Research Article

L-SCOPE: An LLM-Assisted Interactive Platform for Efficient Building Design Optimization

Qianyun Zhou[1], Fan Xue[2*]

1 PhD student, Department of Real Estate and Construction, The University of Hong Kong
2 Associate Professor, Department of Real Estate and Construction, The University of Hong Kong
*Correspondence: xuef@hku.hk

Abstract

The building and construction sector accounts for a significant share of global energy use and greenhouse gas emissions, highlighting the importance of focusing on sustainable design for climate change mitigation. Early-stage design decisions are crucial in shaping a building's life cycle energy performance. However, sustainable building design exploration often relies on simulation-based optimization (SBO), of which complex objectives and numerous design parameters result in high computational complexity and limited user understanding of how individual parameters influence the outcome. This study introduces L-SCOPE, a large language model (LLM)-assisted interactive platform designed to enhance the interpretability and efficiency of complex SBO. The platform provides an end-to-end workflow that includes environment setup, information-theoretic analysis, and automated interval refinement of design parameters. Stakeholders directly participate in the analytical process through a prompt-driven, multi-turn dialogue interface, where L-SCOPE facilitates seamless human-AI interaction by implementing natural language prompts, generative code reviews, and interpretable visual feedback. A case study demonstrates that the platform achieved robust and reliable results, with a tenfold reduction in convergence analysis time and a 0.49% improvement in energy performance compared to conventional SBO methods under the same computational budget. The contributions of this study are twofold: (i) a novel approach that combines LLM capabilities with information-theoretic analysis to advance interpretable and efficient building design optimization; and (ii) an interactive language-based interface that facilitates active participation and decision-making by non-expert stakeholders.

Keywords: Large Language Model (LLM); Human-AI Interaction; Sustainable Design Optimization; Parameter Converging Tool; Design Decision Support

Highlights

- A novel LLM-assisted platform integrating information-theoretic analysis with SBO.
- Enhances interpretability and stakeholder participation via interactive dialogue interface.
- Achieves 10× faster convergence analysis time with 0.49% energy performance improvement under same budget.

1 Introduction

The building and construction sector significantly contributes to global climate change, accounting for approximately 34% of global energy consumption and 37% of greenhouse gas emissions (UNEP 2024). The increasing adoption of sustainable design practices in the building sector has demonstrated that early-stage performance-driven design plays a critical role in shaping long-term energy efficiency and life-cycle environmental impact, thereby helping to mitigate environmental burdens (Wang et al. 2024; Zhan et al. 2024).

Simulation-based optimization (SBO) has become a widely used approach for performance-driven building design. By integrating parametric modeling and performance simulations, SBO enables iterative exploration of design alternatives and evaluation of parameter impacts on performance objectives (Luca et al. 2024). However, SBO in early design often involves numerous parameters and objective functions with black-box characteristics, leading to a complex design search space that increases computational demand and limits the interpretability of design decisions (Zhou & Xue 2025). Conventional optimizers struggle to approximate near-global solutions within limited iterations, underscoring the need for more efficient strategies to guide and shrink the design search space under constrained computational budgets.

Mainstream strategies employed surrogate models to approximate the design search space and sensitivity analysis to identify important design parameters (Danhaive & Mueller 2021; Hinkle et al. 2024; Tian et al. 2024). However, both techniques rely heavily on extensive simulation datasets and often require re-execution for different optimization tasks, resulting in significant time and labour costs. To address these limitations, Zhou and Xue (2025) introduced the Automatic Information Gain-guided Convergence (AIGGC) method, which instead employs small-scale historical optimization data to perform information gain-guided analysis of sub-intervals within each design parameter. The AIGGC method reduces the complexity of the design search space and accelerates convergence while enhancing interpretability. Nevertheless, the absence of an integrated, visual, and interactive platform has limited its broader application in practice.

Large language models (LLMs) have shown considerable potential in natural language understanding and code generation (Fang et al. 2024). By lowering the barrier between domain expertise and computational modelling, LLMs enable non-expert users to specify complex analytical tasks without extensive programming skills. In architecture, engineering, and construction (AEC), recent studies have applied LLMs to support more interactive and accessible human-AI collaboration. These applications are typically grouped into three categories: information retrieval (Guo et al. 2025; Li & Wang 2025), where LLMs translate natural-language queries into building or BIM semantic data; task automation (Gao et al. 2025; Zhang et al. 2025), where LLM-based agents convert user requirements into executable modelling workflows; and interpretability enhancement (Zhang & Chen 2024), where LLMs are combined with interpretable machine learning to clarify rule-based rationale. However, the applications of LLMs in complex SBO tasks remain limited.

