Exploring Cross-Project Building Performance Prediction: Creation and Implementation of a Zone-Based Surrogate Model for Building Performance Optimization

Artem Verkhovskiy¹, Abraham Yezioro¹, Isaac Guedi Capeluto¹
¹Technion - Israel Institute of Technology, Haifa, Israel

Abstract

Sustainable architectural design is increasingly recognized as an important aspect of building construction. However, not all architectural offices can readily achieve sustainability objectives in their projects, due to various constraints such as the lack of accessible tools, systematic approaches, time, and expertise. The described approach aims to contribute to the field of building performance optimization by developing a machine learning-based surrogate modeling framework that can be used for a wide range of projects without the need for project-specific simulations or model training. The proposed methodology utilizes a thermal zone-based dataset to train an artificial neural network-based surrogate model, which is integrated into a workflow that combines the model-to-data conversion algorithm and optimization algorithm. The approach allows architects and engineers to quickly assess the impact of different design parameters on building performance without the need for extensive simulations. A case study of a residential building in Tel Aviv, Israel demonstrated the effectiveness of the approach, with near-optimal solutions obtained in a shorter time than with simulation-based optimization.

Highlights

- A novel cross-project building performance optimization approach using surrogate modeling and a component-based methodology
- User-friendly workflow allows for easy implementation without programming or machine learning expertise
- Time-efficient surrogate model-based optimization yields multiple near-optimal solutions for a design space of a user’s parametric energy model
- The effectiveness of the approach is successfully demonstrated through testing on a residential building test case

Introduction

Sustainable architecture has gained significant attention in recent years, as architects, builders, and policymakers seek to mitigate the environmental impact of new and existing buildings (World Green Building Council, 2018). However, achieving sustainable goals can be challenging for architectural firms due to various constraints such as the lack of accessible tools, systematic approaches, and resources, such as time and expertise (Attia et al., 2013).

Building Performance Optimization (BPO) has become an important approach to sustainable architectural design, intending to optimize energy efficiency, comfort, daylight, or any other aspect of a building's lifecycle. BPO goes beyond single-run simulations, offering a more in-depth examination of building performance by considering many scenarios and design alternatives (Evins, 2013). Studies have shown that BPO during the design process leads to significant improvements in building performance targets compared to traditional single-run simulation studies (Nguyen et al., 2014). Despite its importance, BPO is a complex and time-consuming process that requires specialized knowledge and expertise, as well as access to advanced simulation tools and techniques. These drawbacks make BPO a challenge for many architects, especially those working on smaller projects or with limited resources.

To overcome these challenges, researchers have proposed an approach to BPO that leverages machine learning (ML) algorithms to develop surrogate models that represent the relationship between building design variables and performance outputs. The resulting surrogate models can be used to make predictions about building performance with much less computational effort compared to traditional simulation-based approaches (Roman et al., 2020; Westermann & Evins, 2019). However, most of the recent research in the field of surrogate model-based approaches for BPO focused on developing a surrogate model for a specific project, which limits its applicability to similar projects with similar design space. Thus, the development of a generalized surrogate model that can be applied to a wide range of projects without the need for additional simulations or model training has become an important research topic. This would eliminate the need for project-specific surrogate model training and reduce the time and ML skills required for each project (Figure 1).

![Figure 1: Simplified workflow schemes: (a) State-of-art surrogate model-based BPO process; (b) Proposed surrogate model-based BPO process.](https://doi.org/10.26868/25222708.2023.1330)
Several studies have explored the use of ML techniques to create surrogate models that could apply to multiple projects. For instance, Vazquez-Canteli et al. (2019) and Thrampoulidis et al. (2021) proposed using data from urban-scale simulations to create surrogate models for predicting building performance and finding optimal retrofit solutions. Tsanas & Xifara (2012) developed a surrogate model to reflect the impact of general design parameters on heating and cooling loads in residential buildings, while Pittarello et al. (2021) developed a surrogate model tool to forecast building energy consumption in the early design stages. However, these studies lack consideration of more advanced building plan configurations and design variables like shading and fenestration.

Other studies have shown that surrogate models can predict performance across multiple climate zones using weather data parameters. Rackes et al. (2016) developed a surrogate model for aiding the preliminary design of naturally ventilated low-rise commercial buildings in warm and hot climates, while Westermann et al. (2020) developed a single-building surrogate model that can make predictions for arbitrary climate regions. However, both studies require additional training for various projects and may not be suitable for complex design scenarios.

