2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

# TOWARDS AI-ENHANCED FACADE PLANNING: INTEGRATING HUMAN EXPERTISE WITH MACHINE LEARNING-DRIVEN PARAMETRIC MODELING

Simon K. Hoeng[1], Jonas Wiederer[1], Friedrich Eder[1], Thomas Linner[1], and Mathias Obergriesser[1]
[1]Ostbayerische Technische Hochschule Regensburg, Regensburg, Germany
Corresponding author: simon.hoeng@oth-regensburg.de

## Abstract

Planning modern facade systems is complex, requiring optimization across multiple domains. This paper proposes an AI-enhanced workflow for facade planning, harnessing Computer Vision and human input via a Large Language Model. A generative AI system then guides a parametric model to produce 3D facade designs. Automated checks provide feedback to a Reinforcement Learning system, to iteratively determine optimal solutions. These solutions are verified and finalized by human expertise, ensuring improved outcomes with reduce planning time and effort.
The approach illustrates how linking advanced AI methods with human expertise can address the multifactorial challenges of facade design within current industry practices.

## Introduction

Building facades serve as the exterior envelope of buildings, significantly influencing both their aesthetic representation and overall energetic performance. Mahdavinejad et al. (2024) emphasize that facade design directly affects a building's energy efficiency and occupant comfort. Their research demonstrates that optimized facade designs can lead to substantial reductions in energy consumption and improvements in indoor environmental quality. Conventional planning processes in the Architectural, Engineering, and Construction (AEC) industry often exhibit limitations due to iterative solution-finding approaches and suboptimal structural choices. Elaborated automated design approaches offer significant potential for reducing costs and improving performance throughout both the planning and operational phases of a project. Artificial Intelligence (AI) poses a solution to partly automate and speed up such decision-making processes by proposing and evaluating a broader range of design options within a short time frame. By expanding the decision-making basis and providing more informed choices, these technologies can improve the outcomes of conventional solution-finding processes (Mohammadpour et al., 2019). Therefore, developing tools, workflows, or frameworks that enhance facade design processes by leveraging AI methods can have a significant impact on both environmental and financial factors.

## State of the Art

In scientific literature, there is a considerable number of publications describing the potential applications, implementation strategies, and technical aspects of the aforementioned AI technologies for the AEC domain. Abioye et al. (2021) conducted an extensive literature review on the status, opportunities, and challenges of AI applications in the AEC industry. Their review identifies a significant number of publications, categorized into key application domains such as Computer Vision (CV), Natural Language Processing, or optimization. Among the identified categories, optimization is by far the largest, focusing on the use of AI to make decisions or choices that achieve the best possible outcomes under defined constraints. The prevalence of studies in this category highlights the potential researchers see in AI's ability to address complex decision-making challenges in the AEC domain by leveraging the growing availability of machine-readable data.

### Related Work

The methodology described in this work employs and links multiple AI-related technologies. Selected papers that are relevant in this context are summarized below. CV applications have gained widespread attention across industries, including the AEC sector, demonstrating effective automation in image-based automation tasks, as Xu et al. (2021) showed in their systematic review. Stoitchkov et al. (2019) introduced a framework for plan analysis based on Convolutional Neural Networks, aimed at automating the detection and classification of symbols in technical drawings. To address the challenge of limited training data for this AEC-related CV use case, they proposed a synthetic data generation pipeline that creates training images by compositing symbols onto randomized backgrounds extracted from technical plans. Large Language Models (LLMs) can provide a promising new type of interface for complex task management, simplifying user interaction by enabling both information extraction and content generation within intricate systems. Recent developments have explored the integration of LLMs into BIM environments to enhance information accessibility and reduce engineering overhead. Zheng et al. (2023) propose a dynamic prompt-based virtual assistant framework, which leverages generative pre-trained transformer technologies