This study presents L-SCOPE, an **L**LM-assisted interactive platform for design **S**pace **C**onvergence and **O**ptimized **P**arameter **E**xploration. Operating within the Grasshopper ecosystem, L-SCOPE seamlessly integrates with existing SBO tools. Through a prompt-driven, multi-turn dialogue interface, users can issue natural language queries, customize analysis indices, and review editable code before execution. This human-in-the-loop process mitigates risks of hallucination, improves interpretability through visualization, and preserves user control. The platform integrates optimization outputs, design parameter analysis, and automated domain convergence into an interoperable workflow, extending conventional SBO with interpretable, stakeholder-oriented insights and AI-assisted decision support.

The contributions of this study are twofold: (i) an integrated approach that combines LLM capabilities with information-theoretic analysis to enhance the interpretability and efficiency of building design optimization; and (ii) an interactive language-based interface that facilitates active participation and decision-making by non-expert stakeholders.

2 Related works

In early-stage sustainable building design, designers generally adopt *Rhino-Grasshopper* workflows with SBO tools to explore and evaluate design alternatives (Luca et al. 2024). Common plugins such as *Ladybug* for daylighting, *Honeybee* for energy simulation, and various optimization solvers enable automated design iterations. Popular optimizers, including Genetic Algorithm (GA), CMA-ES, and RBFOpt-based methods, run predefined iterations and output the best-performing solution (Brown et al. 2020). Since the objective functions are black boxes computed by external simulation engines using climate and physical models, direct mathematical formulations are infeasible. Combined with numerous design parameters in early stages, the black-box nature of SBO results in complex search spaces that limit interpretability and computational efficiency (Zhou & Xue 2025).

Recent research has explored deep generative models and surrogate-based methods to approximate complex search spaces and support exploration of specific design regions (Brown et al. 2020; Danhaive & Mueller 2021). Others have applied metamodels for real-time interactive optimization and sensitivity analysis to identify influential design parameters (Hinkle et al. 2024). However, most sensitivity-based methods need large-scale sampling, which is impractical for time-intensive simulations. Similarly, machine learning-based feature selection requires retraining when design parameters or ranges change, increasing computational costs. Moreover, the problem-specific nature of surrogate models limits the generalizability across different design scenarios, while their predictive accuracy tends to degrade in high-dimensional or sparsely sampled design spaces (Zhou & Xue 2025).

The AIGGC method addresses these limitations by reallocating a small portion of the optimization budget to perform information gain analysis on design parameters' sub-intervals (Zhou & Xue 2025). The method reduces search space complexity, accelerates optimization convergence, and improves SBO interpretability, with demonstrated robustness across different scales and optimizers. However, the absence of an integrated, visual, and interactive platform, particularly within the *Rhino-Grasshopper* ecosystem, limits accessibility for practitioners, making it difficult for users to understand and apply the method effectively.

3 L-SCOPE: LLM-assisted design Space Convergence for Optimized Parameter Exploration

L-SCOPE is an interactive platform designed to enhance the efficiency of complex SBO in early-stage design by integrating an LLM-assisted information-theoretic analysis workflow.

3.1 L-SCOPE workflow and components

As illustrated in Figure 1, the platform operates within the *Rhino-Grasshopper* environment. The workflow begins with an initial SBO run, where stakeholders define design parameters, performance objectives, and an initial iteration budget (N_1) using relevant Grasshopper plugins for modelling, simulation, and optimization. The resulting dataset (e.g., *.csv*) is transferred to L-SCOPE, which interprets stakeholder prompts and generates executable Python code for information-theoretic analysis. Based on information gain rankings, L-SCOPE guides stakeholders in refining design parameters' domain intervals. The refined domains are then applied in a second SBO run under the remaining budget (N_2), thereby converging the search toward optimal building design. Each LLM-generated script was cross-validated with a manually implemented reference during development to ensure correctness and reproducibility. In practical use, the platform returns the generated code to the user interface for human inspection before execution, reducing hallucination risks and preserving user control.

