Enhancing Dynamic Control of Inertial District Energy Networks Through a Physics-Informed State-Space Model

Taha Boussaid[1,2] (✉), François Rousset[1], Marc Clausse[1], and Vasile-Marian Scuturici[2]

[1] INSA Lyon, CNRS, CETHIL, UMR 5008, Villeurbanne, 69100, France
{taha.boussaid}@insa-lyon.fr
[2] INSA Lyon, CNRS, LIRIS, UMR 5205, Villeurbanne, 69100, France

Abstract. Control optimization is essential to achieve high performance and cost efficiency in large-scale physical systems such as inertial district energy networks. These systems play a key role in mitigating climate change, particularly in heating and cooling. They integrate multiple energy sources and require intelligent control strategies to minimize costs while preserving high efficiency. However, the complexity of their underlying dynamics and the high computational load associated with their numerical simulation often make predictive control prohibitively slow or limited to short time horizons. In this work, we introduce a hybrid modeling framework where predictive control is accelerated using a physics-informed spatio-temporal graph neural network as a state-space surrogate model. Unlike existing models, our approach incorporates first-principle conservation laws to improve accuracy and generalization. This approach drastically reduces simulation time by four orders of magnitude, enabling faster decision-making. Using real-world data, we introduce a time-series augmentation technique combining Gaussian scaling and time slicing to improve model performance. Extensive experiments were conducted to evaluate the accuracy and generalization capacity of the learned model. Once validated, several optimization techniques were implemented, including evolutionary algorithms and reinforcement learning, which are assessed against rule-based control. Results show that this approach enables scalable predictions and efficient control, achieving up to 29% energy cost savings during mid-season while cutting optimization time from days to mere minutes.

Keywords: Predictive Control · Surrogate Modeling · Sustainable Energy Systems

1 Introduction

Mitigating climate change requires substantial reduction in greenhouse gas emissions [27]. To do so, the international energy agency (IEA) outlines the need to deploy large energy networks with multiple low-carbon-footprint energy sources

to reach net-zero emissions by 2050 [21]. District heating networks are an example of such large inertial energy networks infrastructure [2]. The term 'inertial' refers to the system's thermal inertia, where the thermal mass of pipes and storage induces a time delay between heat production and demand. These networks use simultaneously various renewable energy sources such as biomass, geothermal, solar thermal in addition to thermal energy storage (TES). Incorporating an increasing number of energy sources requires rethinking smart control strategies to ensure efficient system deployment and achieve sustainable energy transition. Nevertheless, the different dynamics of energy sources (non-linearities, response time, intermittence etc.) bring new complexity to numerical simulation, which then makes the optimization of such systems prohibitively time-consuming, ranging from several hours [15] to days [13], using traditional approaches such as mixed integer non-linear programming (MINLP).
To address this limitation, recent works have used deep learning models, known for their fast inference and strong approximation capabilities in capturing complex dependencies and dynamics [10]. This technique was applied to diverse types of dynamical systems such as climate forecasting [33], thermal and electrical load forecasting [35, 12], fluid dynamics and electromagnetics [8], among others.
However, applying deep learning to physical systems presents some challenges. Many studies rely on benchmark datasets with high-frequency sampling (e.g., 1 ms or 4 s) and simplified dynamics, often defined by few state variables or initial conditions [36, 25]. In contrast, real-world systems are rarely monitored at such fine time steps and depend on numerous state variables and external inputs, such as weather disturbances. Moreover, while some real-world applications exist, no systematic methodology or design framework has been established for inertial district energy networks, in contrast to electric grids [40].

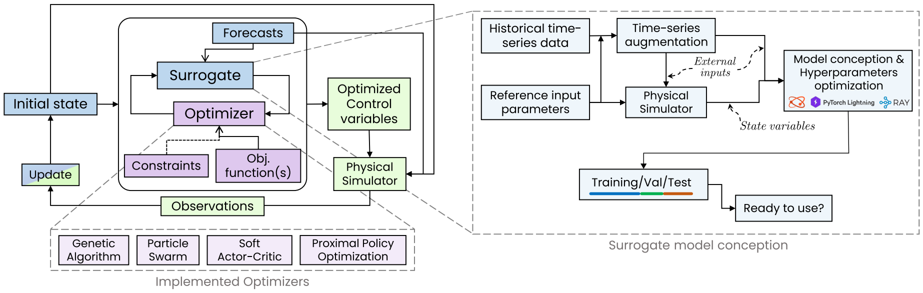


Fig. 1: Proposed methodology: Left block shows neural model predictive control using a validated surrogate model with evolutionary and RL optimizers. Right block schematizes surrogate model design, training, and validation pipeline.

In this work, we propose a hybrid strategy for control optimization of inertial district energy networks. The developed approach, schematized in Fig. 1, leverages the graph representation of such systems to develop an appropriate physics-informed spatio-temporal graph neural network (PI-STGCN). Unlike prior research, our proposition can handle various energy sources at different locations along with multiple consumer nodes to learn a state-space representation between these entities. The surrogate model development pipeline, shown in the right block of Fig. 1, relies on hyperparameters optimization and historical time-series augmentation during the learning phase. We demonstrate that, in addition to expanding the dataset size, the latter technique enables the incorporation of physically plausible scenarios into the training set. The effectiveness of our strategy is demonstrated through its application to a real-world system combining an inertial energy source (biomass) and an intermittent source (solar). The primary contributions can be summarized as follows:

- We introduce a system-agnostic methodology to accelerate optimal control of inertial district energy systems, leveraging their graph topology to model diverse producer and consumer types.
- We adapt Gaussian scaling and time warping to augment time-series data, exposing the model to plausible training scenarios. Furthermore, integrating a generic first-principle conservation equation enhances predictive confidence, resulting in a physics-informed model architecture.
- The proposed state-space surrogate is validated through extensive experiments and benchmarks. Scalability is demonstrated using a synthetic dataset, that is made publicly available. In addition, the optimization is carried out with several optimizers to assess their performance and the resulting control strategies.
- Applied to a real-world system with multiple energy sources and stringent constraints (e.g., power ramps, minimum time-on/off), our method achieves up to 29% energy cost reductions and reduces computational time by up to four orders of magnitude.