The component-based approach to building performance prediction is a more promising strategy for cross-project surrogate modeling, as it allows for more accurate performance prediction through the use of separate surrogate models for each component (zone, wall, window, floor, etc.). Geyer & Singaravel (2018) studied two strategies for this approach, achieving good prediction accuracies using a surrogate model, but it was unable to handle non-orthogonal designs and did not consider shading effects. Based on this approach Singh et al. (2020) developed a BIM-integrated tool for early design stages and Li et al. (2019) explored implementing the component-based approach for more complex building schemes. Zhu et al. (2021) developed a zone-based surrogate model that considered shading and reflection effects and compared the performance of six ML models. Labib (2022) developed a surrogate model for predicting daylight autonomy values for office rooms.

However, none of these studies investigated the coupling of surrogate models with optimization algorithms for automatic design improvement or thoroughly investigated the models’ ability to produce optimal solutions. Additionally, there is a lack of methodologies proposed for advanced design stages, which require more specific design parameters. Therefore, further research is necessary to explore the potential of developing surrogate models that can effectively optimize building design at both early and more advanced stages of the design process.

Methods

The primary objective of this work was to develop a fast and efficient BPO workflow that can be seamlessly integrated into any parametric study. To accomplish this goal, the study proposes to use zone-based surrogate models that are specifically designed to be suitable for use in a variety of projects of the same type in a particular climate zone. To enable the surrogate model to accurately predict the performance of buildings with varying levels of complexity, a proposed approach involves dividing and simplifying the user’s parametric model into separate zones. Performance predictions are then conducted for each individual zone, and the results are combined to generate an assessment of the entire building.

The resulting workflow of BPO consists of the following steps (Figure 2):

1. The user develops a parametric energy model of the building and defines a design space using a set of parameters. These parameters can include the layout of the building, plan configuration, wall construction, window size and materials, shading elements, and multiple other features, that are not related to the input parameters of the surrogate model;
2. The model parameters are linked directly to the optimization algorithm, and can be chosen and configured by the user according to their preferences;
3. The parametric energy model is connected to the model-to-data conversion (M2DC) algorithm, which automatically converts each model option considered during the optimization into data for the surrogate model. This is done by dividing the model into zones, simplifying each zone, and analyzing the parameters of the simplified zones;
4. The parameters of the simplified zones are passed into a pre-trained surrogate model that predicts the performance of each individual zone;
5. The surrogate model’s predictions for all the zones in the current option are combined to determine the building’s overall performance (Heating and Cooling energy consumption);
6. The resulting performance value is used as an objective for the optimization algorithm;
7. At the end of the optimization, the user receives several near-optimal solutions for the given design space.

Figure 2: Proposed workflow scheme.

To develop such a workflow, several challenges were addressed. First, the input parameters for the surrogate model were defined, along with their respective ranges, and relevant performance indicators for BPO were selected. A parametric model of a zone was created to depict the entire design space based on these parameters. This design space was then sampled, and simulations
were performed using the developed parametric model to generate a dataset containing the zone's parameters and corresponding energy performance. Based on this dataset, a surrogate model was trained. To ensure that the surrogate model could be used in any project, an M2DC algorithm was developed to automatically convert the user's model into the input data necessary for the surrogate model without human interaction. These steps will be discussed in greater detail in the following sections.

Zone parametric model development

When creating a surrogate model, it is crucial to consider the impact of design space on the accuracy of predictions. As the number of parameters and their ranges increases, the model's accuracy decreases. Therefore, special attention must be paid to the selection of input parameters. If possible, the design space should be divided into smaller subspaces to mitigate the decrease in prediction accuracy. In this study, the design space is limited to one building type (residential buildings) and one climate region (Tel Aviv, Israel).

To create a surrogate model based on zones, a dataset must be constructed to represent a wide range of possible zone configurations and their performance. It is necessary to carefully select parameters that can effectively characterize the possible zone configurations while maintaining a reasonable level of abstraction.

To test the effectiveness of the method at different levels of problem complexity, two surrogate models were prepared - a full-scale and a reduced model. The reduced surrogate model was trained on a reduced number of parameters, which limits its functionality but enhances the accuracy of its predictions. Table 1 lists the parameters used in the surrogate models.