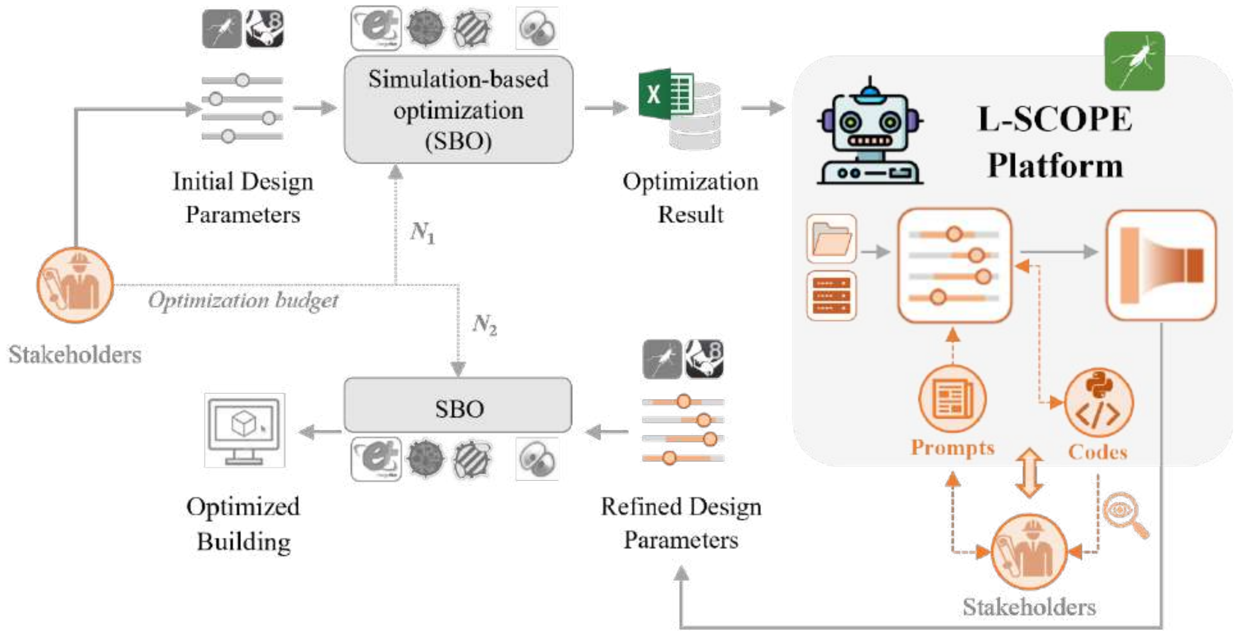


Figure 1. The proposed workflow for L-SCOPE.

As shown in Figure 2, L-SCOPE integrates four Grasshopper components into a three-step pipeline covering environment setup, interactive information-theoretic analysis, and automated refinement for design space convergence. Step 1 establishes the local execution environment and backend server, including components for creating a virtual environment with Python dependencies and launching the LLM server. Step 2 is the core interactive module, where a control panel enables users to issue natural language prompts, review LLM-generated Python code and execution feedback, and visualize key results such as elite probability distributions of sub-intervals and information gain rankings. Based on these outputs, Step 3 automatically refines the sub-intervals of selected design parameters, adapts input ports to match the detected parameters, and outputs updated domain intervals with user-defined precision for subsequent optimization.