2 Related Work

Model predictive control (MPC), as schematized in Fig. 1 (left block), requires an accurate system model to perform predictive simulations. The control algorithm must accurately model and predict the system's behavior under various control scenarios. In control theory, this dynamical model is often expressed in a state-space, where the dynamics follow an ordinary differential equation (ODE) [7]. An optimal control problem is mathematically formulated for an optimization time-horizon $\mathcal{H}^{\text{opt}} = [0,\ t_f]$ as follows:

$$\begin{aligned} \frac{dx\,(t)}{dt} &= f\left(x(t),\ u(t),\ d(t)\right), \quad \text{and} \quad x(0) = x_0, \\ C\left(t_f,\ u\right) &= \int_0^{t_f} g\left(t,\ x_u(t),\ u(t)\right) dt + h(t_f, x_u(t_f)), \\ \text{s.t.} \quad x_u(t) &= \text{argmin}\ \ C(t_f, u) \quad \text{and} \quad l\left(t,\ x(t),\ u(t)\right) \le 0 \ \ \forall t. \end{aligned} \tag{1}$$

where f represents the non-linear system dynamics, $x \in \mathbb{R}^{n_x}$ is the vector of state variables, $u \in \mathbb{R}^{n_u}$ and $d \in \mathbb{R}^{n_d}$ are the vectors of control variables and external disturbances respectively. The cost function C is composed of a running cost g and a terminal cost h evaluated at $t = t_f$, the end of the optimization horizon $\mathcal{H}^{\text{opt}}$. In addition, state and control variables must satisfy a set of constraints l along the optimization horizon. State-space models can be learned in two distinct ways, discrete-time (DT) or continuous-time (CT) models [3, 4]. DT models are more common and easier to construct as data is represented via discrete elements (arrays, vectors, etc.).

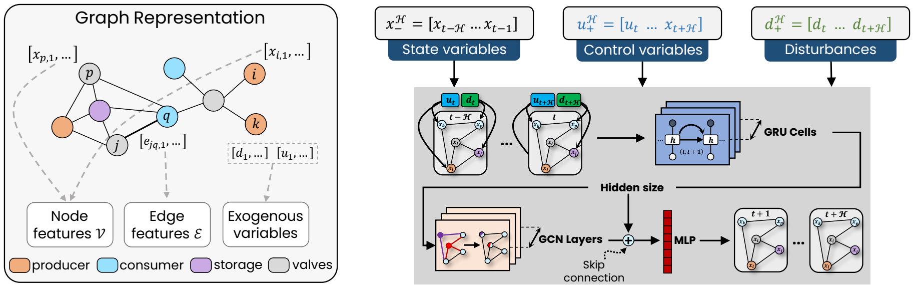
In the field of inertial district energy systems, a number of studies proposed surrogate DT models [29, 9, 19, 20]. For example, the authors of [19] proposed to associate a recurrent neural network (RNN) to each consumer node in a district heating network (DHN). However, they only considered a single producer network, and the surrogate model conception relies on creating and connecting RNN cells. Graph neural networks (GNNs) encode topological features as inductive bias, as in [9], where a graph attention-based model was used to accelerate dynamic simulations by 1 to 2 orders of magnitude. However, no physical constraints were included, and only single-producer networks were studied. Other studies [20, 29] proposed control strategies for district energy systems in which heat load forecasts are generated by GNN-based models, while the physical system still relied on a physical simulation which is computationally expensive. In terms of optimization techniques, evolutionary algorithms such as genetic algorithms (GA) are widely used to handle complex optimizations and have proven to be a reliable technique for inertial district energy systems optimization [39, 34]. Another recent approach for control of energy systems employs deep reinforcement learning (RL) and showcased promising results [14, 26]. For example, a soft actor-critic (SAC) agent achieved 5.79% reduction in fuel costs compared to rule-based control (RBC) over a two-days horizon in [14]. However, the study did not assess performance over longer horizons or across different periods of the year, limiting the results to the selected days.

Our work extends the previously reviewed research by introducing an application agnostic methodology to accelerate predictive control of multi-source district energy systems. The modularity of our framework is also demonstrated through its coupling with various optimizers, including both learning-based and evolutionary-based approaches. The surrogate model leverages spatio-temporal graph neural networks [22] and benefits from recent demonstrations showing that 'time-then-space' architectures offer superior expressivity compared to 'time-and-space' representations [17]. To further enhance generalizability, a physics-informed approach is used in training, where a mass conservation constraint, applicable to all inertial district energy systems, is added to the loss function.

3 Methodology

3.1 State-space surrogate model

Inertial district energy networks consist of several producers delivering energy to consumers via a network of pipes and control valves as shown in Fig. 2a. These



(a) (b)

Fig. 2: Figure (a) illustrates a generic graph representation of the physical components of an inertial district energy system. Figure (b) depicts the surrogate model (PI-STGCN) architecture which integrates past state variables ($x_-^{\mathcal{H}}$), future control signals ($u_+^{\mathcal{H}}$) and disturbances ($d_+^{\mathcal{H}}$).

systems can be modeled as a graph $\mathcal{G} = (\mathcal{V}, \ \mathcal{E})$, where nodes $\mathcal{V}$ represent producers, consumers, storage or valves, and edges $\mathcal{E}$ correspond to pipes. Each graph entity holds multiple interconnected physical features, nodes are characterized by state variables (e.g. $x =$ temperature) and the pipes by descriptive features such as their length and diameter. Moreover the entire system is influenced by exogenous variables, classified as disturbances (e.g. $d =$ weather) or control variables (e.g. $u =$ mass flowrate). Modern district energy networks incorporate diverse producer types at various locations [24]. Thermal energy storage further enhances flexibility by enabling asynchronous production and peak shaving.

The finite horizon optimal control problem in Eq. 1 seeks to optimize system trajectories while satisfying constraints. It minimizes a cost function C (e.g. fuel costs) over control variables u within the time horizon $\mathcal{H}^{\text{opt}}$. Here, the system dynamics (f in Eq. 1, computationally heavy) are replaced by a deep learning model f_θ (i.e., inference function), where θ represents model parameters. Due to system inertia and long-duration constraints on producers, predictive models require not only future control inputs and forecasted disturbances, but also historical state observations. This motivates the use of an autoregressive model formulation:

$$x_+^{\mathcal{H}^{\text{sm}}} = f_\theta\left(x_-^{\mathcal{H}^{\text{sm}}}, \ u_+^{\mathcal{H}^{\text{sm}}}, \ d_+^{\mathcal{H}^{\text{sm}}}\right). \tag{2}$$