A parametric energy model representing the developed design space was created using Python and the Ladybug Tools SDK to demonstrate the impact of the full set of parameters on the zone's performance through simulations and collect the necessary data for surrogate model training and testing. Figure 3 displays a couple of possible configurations of the parametric model. The energy loads and schedules (related to occupants, lighting, and equipment) specified in the Israeli Standard 5282 for residential buildings (SI 5282: Energy Rating of Buildings, 2017) were used as settings for the energy model. Heavy external wall structure with varying EPS insulation thickness (as per Table 1) was considered for the study. As the developed method aims to optimize a standard floor within a multi-story residential building, the floor and ceiling of the zone are considered to be adiabatic.

To quantify the zone's performance, combined energy load from heating and cooling was selected as the optimization objective in this study. EnergyPlus was utilized as the simulation engine.

Table 1: Surrogate model design space. Parameters marked with * were used only for the full-scale surrogate model.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Description</th>
<th>Unit</th>
<th>Value Type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Width</td>
<td>Zone width (East - West)</td>
<td>m</td>
<td>Continuous</td>
<td>2 - 16</td>
</tr>
<tr>
<td>Length</td>
<td>Zone length (North - South)</td>
<td>m</td>
<td>Continuous</td>
<td>2 - 16</td>
</tr>
<tr>
<td>Height *</td>
<td>Floor to ceiling height</td>
<td>m</td>
<td>Continuous</td>
<td>2.5 - 5</td>
</tr>
<tr>
<td>Azimuth *</td>
<td>Angle between North axis and zone North</td>
<td>°</td>
<td>Continuous</td>
<td>-45 - 45</td>
</tr>
<tr>
<td>Insulation thickness *</td>
<td>3 options for the insulation layer thickness</td>
<td>m</td>
<td>Discrete</td>
<td>0.02, 0.04, 0.06</td>
</tr>
<tr>
<td>Glazing type *</td>
<td>2 options for window construction</td>
<td>-</td>
<td>Discrete</td>
<td>Single, double</td>
</tr>
</tbody>
</table>

Parameters repeated for each separate side of the zone (North, East, South, West):

- WWR: Window-to-Wall Ratio - Continuous - 0 - 1
- Shade size: length of outdoor structure at zone ceiling level - Continuous - 0 - 4
- CSR: Context-to-Sky Ratio - Continuous - 0 - 1
- Wall boundary condition: Adiabatic part of the wall - Continuous - 0 - 1

Database creation

To build an accurate surrogate model, two sets of data must be prepared: a training set and a test set that have identical structures (i.e. the same set of parameters and performance indicators in the case of this study). The quality of the surrogate model's predictions is largely determined by the choice of the data that is included in the dataset. To ensure that the training set is representative of the design space and includes a diverse range of parameter combinations the Latin Hypercube Sampling (LHS) was employed. As the sampling strategy for the test set does not impact the quality of training, it was sampled randomly. To sample the design space, a Grasshopper plugin called Design Space Exploration was used.

To improve the accuracy of predictions, the design space was divided into smaller subsets. A separate surrogate model was created for each subset to better capture the performance characteristics of each group of zones. For both the reduced and full-scale surrogate model this subdivision was made based on the wall boundary conditions, which have several possible combinations, for the full-scale model the design space was additionally subdivided by the insulation thickness and glazing type, which are discrete parameters. The total number of samples was 449,900 (409,000 - training set; 40,900 – test set) for the reduced model and 96,630,600 (87,846,000 – training set; 8,784,600 – test set) for the full-scale model.
Simulations for the reduced model were conducted on a regular desktop computer, which took approximately 20 days to complete. However, performing all necessary simulations for a full-scale model on the same hardware would take an impractical amount of time (several years). Therefore, it was decided to use a High-performance computing (HPC) environment to accelerate the simulations. To take advantage of HPC, the simulations were parallelized using Python multiprocessing. The use of HPC made it possible to generate the full-scale database in a reasonable amount of time (two months).

The data collection process for machine learning training in the proposed approach is a one-time procedure (for one climate and building type) required for the development of a surrogate model. Therefore, end-users will not need to repeat this time-consuming and laborious process.

**Surrogate model development**

Once the design space sampling is completed for the parametric model of a zone and all simulations are executed based on the sampled data, the subsequent step involves training the surrogate model. A surrogate model is an approximate mathematical model that can represent the behavior of an existing algorithm or real-world system (Fang et al., 2005), in the case of this study surrogate model approximates the behavior of a simulation engine for a single zone.