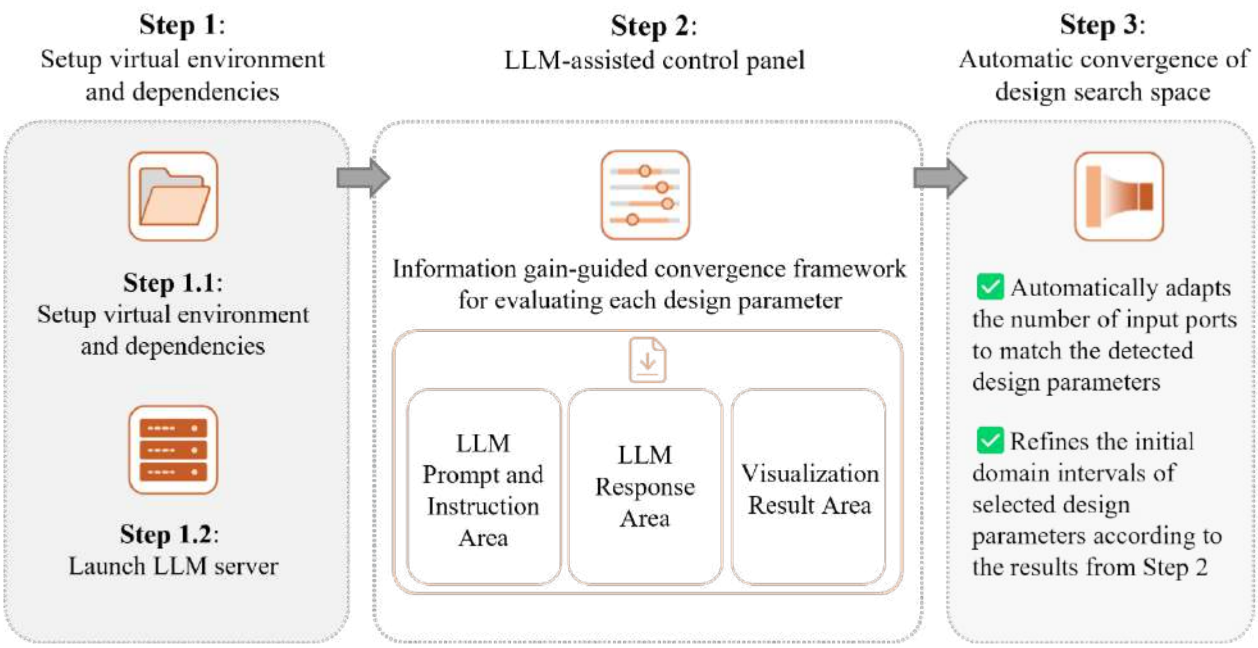


Figure 2. The four components of L-SCOPE.

3.2 L-SCOPE interactive panel

Figure 3 displays the layout of the interactive panel within L-SCOPE. The platform supports multi-modal outputs, including Python code, console messages, chat history, structured data, and visualizations. After loading the dataset, the panel displays a parallel coordinates plot to help users inspect the distribution of design parameters. On the left, analysis factors can be specified through sliders or natural language prompts. Intermediate results are shown through console outputs and visual feedback, ensuring transparency and traceability. The final output is a structured summary of key information, such as the sub-intervals targeted for refinement and design parameters classified as constant.

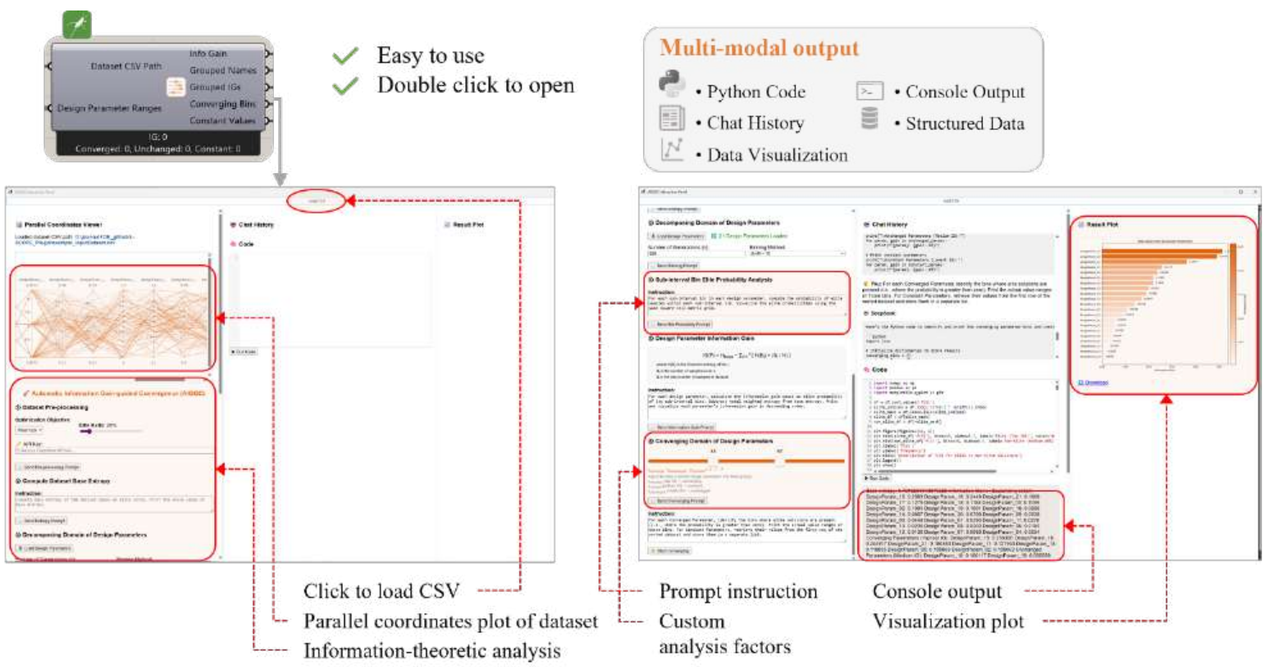


Figure 3. Details of L-SCOPE interactive panel.