Where $\mathcal{H}^{\text{sm}}$ is the predictive range of the surrogate model, typically shorter than the optimization horizon $\mathcal{H}^{\text{opt}}$. The symbol $+$ indicate predicted variables, meaning values from the current time t to $t + \mathcal{H}^{\text{sm}}$. Symbol $-$ indicates past observations or measurements of state variables, meaning values from $t - \mathcal{H}^{\text{sm}}$ to t. The surrogate model parameters θ are optimized through supervised learning using a dataset generated from a high-fidelity physical simulator validated against real-world data. This dataset captures the system response (i.e. state variables) to various control inputs and disturbances. The inference function f_θ is a result of the surrogate model architecture shown in Fig. 2b. In addition to the past state variables, the future control variables u and disturbances d are diffused to each node so that the contained information is available to all network components. Next, these node inputs go through the encoder-processor-decoder where gated recurrent units (GRU) are used for encoding and graph convolution (GCN) for message passing. Fig. 2b also introduces three hyperparameters that will be optimized: number of GRU layers, the hidden size (HS) and the number of GCN layers. The surrogate model is trained as the following optimization problem:

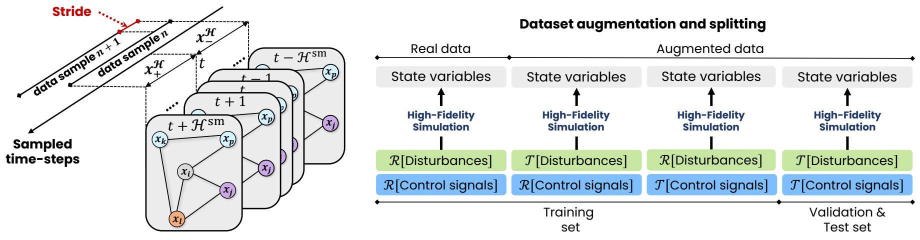
$$\begin{aligned}
\underset{\theta}{\text{minimize}} \quad & \frac{1}{N_b}\sum_{b}\frac{1}{\mathcal{H}^{\text{sm}}}\sum_{t_b}^{t_b+\mathcal{H}^{\text{sm}}}\left[\frac{1}{|\mathcal{V}|}\sum_{n}\|\hat{x}_{b,n,t}-x_{b,n,t}\|_2^2+\lambda\cdot\mathcal{F}_m^2\left(\hat{x}\right)\right], \\
\text{s.t.} \quad & \mathcal{F}_m\left(\hat{x}\right)=\sum_{\text{producers}}\hat{x}_{b,n,t}\;-\sum_{\text{consumers}}\hat{x}_{b,n,t}\;, \qquad (3) \\
& \hat{x}_+^{\mathcal{H}^{\text{sm}}}=f_\theta\left(x_-^{\mathcal{H}^{\text{sm}}},\; u_+^{\mathcal{H}^{\text{sm}}},\; d_+^{\mathcal{H}^{sm}}\right).
\end{aligned}$$

In Eq. 3, the loss term is weighted (via $\lambda > 0$) with a physical constraint term represented by $\mathcal{F}_m$. This term is the mass flow rates conservation over the network: the sum of the mass flow rates sent to the consumers must be equal to the sum of mass flow rates sent by the producers. The loss is averaged and calculated over a batch of size N_b and across all the nodes in the network $\mathcal{V}$. This surrogate model (PI-STGCN) is compared to other approaches, namely vector-autoregressive (VARx), a multi-layer perceptron (MLP) and a recurrent neural network based on gated recurrent units (RNN).

3.2 Training and validation pipeline

The training and validation pipeline explained in this section is schematized in the right block of Fig. 1. To construct the training dataset, historical measurements are in general available for such systems, specially weather, heat demand and control variables. Therefore, data samples (i.e., $\{x_+^{\mathcal{H}^{\text{sm}}}, x_-^{\mathcal{H}^{\text{sm}}}, u_+^{\mathcal{H}^{\text{sm}}}, d_+^{\mathcal{H}^{\text{sm}}}\})$ are constructed by sliding over the training dataset as shown in Fig. 3a by a number of time steps called ‘stride’, its minimum value corresponds to the physical simulation time step, and the impact of choosing larger values is analyzed in section 5.

The training dataset is constructed following the augmentation procedure schematized in Fig. 3b. In fact, the effectiveness of deep learning models depends on large training datasets. However, labeled data in many real-world time series applications are often scarce [37]. In model predictive control, the system response (i.e. state variables) are influenced by both the control variables and the external disturbances, via physical equations and constraints (recall Eq. 1). Therefore, the surrogate model must effectively capture the hidden dependencies between the state variables and the exogenous ones. This is the core idea schematized in Fig. 3b where the latter variables are augmented using the following procedure. We propose to augment time series by combining Gaussian scaling



(a) (b)

Fig. 3: Figure (a) illustrates dataset construction using a sliding window, with the 'stride' hyperparameter discussed later. Figure (b) presents the augmentation method based on state-space formulation. The original dataset, derived from real-world exogenous variables ($\mathcal{R}$), is used as a baseline. Each exogenous variable is then independently augmented ($\mathcal{T}$) and simulated using a high-fidelity physical digital twin.

and window warping [37, 32]. Let $z_{t|t+k} = [z_t, z_{t+1}, \dots, z_{t+k}]$ denote a time series sample spanning k time steps, starting from t, the resulting augmented time series $z_{t|t+k}^{\text{aug}}$ is computed via:

$$\mathcal{T}\left[z_{t|t+k}\right] = [\underbrace{\gamma_0 \cdot z_t, \dots,}_{\mathit{\Delta}\ window} \gamma_1 \cdot z_{t+\mathit{\Delta}}, \dots, \underbrace{\gamma_j \cdot z_{t+j\mathit{\Delta}}, \dots, \gamma_j \cdot z_{t+k}}_{\mathit{\Delta}\ window}], \tag{4}$$

where $\gamma \sim \mathcal{N}\left(1, \sigma_{aug}^2\right)$, $\mathcal{N}$ is the normal distribution, and $1 < \mathit{\Delta} < k$ is a fixed time window. $\mathit{\Delta}$ acts as the sampling frequency for scaling coefficients, meaning the data is scaled every $\mathit{\Delta}$ steps. State variables cannot be directly augmented, as their evolution is governed by physical laws. Instead, only control (u) and disturbance (d) inputs are augmented, thus, the resulting state trajectories are implicitly augmented through simulation of these inputs. The dataset is then scaled using min-max normalization and split to three distinct sets: training, validation and test (see Fig. 3b). The PI-STGCN model is trained using AdamW. Moreover, hyperparameters optimization is performed using the asynchronous successive halving algorithm (ASHA) from [23]. The considered hyperparameters and their corresponding range are given in Tab. 1 (a). The best model is then trained to reach its optimal performance, all experiments are performed on a 48 GB NVIDIA A40 GPU.