A feed-forward multilayer perceptron architecture of ANN was chosen as the machine learning model for surrogate modeling. This type is the most commonly used for meta-modeling in studies related to BPS and proved to be effective for building performance prediction (Roman et al., 2020), as it is capable of modeling non-linear relationships between input and output variables. ANN model was developed using the PyTorch library. The hyperparameters were selected individually for each surrogate model using a grid search strategy with a limited number of combinations (Table 2).

A separate ANN was trained for each subset of the design space. K-Fold Cross-Validation was used while training (with K depending on the size of a subset). The accuracy of each individual model assessed with a test set varied, depending on the complexity of the subset. However, the medium average error (MAE) for all models remained below 0.55 kWh/m2 year. It is important to note that these ANNs were specifically designed to predict the performance of individual zones. The validation of the full surrogate model's performance as part of a complete BPO workflow developed during the research will be discussed in further detail in the Results section.

**Table 2: Surrogate model hyperparameter options.**

<table>
<thead>
<tr>
<th>Hyperparameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimizer</td>
<td>SGD, Adam, Adagrad, Adadelta</td>
</tr>
<tr>
<td>Activation Function</td>
<td>ReLU, Tanh, Sigmoid, Softmax, LeakyReLU, ELU</td>
</tr>
<tr>
<td>Alpha Value</td>
<td>0.001, 0.005, 0.01, 0.02, 0.05, 0.08, 0.1</td>
</tr>
<tr>
<td>Number of Layers</td>
<td>1, 2, 3</td>
</tr>
<tr>
<td>Neurons per Layer</td>
<td>40, 60, 80, 100, 120</td>
</tr>
<tr>
<td>Number of Epochs</td>
<td>100, 200, 300, 400, 600, 800</td>
</tr>
</tbody>
</table>

Model-to-data conversion algorithm development

To utilize the proposed approach in practice, a conversion algorithm was developed to automate the process of converting a model provided by a user or controlled by an optimization algorithm into the necessary data for a zone-based surrogate model. A manual conversion would hinder the interaction between the surrogate model and a user’s parametric model, making it unsuitable for BPO.

In order for the surrogate model to predict a building's performance, the building needs to be split into separate thermal zones and each zone needs to be converted into a set of inputs similar to those used to train the surrogate model. However, the parametric description of the plan configuration used in the study was limited to only two parameters (length and width), resulting in the need to simplify more complex-shaped zones into rectangles for the surrogate model to predict their performance. Additionally, parameters related to the walls in the original zone, such as the window-to-wall ratio (WWR), shade size, context-to-sky ratio (CSR), and wall boundary condition, must be distributed among the walls in the simplified zone. It is crucial to ensure that such simplification maintains the simulation result for both the original and simplified zones unchanged or very close, guaranteeing the accuracy of the predictions.

M2DC algorithm's work involves the following stages:

1. Dividing the entire building into separate thermal zones;
2. Simplifying each zone into a rectangular shape;
3. Converting each simplified zone into the input data necessary for the ML algorithm to operate.

Various options of the algorithm were evaluated during the study, and the results were compared to determine the most effective method. The following section describes the simplification steps of this method and their descriptions, along with illustrations for three cases (Figure 4, a):

1. To proceed with the algorithm, each wall of a thermal zone is represented as a dataset that includes, among other things, a back ray (from the wall's clockwise start point, directed parallel to the wall, away from it), a forward ray (from the wall's clockwise end point, directed parallel to the wall, away from it), and additional area (an initially zero parameter used to account for distributed area from appendices) (Figure 4, b);
2. For each corner of a zone, the direction of the rays of the walls starting at this corner is analyzed. If both rays intersect other walls of the zone, they are used in the next step (Figure 4, c);
3. Each ray from the previous step cuts off two appendices from the main area of the zone. The areas of these appendices are compared, and the smallest of them is chosen (Figure 4, d);
4. The smaller appendix is cut off from the zone, and its area is saved for the next step (Figure 4, e);
5. The wall from which the cutting ray originated is updated, and the area of the appendix is added to the "additional area" in the wall dataset. The wall geometry is modified to create a new closed zone shape. If the updated wall adjoins another wall parallel to it, these two walls are merged into one, and their "additional areas" are added up. New forward and back rays are created to match the new wall geometry (Figure 4, f);
6. Steps 2 to 5 are repeated until there are no corners with both rays intersecting other walls (Figure 4, g);
7. For the new zone shape, the bounding rectangle with the minimal possible area is created (Figure 4, h);
8. The areas that belong to the bounding rectangle but not to the remaining zone shape are found. Such areas are added to the zone shape, and the walls adjacent to those areas are modified. The geometry of the walls is modified to match the bounding rectangle, and the included areas' sizes are subtracted from the "additional area" of the corresponding walls (Figure 4, i). If several walls adjoin such an area, then the area size is distributed between the walls of the bounding rectangle depending on the direction and length of the original walls (Figure 4, j);
9. The calculated "additional areas" for each side of the resulting bounding rectangle are added to the zone shape geometry (Figure 4, k);
10. The parameters of each wall from the original shape geometry are distributed between the four walls of the resulting geometry, depending on their orientation (Figure 4, l). The resulting rectangle's orientation deviation from the north will not affect the distribution, since the room orientation is defined as a separate parameter (Figure 4, m);