4 Experimental results

4.1 Case study and experimental settings

To evaluate the effectiveness of the proposed platform, a case study was conducted based on the published work of Zhou and Xue (2023). The study focused on a Modular Integrated Construction (MiC) project for a university student hostel located in Hong Kong. In this case, the optimization objective was to minimize the Energy Use Intensity (EUI), defined as the total annual energy consumption divided by the gross floor area (kWh/m^2·yr). The EUI value was calculated via the *Honeybee-Radiance* (*Ver.1.8*) and *Honeybee-OpenStudio* (*Ver.1.8*) components, with a sensor grid size of 0.5 meters. As shown in Figure 4, the project features a symmetrical layout in which modules B^w, C^w, and C^t remain fixed throughout the study, and modules M_A, M_B, and M_C were reorganized during the optimization process. In addition to module layout, window size and material properties were also considered due to their significant impact on energy performance (Qin & Pan 2020). The design parameters and their corresponding domain intervals are summarized in Table I.

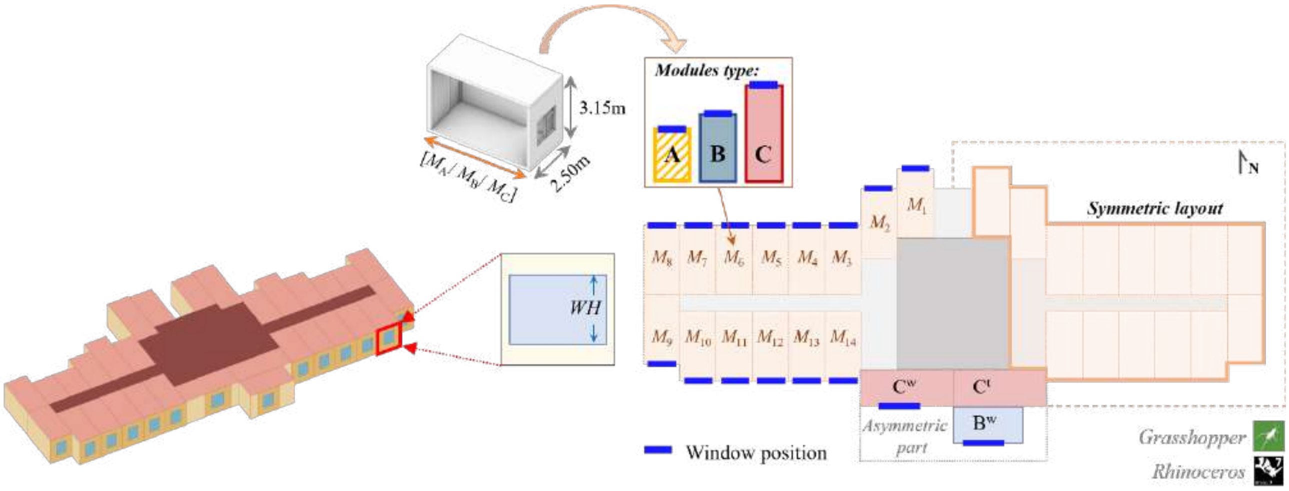


Figure 4. MiC case layout and schematic of module types.

Table I. Description of design parameters for MiC parametric design.

Design parameters	Parameter type	Unit	Parameter domain intervals	Precision
MiC Module Type (M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}, M_{14})	Categorial	-	[M_A, M_B, M_C]	-
Window-to-Wall Ratio (WWR_N, WWR_S)	Continuous	-	$0.10 \leqslant WWR_N, WWR_S \leqslant 0.60$	0.01
Window Height (WH_N, WH_S)	Discrete	m	$1.0 \leqslant WH_N, WH_S \leqslant 2.5$	0.1
Window U-value (U_W)	Continuous	W/(m^2·K)	$0.50 \leqslant U_W \leqslant 5.78$	0.01
Window Solar Heat Gain Coefficient (g)	Continuous	-	$0.100 \leqslant U_W \leqslant 0.775$	0.001
Window Visible Transmittance (τ_v)	Continuous	-	$0.300 \leqslant U_W \leqslant 0.881$	0.001

The experiments were conducted on a desktop computer with an Intel (R) Core i7-10700 CPU @ 2.90 GHz processor and 32 GB memory. In this study, *DeepSeek* serves as the backbone LLM for code generation and analysis. The experimental configuration in this study follows the settings recommended in our previous work (Zhou & Xue 2025). The total iteration budget was set to 300, divided into an initial stage (N_1=100) and a second stage (N_2=200), optimized using a genetic algorithm (GA) through the *Galapagos* component in Grasshopper. The experiments were repeated 10 times (as each run set required 36 h) to mitigate the stochastic effects of GA and evaluate the robustness of L-SCOPE. An elite ratio r_e=20% was used to classify initial-stage results into 'elite' and 'non-elite' groups. Each design parameter's domain interval was divided into 10 sub-interval bins for calculating elite probability and entropy. Based on the calculated information gain, design parameters were then ranked and categorized into three groups using two thresholds: r_{conv} and r_{const}, both set to one-third. Parameters with high information gain were refined by excluding sub-intervals with zero elite probability; those with low information gain were fixed to constant values; and the remaining parameters were left unchanged. For detailed definitions and algorithmic procedures, please refer to (Zhou & Xue 2025).