3.3 Optimization techniques

As shown in Fig. 1, the model predictive control leverages the trained surrogate model to perform control optimization. The complexity of the studied system dynamics requires reliable and robust optimizers. In the literature, evolutionary algorithms such as genetic algorithms (GA) and particle swarm optimization (PSO) were widely adopted [5, 28, 39]. Recently, deep reinforcement learning

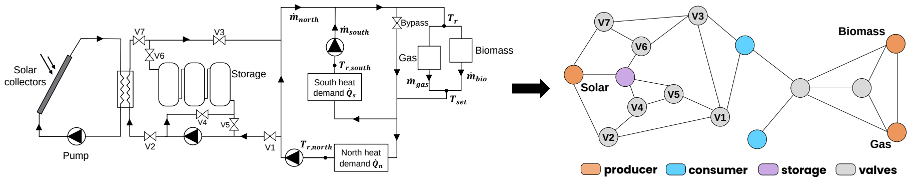


Fig. 4: Schematic of the real-world system and its graph representation, including biomass and gas boilers, solar field with thermal storage.

has also demonstrated its ability to handle the complex optimization of such systems [14, 26]. Therefore we implement four optimizers:

- **Evolutionary algorithms:** genetic algorithm (GA) and particle swarm optimization (PSO) are implemented using `pymoo` package [6]. The first is known for its robust exploration which avoids local optima, the second is used for its faster convergence rate [16].
- **Deep reinforcement learning:** the two on-policy and off-policy paradigms are tested by implementing proximal policy optimization (PPO) and soft actor-critic (SAC) respectively [31].

The optimal control problem in Eq. 1 can be directly solved using evolutionary algorithms, as they handle cost minimization and constraints separately. For RL, optimization constraints are added as penalty terms in the reward function.

4 Experiments

4.1 Study case description

The real-world system is illustrated in Fig. 4 and we have access to a physical simulator developed in `Dymola`, along with real consumption data and weather history . It incorporates three producers: biomass boiler, gas boiler, and a solar field connected to thermal storage. Several valves can be seen between the solar field and the storage, they allow different cycles: charging or discharging the storage, or direct injection from the solar field to the network. This system has several constraints, making it a representative case that requires complex control strategies: A) The biomass boiler must remain on for at least 72 h when started and off for at least 12 h when stopped. Besides, power variations are limited with ramp constraints. B) Solar energy must be used when available to avoid overheating and maximize renewable integration. C) Producers must supply sufficient heat to meet demand while maintaining temperature levels above a fixed threshold.

In the state-space formulation, mass flow rates are state variables for boilers and control valves. Boiler mass flow rates are proportional to power output, while valve flow rates indicate open or closed states. For consumers and storage, fluid temperature is considered as the state variable. In storage, temperature reflects stored heat, while consumer temperatures indicate the heat demand. The system also depends on external disturbances: solar irradiance G_{irr} and ambient temperature T_{amb}, which affect solar energy production, as well as heat demand in the northern $\dot{Q}_{\text{n}}$ and southern $\dot{Q}_{\text{s}}$ clusters. The predictive control optimizes energy source usage, determining when to activate sources, set power levels, and adjust flow rates. To do so, the network operators control the mass flow rates sent to each cluster, which updates Eq. 1 as follows:

$$u(t) = \left[\dot{m}_{\text{n}}(t),\ \dot{m}_{\text{s}}(t)\right], \quad \text{and} \quad d(t) = \left[G_{\text{irr}}(t),\ T_{\text{ext}}(t),\ \dot{Q}_{\text{n}}(t),\ \dot{Q}_{\text{s}}(t)\right]. \qquad (5)$$

The cost function C is set to the economic operational cost of the network and can be written as:

$$C\left(t_f,\ u\right) = \int_0^{t_f} c_{\text{bio}} \cdot \dot{Q}_{\text{bio}}(t) + c_{\text{gas}} \cdot \dot{Q}_{\text{gas}}(t)\ dt. \qquad (6)$$

Where c_{bio} and c_{gas} are fuel costs (in €/kWh) and $\dot{Q}$ the energy provided by biomass and gas respectively. Moreover, the optimal control must satisfy two constraints:

$$\left(\frac{u_{i,t} - u_{i,t-1}}{dt}\right)^2 \leq \delta_u, \quad \text{and} \quad x_{i,t} \geq x_{min} \ \ \forall t \in \mathcal{H}^{opt},\ i \in \{\text{n}, \text{s}\}. \qquad (7)$$

The first constraint limits control variable variations to protect hydraulic pumps, while the second ensures outlet temperatures stay above a threshold for comfort and safety. As outlined in section 3.3, constrained optimizations must be adapted to align with the reward-learning framework of RL agents. Accordingly, both PPO and SAC were trained using the following reward function:

$$r_t = -\frac{C_t}{C^*} - \frac{\lambda_u}{\mathcal{H}^{\text{sm}}} \sum_i \sum_t^{t+\mathcal{H}^{\text{sm}}} \left(\frac{u_{i,t} - u_{i,t-1}}{dt}\right)^2 - \frac{\lambda_x}{\mathcal{H}^{\text{sm}}} \sum_i \sum_t^{t+\mathcal{H}^{\text{sm}}} e^{x_{min} - x_{i,t}}. \qquad (8)$$

This formulation balances three terms: minimizing economic costs (normalized by the RBC cost C^*), enforcing smooth control variations (weighted by λ_u), and ensuring temperature constraints (weighted by λ_x).

4.2 Scalability test cases

Confidentiality limits access to real-world district energy network data, especially for large, variable-sized networks. While the studied system (in section 4.1) is representative in terms of energy sources and constraints, the lack of publicly available data makes synthetic testing essential. To address this, we introduce a synthetic dataset to evaluate scalability, focusing on the number of connected nodes[3].

[3] Link to dataset: `https://doi.org/10.17605/OSF.IO/ZBJ5W`

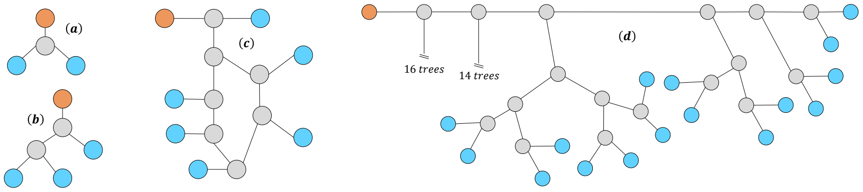


Fig. 5: Synthetic use cases for surrogate model scalability assessment. The number of nodes is 4 in (a), 6 in (b), 14 in (c) and 100 in (d).