![Figure 4: Model-to-data conversion algorithm (M2DC): (a) example cases; (b) represent each wall as data collection; (c) analyse zone corner; (d) compare appendices areas; (e) cut smaller appendix; (f) update wall data; (g) repeat steps 2 to 5 for all appendices; (h) find a minimal bounding rectangle; (i) distribute differences; (j) distribute differences, additional case; (k) finalize the simplified shape; (l) distribute wall parameters; (m) bounding rectangle orientation parameter.](https://doi.org/10.26868/25222708.2023.1330)
To analyze the environment of each zone two parameters were used - CSR and shading length. The M2DC algorithm employs rays to determine these parameters. For CSR, three rays are directed horizontally from the center of a window at angles of -30, 0, and 30 degrees. When these rays intersect with other geometry, the algorithm determines the height of the geometry from the intersection point and the length from the window to the intersection point for each beam. The CSR is then calculated as the part of the sky dome that is closed off by the context, based on the average values from the three rays. Similarly, shading length is determined by sending a ray vertically from the center of a window, and when it intersects with other geometry, the algorithm determines the length of the geometry from the facade of the zone and the height from the center of the window to the geometry. While this approach is not the most accurate, it was chosen for its speed (Figure 5).

**Figure 5: M2DC. Environment analysis.**

**Full workflow construction**

By combining the M2DC algorithm with the developed zone-based surrogate model, a user-friendly workflow for cross-project BPO was created. In combination with an optimization algorithm (e.g., Non-Dominated Sorting Genetic Algorithm (NSGA-II), which was implemented in the case study using the Grasshopper plugin for Grasshopper) this approach will enable architects and engineers to efficiently explore the design space and determine near-optimal solutions for their projects. To construct the full workflow, the Grasshopper environment was chosen. The M2DC algorithm and the surrogate model, which were developed in Python, were imported into the Grasshopper environment using the Gh_CPython plugin. To use the surrogate model, which is based on the PyTorch library, the algorithm was deployed on a local server using the Flask library. This enables the surrogate model to reference the pre-deployed library on the local server for each prediction, saving significant time for BPO tasks that require many predictions.

**Results**

To evaluate the proposed workflow, a test case of a parametric model of a typical floor in a residential building in Tel Aviv (Figure 6) was used. The objective was to determine the possibility of finding near-optimal solutions for the design space and to assess the accuracy of the developed zone-based surrogate models and the workflow as a whole. Three tests were conducted. The first test used a reduced surrogate model and a limited design space with fixed building orientation, wall and window structures, floor height, and no context. The second and third tests utilized a full-scale surrogate model and the complete design space of the parametric model. The third test considered the context of the building (Figure 7).