4.2 Experimental results and analysis

4.2.1 Optimized results with L-SCOPE

The experimental results, summarized in Table II, compare the performance of the Genetic Algorithm (GA) with and without the L-SCOPE platform over ten independent runs. On average, L-SCOPE achieved a lower *EUI* of 113.466 kWh/m^2·yr, compared with 114.025 kWh/m^2·yr obtained from conventional SBO, representing a 0.49% performance gain under the same computational budget. For the best-optimized solutions, L-SCOPE further reduced the *EUI* from the baseline value of 116.747 to 112.684 kWh/m^2·yr, corresponding to a 3.48% improvement. A two-tailed independent t-test yielded a statistically significant result (p = 0.019 < 0.05), supporting the robustness and reliability of the observed improvements.

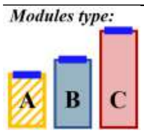
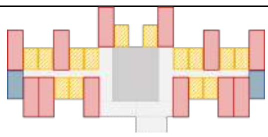
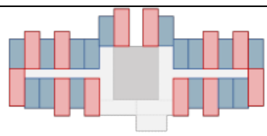
Table II. The magnitude and significance of L-SCOPE's performance in ten runs.

SBO algorithm	Objective function f (Baseline value v_0)	Average optimization results				Best solution	
		Conventional SBO	With L-SCOPE	% Gain.*	Sig.#	Conventional SBO	With L-SCOPE
GA	*EUI* (kWh/m^2·yr)	114.025	113.466			113.388	112.684
	$\Delta = f - v_0$	− 2.722	− 3.281	**0.49%**	**0.019**	− 3.359	− 4.063
	% Imp.†	2.33%	2.81%			2.88%	3.48%

†: Improvement relative to baseline *EUI*, computed as$|\Delta|/v_0 \times 100\%$, where v_0 =116.747 kWh/m^2·yr.
*: Relative performance gain of L-SCOPE over Conventional SBO in average EUI results.
#: Two-tailed p-value of independent t-test, bold when p-value < 0.05.

Table III shows a visual comparison between the baseline and L-SCOPE-optimized layouts. The optimized design obtained a 4.063 kWh/m^2·yr reduction in *EUI*, with a clearer preference for using Module B and C over A. The optimized MiC layout also exhibits a more compact and flattened horizontal profile.

Table III. Comparison between the best of optimized solutions with L-SCOPE and the baseline solution.

	Baseline model layout	Best optimized solution based on L-SCOPE
		
EUI (kWh/m^2·yr)	116.747	112.684

4.2.2 Robustness evaluation and time efficiency of L-SCOPE

To evaluate the robustness of the LLM-generated code, the platform was tested using a fixed sequence of seven prompts across ten independent runs. In all cases, the LLM reliably produced syntactically correct and executable Python code. Only minor formatting artifacts, such as Markdown-style annotations (e.g., triple backticks), occasionally appeared but were easily removed without affecting execution or reproducibility. No runtime errors or logic inconsistencies were observed, demonstrating the platform's robustness in maintaining context and variable consistency across multi-step prompt interactions. Moreover, validation against a manually implemented reference confirmed that the LLM-generated code replicated the core information-theoretic analytical logic, achieving consistent information gain rankings, accurate identification of converging design parameters, and refinement of their corresponding sub-interval domains.

As illustrated in Figure 5, L-SCOPE consistently translated sequential prompts into valid code and coherent visualizations. All ten runs exhibited uniform colour schemes and layout structures, with only minor variations in bar chart rendering. These results confirm the platform's reliability in producing reproducible, interpretable outputs through an end-to-end analysis workflow.