A dynamic simulator was used to generate a one-year simulation for each network in Fig. 5 using a 10, 15 and 20-minute time step. The used open-source physical solver (`TESPy` [38]) has been validated on different use cases [18, 11]. The networks include consumers with different usage profiles -residential, commercial, administrative- generated using the `demandlib` package [30].
Each node has four state variables: inlet/outlet pressure and inlet/return temperatures. Disturbances include heat demand per consumer and ambient temperature, while control variables are supply temperature and mass flow rate of the producer node. Therefore, the scalability evaluates multiple aspects: adaptability to networks of varying sizes and the flexibility of the architecture in handling different numbers of state variables, control variables and disturbances.

5 Results

The prediction performance of the proposed surrogate model is evaluated using multiple regression metrics. All models are tested on the same dataset, consisting of the fully augmented year (recall Fig. 3b). Examples of augmented inputs are shown in Fig. 6a and 6b. To ensure realistic signals, augmentation is applied every $\Delta = 3$ hours, with σ_{aug} calibrated so that $\gamma_i \in [0.9, 1.1]$, keeping input variations within feasible bounds, especially for control variables like pumps flow rates. Two accuracy metrics are used: the mean squared error (MSE), which quantifies overall absolute prediction error on normalized data, and the symmetric mean absolute percentage error (sMAPE), which expresses the mean absolute error as a percentage. For precision, the coefficient of determination (R^2) is computed, and the percentage of mass balance compliance ($\mathcal{F}_m$) is reported.

5.1 Prediction performance

In the following, results are shown for the best model configuration found through hyperparameters optimization. The ASHA samples 150 different configuration from the search space specified in Tab. 1 (a). Unless pruned earlier by the optimizer, each configuration was trained for a maximum of 30 epochs. The top two configurations in Tab. 1 (b) share the same architecture, differing only in l_r and λ, with 3.2M trainable parameters. Other surrogate models are set to the

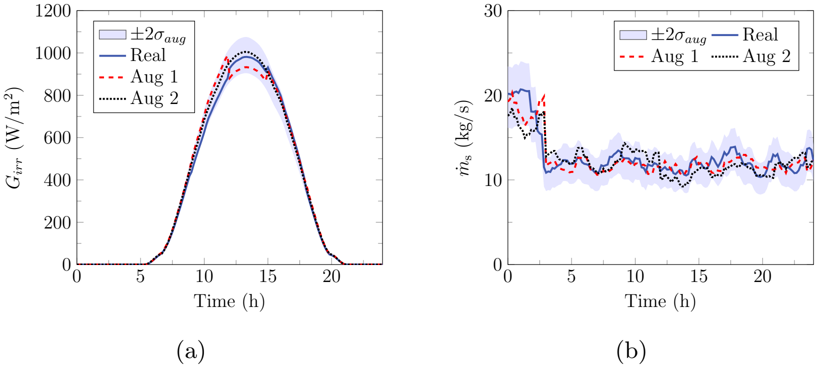


Fig. 6: Figures (a) and (b) show typical examples of solar irradiance and control variable augmentation over a one-day period.

same parameter count for performance comparison, except for VARx, which has a fixed number of parameters. The stride hyperparameter is set to the smallest possible value (stride = 10 min).

Table 1: Hyperparameter search space and results. For (b): MSE is $\times 10^{-2}$, l_r is $\times 10^{-4}$ and $\mathcal{H}^{\text{sm}} = 12$ h (i.e., same for all 5 configurations).

(a) ASHA search space

hyperparameter	**search space**
Batch size (N_b)	$\{64,\ 128\}$
Hidden size (HS)	$\{128,\ 256,\ 512\}$
GRU & GCN layers	$\{1,\ 2,\ 4\}$
Horizon ($\mathcal{H}^{\text{sm}}$)	$\{6\text{h},\ 12\text{h},\ 24\text{h}\}$
$l_r \in [10^{-4},\ 10^{-1}]$	$\lambda \in [10^{-2},\ 1]$

(b) Top 5 configurations

rank	mse	gru/gcn	l_r	λ	HS	N_b
1	1.163	2/1	0.7	1	512	128
2	1.164	2/1	0.4	0.1	512	64
3	1.171	4/4	0.5	0.1	128	64
4	1.183	4/1	0.6	0.3	256	64
5	1.184	2/4	1.2	0.1	256	64

In Fig. 7a-7c, the proposed surrogate model (PI-STGCN) outperforms all other models. Specifically, sMAPE is around 1.3%, and mass balance $\mathcal{F}_m$ is satisfied for 99% of tested samples. The impact of including $\mathcal{F}_m$ in the loss function (Eq. 3) is clear when comparing with STGCN, which shows degraded performance across all metrics. The MLP model performs significantly worse than the others, as it lacks both temporal dependencies (captured by all the others) and spatial dependencies (captured by STGCN and PI-STGCN).
The sensitivity analysis in Fig. 7b highlights the benefits of dataset augmentation. Training with 1 real year (RY) and 2 augmented years (AY) reduces sMAPE by 30%, MSE by 10%, and increases R^2 by 4% compared to using only 1 RY. Fig. 7d examines the impact of the 'stride' hyperparameter from Fig. 3a.