**Figure 6: Three solutions from the design space of a test-case parametric model without context.**

**Figure 7: A solution from the design space of a test-case parametric model with context.**

For each of the tests, the NSGA-II optimization algorithm was employed to perform two optimizations: one using simulations, and the other using the surrogate model. The objectives of the optimization were to find Pareto Front optimal solutions that balance the largest possible glazing area and the highest energy efficiency of the building, using the combined heating and cooling loads and the total glass area of the floor as objectives. The optimization algorithm was connected to the parameters of the test-case parametric model. Simultaneously, the energy model was processed by the M2DC algorithm, which subdivided the model into zones and transformed them into parameters for input into the zone-based surrogate model. The surrogate model then generated predictions for each individual zone and utilized these predictions to calculate the overall performance of the entire floor. Finally, the resulting performance value was fed back into the optimization algorithm (Figure 2). For all the tests Identical hyperparameters for the optimization algorithm were set (40 generations with 50 instances, crossover probability 0.9, mutation probability 1/n, crossover distribution index - 20, mutation distribution index - 20). The optimization using surrogate models was around 2.6 times faster on average, with most of the time being spent on the work of the M2DC algorithm. This means that surrogate model-based optimization would take almost the same time even if multiple objectives that require simulations were used. All 2000 typical floor design instances used for prediction during each surrogate model-based optimization were simulated with Energy Plus to establish the accuracy of the predictions made. The comparison of the Pareto Front solutions obtained with surrogate model-based optimization and simulation-based optimization for all the tests showed that the near-optimal solutions found using the proposed approach are close to those obtained with the simulation-based optimization (Figure 8, top), which confirms the effectiveness of this approach for BPO. The comparison of predictions and simulations showed that the prediction accuracy decreases with the increasing complexity of the
design space (Figure 8, bottom). For the first test, the maximum error was 1.7 kWh/m² year (EP simulation: 51.73; Prediction: 50.04; 3.27% error) with a mean absolute error (MAE) of 0.8 kWh/m² year and $R^2 = 0.9928$, indicating a high degree of accuracy. However, for the more complex design space, the accuracy of predictions dropped. Second test: maximum error - 4.9 kWh/m² year (EP simulation: 38.09; Prediction: 42.99; 12.86% error); MAE - 1.2 kWh/m² year; $R^2 = 0.9765$.

Third test: maximum error - 5.0 kWh/m² year (EP simulation: 29.87; Prediction: 34.83; 16.6% error); MAE - 1.2 kWh/m² year; $R^2 = 0.8953$.

Discussion

The study results demonstrate that the proposed workflow enables the identification of near-optimal solutions for a user-developed parametric model's design space efficiently and quickly. In contrast to earlier methods proposed by researchers, this study presents the development of a versatile surrogate model that can be readily applied to both early and advanced stages of design in various projects, eliminating the need for additional ML training. By eliminating the time investment and the requirement for ML expertise, this approach enhances accessibility for architects and designers. However, the tests indicate that the developed surrogate model may lack the necessary accuracy to predict individual instances of the studied design space, particularly for models with context. To enhance the proposed algorithm's interaction with the context, more precise but time-consuming context parameters, such as sky exposure, may be employed instead of those used in this study (shade size and CSR). Moreover, to improve the accuracy of the predictions in general, more advanced ANN architectures and a wider range of hyperparameters can be investigated, or the performance of other ML models can be compared in the context of the proposed approach. While the proposed approach has the potential to enhance the BPO process, several limitations should be acknowledged. Firstly, the cross-project surrogate model was developed using data from one specific climate region (Tel Aviv, Israel) and for one building type (residential). As a result, the developed model is not efficient for cases out of this scope. To expand the scope of the application weather data can be used as input parameters for a surrogate model, as proposed in other studies (Rackes et al., 2016; Vazquez-Canteli et al., 2019; Westermann et al., 2020) to adapt the model to different climatic zones. However, it is important to note that adding more parameters to the surrogate model may decrease the quality of predictions. It is also important to note that the parameters used to develop the surrogate model were limited to a specific range, and any zones with parameters outside of this range may lead to inaccurate predictions. Nevertheless, the range selected for the study should cover most typical cases.

![Figure 8: Comparison of Pareto Front solutions obtained with surrogate model-based optimization and simulation-based optimization (top) and regression between ANN prediction results and simulation results (bottom) for 3 tests: (a) reduced surrogate model; (b) full-scale surrogate model, no context; (c) full-scale surrogate model with context.](image-url)
Conclusion

In this study, a quick and efficient cross-project workflow was proposed for finding near-optimal solutions for the design space of user-developed parametric models using a zone-based surrogate model. The workflow was tested on a parametric model of a typical floor in a residential building in Tel Aviv, and the results showed that the proposed method can significantly reduce the time and effort required to quickly explore the design space and find near-optimal solutions. This will provide architects and engineers with an efficient and reliable tool for BPO that can significantly reduce the time, resources, and knowledge required for the improvement of the sustainability and energy efficiency of buildings, which may lead to more efficient building design processes, with the potential to reduce energy consumption and greenhouse gas emissions in the building sector.

References