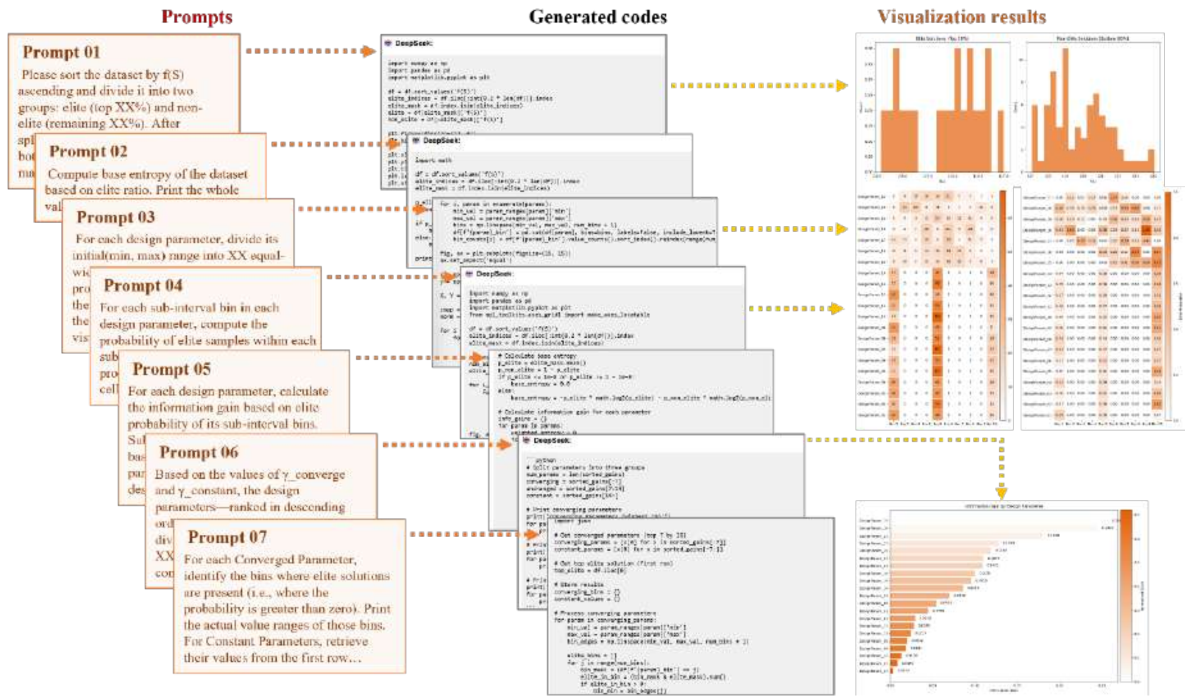


Figure 5. End-to-end outputs from sequential prompt execution in L-SCOPE.

The platform also improves analysis efficiency. As shown in Table IV, manual information-theoretic analysis typically takes 40 to 60 minutes, covering tasks such as code development, debugging, and identifying converging design parameter intervals. In contrast, L-SCOPE completes the whole process in 4 to 6 minutes, reducing analysis time by a factor of ten and accelerating early-stage design exploration.

Table IV. Estimated time range of manual and LLM-assisted analysis.

	Manual	L-SCOPE
Estimated time (min)	40-60	4-6

5 Discussion

This study presents the first implementation of the L-SCOPE platform for early-stage SBO in architectural design. The platform integrates a large language model (*Deepseek*) with an information gain-guided convergence framework to address the limitations of low interpretability and search inefficiency in existing tools. L-SCOPE enables non-expert users to participate directly in the analysis through a prompt-driven, human-in-the-loop workflow that interactively guides the refinement of design parameters' domain intervals. Compared to conventional SBO methods, L-SCOPE achieved a tenfold reduction in convergence analysis time and a 0.49% enhancement in energy performance.

Experimental results confirmed the reliability of the LLM-generated code and the robustness of analytical outputs across repeated runs.

The limitations of this study are twofold: first, it lacks comparative evaluation against alternative design space reduction techniques; second, it has not been extended to address multi-objective optimization tasks, such as daylighting, thermal comfort, or construction cost, which remain a direction for future development.

6 Conclusions

This study proposes L-SCOPE, an LLM-assisted interactive platform designed to support interpretable and efficient early-stage building design optimization. The platform consists of four Grasshopper components, enabling an end-to-end workflow that covers environment setup, server initialization, information-theoretic analysis, and automated interval refinement. By combining large language models with information gain-guided convergence, L-SCOPE facilitates seamless human-AI interaction and empowers non-expert users to engage in design space convergence through natural language prompts. Experimental results demonstrate the platform's robustness, reproducibility, and significant improvements in analysis efficiency. L-SCOPE contributes to both the enhanced interpretability and computational efficiency of the optimization process, and the promotion of broader stakeholder participation through an interactive language-based interface.

Future developments will focus on evaluating L-SCOPE's broader applicability to diverse performance objectives, benchmarking alternative language models, and validating user accessibility and decision support effectiveness through practitioner studies.

Acknowledgements

The study was supported by Hong Kong Research Grants Council (RGC) (Nos. C7080-22GF, T22-504/21-R).