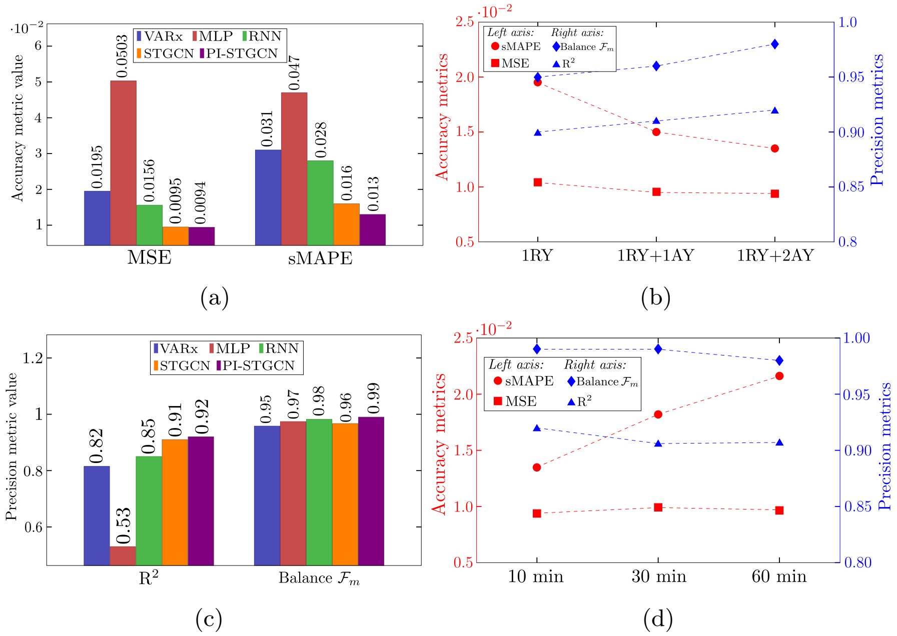
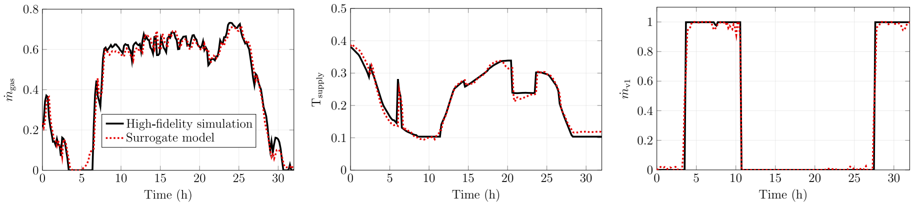


Fig. 7: Figures (a) and (c) compare the PI-STGCN model with other approaches using the real-world case from Fig. 4. Figures (b) and (d) analyze the model's sensitivity to the augmented dataset size (RY: real year, AY: augmented year) and the stride hyperparameter (10, 30 and 60 minutes) respectively.

Three stride values were tested using the optimal model and 1RY+2AY dataset. The smallest stride (10 min) matches the real sampling frequency of weather data and numerical simulations. Results show a 42% lower sMAPE with s = 10 min vs. 1 h. In fact, a smaller stride increases training samples and enhances the model's ability to capture fast dynamics, particularly mass flow rates. However, high performance is still maintained across all tested strides, demonstrating robustness to different sampling rates. An example of PI-STGCN predictions on a test sample are shown in Fig. 8.

It can be seen that different dynamic patterns are well captured, both fast (8a) and relatively slow (8b) evolutions are learned. Moreover, the on/off behavior of control valves (8c) is precisely learned, this makes the model remarkably accurate. More quantitatively, the mean absolute errors (MAE) are in acceptable range for network operators: the supply temperature error remains below 1 K, the return temperatures below 1.5 K, and biomass/gas flow errors stay under 2 kg/s. Finally, the surrogate model achieves this high accuracy while reducing computational time by four orders of magnitude per data sample (speed-up factor $1.9 \cdot 10^4$) compared to `Dymola` physical simulator.

The results of the scalability assessment (using the networks of Fig. 5) are reported in Tab. 2.



(a) (b) (c)

Fig. 8: An example of PI-STGCN predictions (red dotted curve) compared to high-fidelity simulation (black curve) over an illustrative period. The model accurately captures system dynamics, with prediction errors within an acceptable range for network operators.

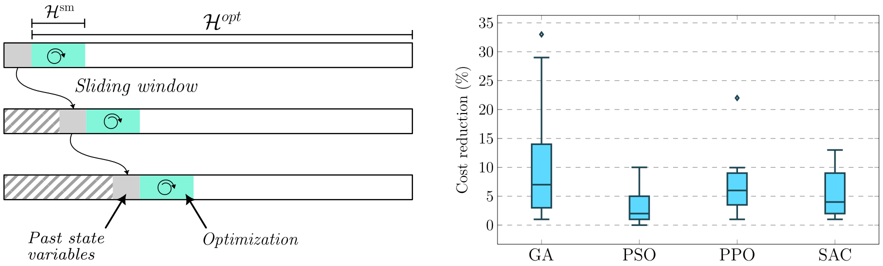
Table 2: PI-STGCN scalability evaluation on the test cases from Fig 5.

Network	$\mathcal{V} = 4$	$\mathcal{V} = 6$	$\mathcal{V} = 14$	$\mathcal{V} = 100$
MSE	$1.2 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	$1.9 \cdot 10^{-4}$	$8.1 \cdot 10^{-5}$
sMAPE	0.21	0.18	0.15	0.10
R^2	0.98	0.99	0.97	0.98
Simulation/sample	27 s	41 s	203 s	665 s
Inference/sample	20 ms	23 ms	52 ms	330 ms
Simulation/node	6.7 s	6.8 s	14.5 s	6.7 s
Inference/node	5 ms	3.8 ms	3.7 ms	3.3 ms

The surrogate model maintains high accuracy across all tested networks, with R^2 consistently above 0.97 and MSE decreasing as network size increases, indicating robustness in larger test cases. Physical simulations become increasingly expensive, with per-sample run-time rising from 27 s to 665 s, while surrogate inference remains efficient, increasing only from 20 ms to 330 ms. On average, this corresponds to a three order of magnitude reduction, which is lower than the $1.9 \cdot 10^4$ reduction observed when comparing inference to `Dymola` simulations. In fact, `TESPy` simulation used here includes certain physical simplifications, making it less computationally intensive than `Dymola`. On a per-node basis, inference time decreases with network size, from 5 ms to 3.3 ms, whereas simulation time remains stable around 6.7 s per node. This suggests that the surrogate model's inference time scales sub-linearly with the number of nodes.

5.2 Application to control optimization

After validating the learned state-space model, we demonstrate its use in optimal control. As shown in Fig. 1, the surrogate model estimates objective function values for control scenarios generated by the optimizers. To evaluate performance across seasons, optimal control is run over a full year using a sliding window with $\mathcal{H}^{opt} = 3$ days, illustrated in Fig. 9a. GA and PSO were executed until cost variations remained within a 5 euros tolerance for at least 10 generations or until the maximum number of iterations (200) was reached. PPO and SAC explored optimization periods until reaching a maximum cumulative reward, requiring 1.5 million (2h 40min) and 1 million (5h 55min) iterations, respectively. Prior to training, reward weights (λ_u, λ_x) in Eq. 8 were optimized using ASHA with Optuna's multi-objective framework [1]. The search space was $[10^{-3}, 1]$ for each weight, and the best pair was selected to maximize cost reduction while minimizing penalty terms (see details in Appendix. A.1).



(a) (b)

Fig. 9: Figure (a) illustrates the sliding window control deployed. Figure (b) summarizes the cost reductions results for the four implemented optimizers.

