathematics, Jan. 2009. [Online]. Available: <http://epubs.siam.org/doi/book/10.1137/1.9780898717716>
- [56] DOE and PNNL, “Commercial Prototype Building Models,” 2023. [Online]. Available: https://www.energycodes.gov/development/commercial/prototype_models
- [57] P. Favaro, J.-F. Toubeau, F. Vallée, and Y. Dvorkin, “Decision-Focused Learning for Complex System Identification: HVAC Management System Application,” in *Proceedings of the 16th ACM International Conference on Future and Sustainable Energy Systems*, ser. E-Energy '25. New York, NY, USA: Association for Computing Machinery, Jun. 2025, pp. 347–358. [Online]. Available: <https://dl.acm.org/doi/10.1145/3679240.3734584>



Pietro Favaro received his Master’s degree in Electrical Engineering from the University of Mons, Belgium, in 2022, alongside a Master’s degree in Smart Cities and Communities from Heriot-Watt University, Scotland. He is currently pursuing a PhD at the University of Mons as an FRS-FNRS Fellow. In 2023, he was awarded a one-year fellowship from the Belgian-American Educational Foundation he spent at Johns Hopkins University. His research focuses on the scheduling of complex flexible assets, including pumped hydro energy storage and build-

ings.



Jean-François Toubeau (Member, IEEE) received the degree in civil electrical engineering and the Ph.D. degree in electrical engineering from the University of Mons, Belgium, in 2013 and 2018, respectively, where he is currently a Full-Time Senior Researcher within the “Power Systems and Markets Research Group.” His research mainly focuses on bridging the gap between machine learning and decision-making in modern power systems.



François Vallée received the degree in civil electrical engineering and the Ph.D. degree in electrical engineering from the Faculty of Engineering, University of Mons, Belgium, in 2003 and 2009, respectively, where he is currently a Full Professor and the Leader of the “Power Systems and Markets Research Group.” His research interests include PV and wind generation modeling for electrical system reliability studies in presence of dispersed generation, and adequacy studies. His Ph.D. work has been awarded the SRBE/KBVE Robert Sinave Award in

2010. He is currently serving as an Associate Editor for IEEE Transactions on Power Systems.

Yury Dvorkin received his Ph.D. degree in Electrical Engineering from the University of Washington in 2016. He is currently an Associate Professor at Johns Hopkins University. His research focuses on modeling and algorithmic solutions for an efficient, reliable, and resilient energy transition.