Data Availability Statement

A packaged version of the L-SCOPE plugin and example Grasshopper files are publicly available at: https://github.com/Joannazhou-qianyun/L-SCOPE. The repository also contains documentation and instructions for installation and use. All data supporting the findings of this study can be reproduced using the provided example files.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Authorship Contribution Statement

Qianyun Zhou: Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Data curation, Conceptualization. Fan Xue: Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

References

Brown, N. C., Jusiega, V. & Mueller, C. T. (2020). Implementing data-driven parametric building design with a flexible toolbox. *Automation in Construction, 118*, 103252. doi:10.1016/j.autcon.2020.103252

Danhaive, R. & Mueller, C. T. (2021). Design subspace learning: Structural design space exploration using performance-conditioned generative modeling. *Automation in Construction, 127*, 103664. doi:10.1016/j.autcon.2021.103664

Fang, C., Miao, N., Srivastav, S., Liu, J., Zhang, R., Fang, R., Tsang, R., Nazari, N., Wang, H. & Homayoun, H. (2024). Large Language Models for Code Analysis: Do LLMs Really Do Their Job? *33rd USENIX Security Symposium (USENIX Security 24)* (pp. 829-846). Philadelphia: USENIX Association.

Gao, M. Y., Li, C., Petzold, F., Tiong, R. L. & Yang, Y. (2025). Lifecycle framework for AI-driven parametric generative design in industrialized construction. *Automation in Construction, 174*, 106146. doi:10.1016/j.autcon.2025.106146

Guo, P., Xue, H., Ma, J. & Cheng, J. C. (2025). Advancing BIM information retrieval with an LLM-based query-domain-specific language and library code function alignment system. *Automation in Construction, 178*, 106374. doi:10.1016/j.autcon.2025.106374

Hinkle, L. E., Pavlak, G., Curtis, L. & Brown, N. C. (2024). Implementing dynamic subset sensitivity analysis for early design datasets. *Automation in Construction, 158*, 105198. doi:10.1016/j.autcon.2023.105198

Li, M. & Wang, Z. (2025). BuildingGPT: Query Building Semantic Data Using Large Language Models and Vector-Graph Retrieval-Augmented Generation. *Building and Environment*, 113855. doi:10.1016/j.buildenv.2025.113855

Luca, F. D., Natanian, J. & Wortmann, T. (2024). Ten questions concerning environmental architectural design exploration. *Building and Environment, 261*, 111697. doi:10.1016/j.buildenv.2024.111697

Qin, H. & Pan, W. (2020). Energy use of subtropical high-rise public residential buildings and impacts of energy saving measures. *Journal of Cleaner Production, 254*, 120041. doi:10.1016/j.jclepro.2020.120041

Tian, L., Hao, T., He, X., Chan, I., Niu, J., Chan, P., Ng, W. & Huang, J. (2024). Examining the non-linear relationship between urban form and air temperature at street level: A case of Hong Kong. *Building and Environment, 264*, 111884. doi:10.1016/j.buildenv.2024.111884

UNEP. (2024). *Global status report for buildings and construction: Beyond foundations: Mainstreaming sustainable solutions to cut emissions from the buildings sector.* Nairobi: United Nations Environment Programme. Retrieved 11 28, 2024, from https://www.unep.org/resources/report/global-status-report-buildings-and-construction

Wang, X., Teigland, R. & Hollberg, A. (2024). Identifying influential architectural design variables for early-stage building sustainability optimization. *Building and Environment, 252*, 111295. doi:10.1016/j.buildenv.2024.111295

Zhan, J., He, W. & Huang, J. (2024). Comfort, carbon emissions, and cost of building envelope and photovoltaic arrangement optimization through a two-stage model. *Applied Energy, 356*, 122423. doi:10.1016/j.apenergy.2023.122423

Zhang, L. & Chen, Z. (2024). Large language model-based interpretable machine learning control in building energy systems. *Energy and Buildings, 313*, 114278. doi:10.1016/j.enbuild.2024.114278

Zhang, L., Fu, X., Li, Y. & Chen, J. (2025). Large language model-based agent Schema and library for automated building energy analysis and modeling. *Automation in Construction, 176*, 106244. doi:10.1016/j.autcon.2025.106244

Zhou, Q. & Xue, F. (2023). Pushing the boundaries of modular-integrated construction: A symmetric skeleton grammar-based multi-objective optimization of passive design for energy savings and daylight autonomy. *Energy and Buildings, 296*, 113417. doi:10.1016/j.enbuild.2023.113417

Zhou, Q. & Xue, F. (2025). Automatic information gain-guided convergence for refining building design parameters: Enhancing effectiveness and interpretability in simulation-based optimization. *Building and Environment, 275*, 112788. doi:10.1016/j.buildenv.2025.112788