The optimization results are summarized in the box plot in Fig. 9b, showing cost reductions only for periods where optimizers were effective. During high-demand winter periods, when the boilers are at full capacities, cost reduction margins are smaller, with optimized costs just 0.1% lower than RBC, so they are not reported here. In contrast, mid-season periods showed greater savings due to the increased flexibility offered by solar energy and the variable heat demand. GA achieved the highest reductions, up to 33% with a median of 7%, followed by PPO (6%), SAC (4.5%), and PSO (3%). The two evolutionary-based methods show contrasting behavior: PSO had a faster convergence rate but tended to get stuck in local optima, explaining its lower performance. For RL agents, some periods with higher cost reductions were excluded due to violations of temperature constraints (Eq. 7), thus, the reward function needs further refinement. Another alternative is a hierarchical architecture with two agents: one enforcing constraints, the other optimizing cost.

A key feature of our methodology is the significant reduction in computation time. For instance, a single period optimization using GA or PSO takes about 10 minutes, while it would require more than 24 hours if `Dymola` simulator was used. This improvement enables real-time optimal control and underscores the advantages of deep learning models in accelerating the optimization and deployment of inertial district energy systems.

6 Conclusion

In this work, we presented a deep learning-based methodology for optimal control of computationally intensive multi-source inertial district energy networks. We provide an system-agnostic framework for training and validating a physics-informed surrogate model, leveraging historical data augmentation and hyperparameters optimization. The dynamics were precisely learned, with sMAPE values lower than 1.3% and R^2 values over 0.97. Furthermore, the trained model modularity is demonstrated by leveraging it with both evolutionary-based optimizers and reinforcement learning agents. Our results demonstrate significant improvements, with cost reductions of up 30% in mid-season and a drastic decrease in computational time (up to four orders of magnitude). Future lines of work will address the current limitations: Improve the reward function to increase the effectiveness of RL strategies and include stochastic forecasts to better capture real-world noise and variability. Another perspective is considering cases with unusual disturbances (e.g. extreme weather) and/or equipment failures, followed by an analysis of the control strategies found by the optimizers.

Acknowledgments. This work was supported by the French Ministry of Higher Education and Research.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: A next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2019)
2. Angelidis, O., Ioannou, A., Friedrich, D., Thomson, A., Falcone, G.: District heating and cooling networks with decentralized energy substations: Opportunities and barriers for holistic energy system decarbonisation. Energy **269**, 126740 (2023)
3. Ayed, I., de Bézenac, E., Pajot, A., Brajard, J., Gallinari, P.: Learning dynamical systems from partial observations. arXiv preprint arXiv:1902.11136 (2019)
4. Beintema, G.I., Schoukens, M., Tóth, R.: Continuous-time identification of dynamic state-space models by deep subspace encoding. In: ICLR (2023)
5. Betancourt Schwarz, M., Veyron, M., Clausse, M.: Impact of flexibility implementation on the control of a solar district heating system. In: Solar. vol. 4, pp. 1–14. MDPI (2023)
6. Blank, J., Deb, K.: pymoo: Multi-objective optimization in python. IEEE Access **8**, 89497–89509 (2020)
7. Blaud, P.C., Chevrel, P., Claveau, F., Haurant, P., Mouraud, A.: From multi-physics models to neural network for predictive control synthesis. Optimal Control Applications and Methods (2021)
8. Botache, D., Decke, J., Ripken, W., Dornipati, A., Götz-Hahn, F., Ayeb, M., Sick, B.: Enhancing multi-objective optimisation through machine learning-supported multiphysics simulation. In: Joint ECML-PKDD. pp. 297–312. Springer (2024)

9. Boussaid, T., Rousset, F., Scuturici, V.M., Clausse, M.: Enabling fast prediction of district heating networks transients via a physics-guided graph neural network. Applied Energy **370**, 123634 (2024)
10. Bronstein, M.M., Bruna, J., Cohen, T., Veličković, P.: Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv preprint arXiv:2104.13478 (2021)
11. Chen, C., Witte, F., Tuschy, I., Kolditz, O., Shao, H.: Parametric optimization and comparative study of an organic rankine cycle power plant for two-phase geothermal sources. Energy **252**, 123910 (2022)
12. Chitalia, G., Pipattanasomporn, M., Garg, V., Rahman, S.: Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks. Applied Energy **278**, 115410 (2020)
13. Delubac, R., Serra, S., Sochard, S., Reneaume, J.M.: A dynamic optimization tool to size and operate solar thermal district heating networks production plants. Energies **14**(23), 8003 (2021)
14. Deng, J., Eklund, M., Sierla, S., Savolainen, J., Niemistö, H., Karhela, T., Vyatkin, V.: Deep reinforcement learning for fuel cost optimization in district heating. Sustainable Cities and Society **99**, 104955 (2023)
15. Dorotić, H., Pukšec, T., Duić, N.: Multi-objective optimization of district heating and cooling systems for a one-year time horizon. Energy **169**, 319–328 (2019)
16. Gad, A.G.: Particle swarm optimization algorithm and its applications: a systematic review. Archives of computational methods in engineering **29**(5), 2531–2561 (2022)
17. Gao, J., Ribeiro, B.: On the equivalence between temporal and static graph representations for observational predictions. In: International Conference on Machine Learning. PMLR (2022)
18. Gasanzade, F., Witte, F., Tuschy, I., Bauer, S.: Integration of geological compressed air energy storage into future energy supply systems dominated by renewable power sources. Energy Conversion and Management **277**, 116643 (2023)
19. de Giuli, L.B., La Bella, A., Scattolini, R.: Physics-informed neural network modeling and predictive control of district heating systems. IEEE Transactions on Control Systems Technology (2024)
20. Huang, Y., Zhao, Y., Wang, Z., Liu, X., Liu, H., Fu, Y.: Explainable district heat load forecasting with active deep learning. Applied Energy **350**, 121753 (2023)
21. IEA: Net zero roadmap: A global pathway to keep the 1.5 °c goal in reach (2023), licence: CC BY 4.0
22. Ji, J., Wang, J., Huang, C., Wu, J., Xu, B., Wu, Z., Zhang, J., Zheng, Y.: Spatio-temporal self-supervised learning for traffic flow prediction. In: Proceedings of the AAAI conference. vol. 37, pp. 4356–4364 (2023)
23. Li, L., Jamieson, K., Rostamizadeh, A., Gonina, K., Hardt, M., Recht, B., Talwalkar, A.: Massively parallel hyperparameter tuning. arXiv preprint arXiv:1810.05934 (2018)
24. Pakere, I., Feofilovs, M., Lepiksaar, K., Vītoliņš, V., Blumberga, D.: Multi-source district heating system full decarbonization strategies: Technical, economic, and environmental assessment. Energy **285**, 129296 (2023)
25. Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., Battaglia, P.W.: Learning mesh-based simulation with graph networks. arXiv preprint arXiv:2010.03409 (2020)
26. Pinto, G., Deltetto, D., Capozzoli, A.: Data-driven district energy management with surrogate models and deep reinforcement learning. Applied Energy **304**, 117642 (2021)

27. Portner, H., Roberts, D., Adams, H., Adler, C., Aldunce, P., Ali, E., Begum, R.A., Betts, R., Kerr, R.B., Biesbroek, R., et al.: Climate change 2022: Impacts, adaptation and vulnerability. Tech. rep., IPCC (2022)
28. Qin, C., Yan, Q., He, G.: Integrated energy systems planning with electricity, heat and gas using particle swarm optimization. Energy **188**, 116044 (2019)
29. Saloux, E., Runge, J., Zhang, K.: Operation optimization of multi-boiler district heating systems using artificial intelligence-based model predictive control: Field demonstrations. Energy **285**, 129524 (2023)
30. Schachler B, M.C.e.a.: demandlib: creating heat and power demand profiles from annual values (2021), `https://oemof.org/libraries/demandlib`
31. Sutton, R.S.: Reinforcement learning: An introduction. A Bradford Book (2018)
32. Tran, L., Choi, D.: Data augmentation for inertial sensor-based gait deep neural network. IEEE Access **8**, 12364–12378 (2020)
33. Verma, Y., Heinonen, M., Garg, V.: ClimODE: Climate and weather forecasting with physics-informed neural ODEs. In: ICLR (2024)
34. Wang, X., Liu, Y., Zhao, J., Liu, C., Liu, J., Yan, J.: Surrogate model enabled deep reinforcement learning for hybrid energy community operation. Applied Energy **289**, 116722 (2021)
35. Wang, Z., Liu, X., Huang, Y., Zhang, P., Fu, Y.: A multivariate time series graph neural network for district heat load forecasting. Energy **278**, 127911 (2023)
36. Weigand, J., Deflorian, M., Ruskowski, M.: Input-to-state stability for system identification with continuous-time runge–kutta neural networks. International Journal of Control **96**(1), 24–40 (2023)
37. Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H.: Time series data augmentation for deep learning: A survey. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 4653–4660 (2021). https://doi.org/10.24963/ijcai.2021/631
38. Witte, F., Tuschy, I.: TESPy: Thermal Engineering Systems in Python. Journal of Open Source Software **5**(49), 2178 (2020). https://doi.org/10.21105/joss.02178
39. Wu, M., Du, P., Jiang, M., Goh, H.H., Zhu, H., Zhang, D., Wu, T.: An integrated energy system optimization strategy based on particle swarm optimization algorithm. Energy Reports **8**, 679–691 (2022)
40. Yu, Y., Jiang, X., Huang, D., Li, Y., Yue, M., Zhao, T.: Pidgeun: Graph neural network-enabled transient dynamics prediction of networked microgrids through full-field measurement. IEEE Access (2024)

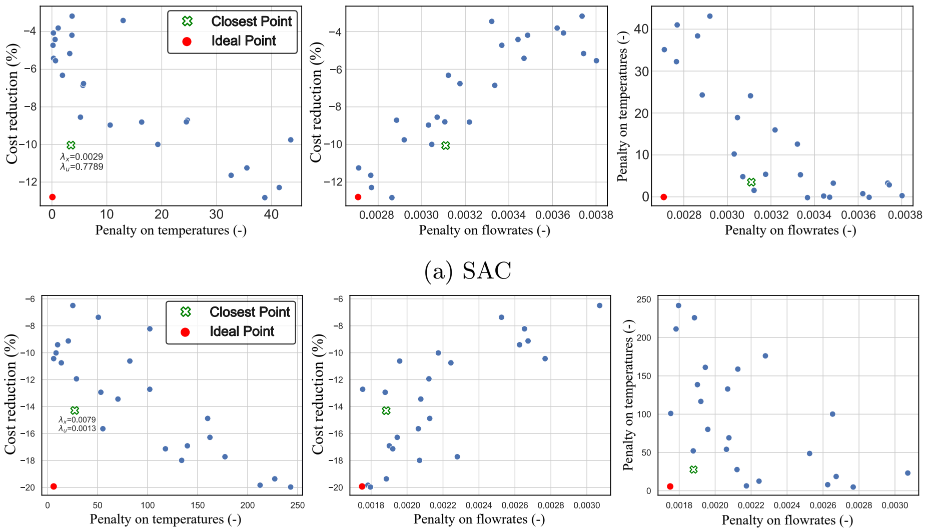
A Appendix

A.1 RL hyperparameters optimization

For each agent, ASHA samples 25 configurations. SAC is trained for 150,000 steps, while PPO runs for 250,000 steps due to slower convergence. Each term in the reward function (Eq. 8) is evaluated as a separate metric: the negative cost reduction (%), the temperature penalty, and the flowrates penalty (both normalized). The optimization minimizes these three metrics, and the resulting Pareto fronts are shown in Fig. 10.

The optimal hyperparameters (λ_u, λ_x) are selected as the closest point (distance wise) to the "ideal" minimum for each objective individually. The other hyperparameters are taken from RL Zoo[4] and are reported below in Tab. 3.

[4] `https://github.com/DLR-RM/rl-baselines3-zoo`



(a) SAC

(b) PPO

Fig. 10: Hyperparameter optimization results. Each subplot shows a projection of the 3-objective Pareto surface for SAC and PPO.

Table 3: Other hyperparameters for SAC and PPO.

Hyperparameter	**SAC**	**PPO**
Shared		
Batch size (N_b)	128	64
Discount factor (γ)	0.99	
Learning rate	$3 \cdot 10^{-4}$	
SAC-Specific		
Buffer size ($N_{\mathcal{B}}$)	10^4	-
Training frequency & Gradient cycle	1	-
Target smoothing coefficient	0.005	-
Target update interval	100	-
PPO-Specific		
Number of collected steps (N_{steps})	-	1920
Gradient cycle (N_{epochs})	-	10
Actor & Critic Networks		
Number of hidden layers	2	2
Number of hidden units	256	64
Non-linear activations	`ReLU`	`Tanh`