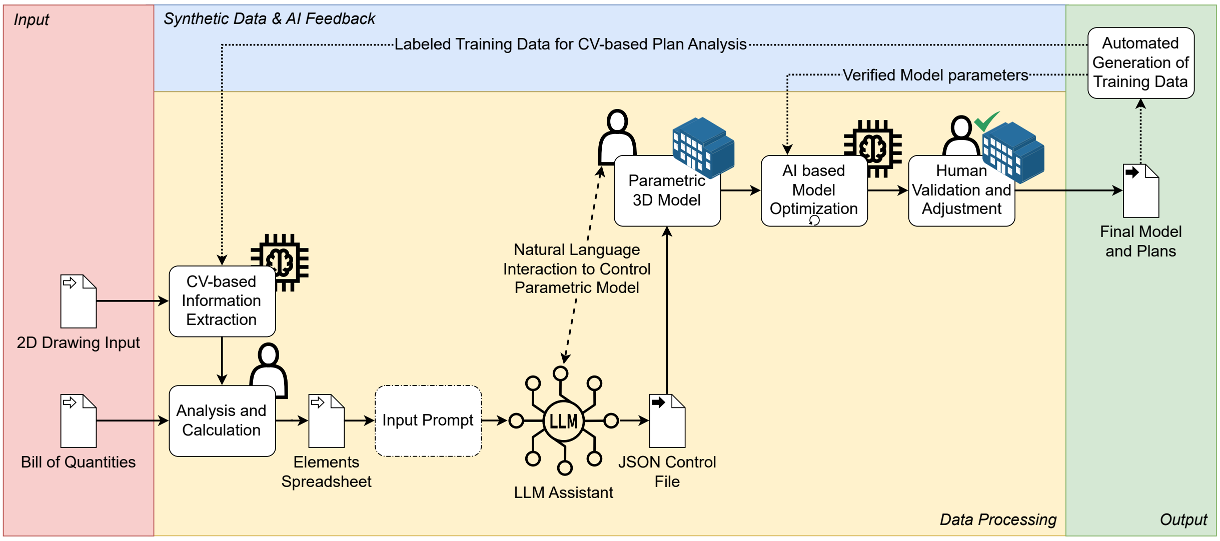


*Figure 1: Methodological overview of the AI-enhanced, closed-loop planning workflow, illustrating how modular subprocesses, human validation, and feedback-driven optimization are integrated into a cohesive framework*

to support natural language based search within BIM systems. The approach enables users to query BIM data in natural language and receive relevant information along with 3D visualizations, thereby lowering the barrier for interacting with complex building data. Complementing this approach, Du et al. (2024) focus on the generative side of LLM-based BIM interaction by introducing an agent framework capable of autonomously performing modeling tasks. Embedded into a BIM authoring tool, the agent interprets user intentions expressed in natural language, selects appropriate tools, and executes modeling workflows. Their findings highlight the promise of LLM-based agents in supporting intelligent design automation. However, they also report challenges, particularly the risk of hallucinations—where the agent invokes incorrect or unreliable actions.

Reinforcement Learning (RL) is emerging as a powerful AI technology for iterative optimization in AEC, particularly for addressing complex, multi-objective design and construction challenges. For instance, Elmaraghy et al. (2023) proposed an RL framework inspired by real-time strategy games—where resource planning and long-term strategic coordination are crucial—to autonomously optimize building design and construction processes in a physics-based, game-engine environment. Model-based data workflows with integrated verification and optimization mechanisms are becoming increasingly prevalent in the AEC industry. Shen and Pan (2023) proposed a systematic framework combining advanced BIM techniques with intelligent algorithms (including Bayesian optimization and explainable machine learning) to predict and optimize building energy performance for sustainable development. Ahmad et al. (2024) are exploring an AI application for optimizing a parametric shading structure to retrofit mid-rise residential buildings in Abu Dhabi for enhanced energy efficiency. Using tools like Grasshopper, EnergyPlus, and genetic algorithms within the Galapagos plug-in, the research developed an advanced optimization model. The optimization was successfully applied to a case study, leading to improved energy performance regarding the required cooling loads.

### Scientific Contribution

Despite the growing body of research on AI in AEC, there remains a gap in studies that fully leverage AI-based synergistic approaches for enhanced automation in facade planning. While prior studies have focused mainly on isolated AI technologies, this research proposes a holistic and application-oriented methodology that links multiple AI techniques to achieve a high level of automation, while incorporating human expertise as both a quality control mechanism and an accelerator. The resulting continuous digital pipeline transforms fragmented input data such as drawings, tender documents, and process requirements into an optimized building model. The paper presents a pragmatic implementation and integration of CV, LLMs, and RL within a human-in-the-loop framework, thereby contributing a conceptually cohesive, self-improving, closed-loop method to address the labor-intensive nature of current facade design practice. By linking multiple modular AI technologies through simple control structures, this paper demonstrates a realistic and applicable perspective on how AI can be incorporated into existing workflows, even in resource-constrained environments.

## Methodology: AI-Enhanced Parametric Model Framework

The methodology depicted in Figure 1 presents a conceptually structured, modular workflow that integrates distinct AI technologies into a cohesive and partially automated process for model based planing and optimization in the AEC domain. It builds on decomposing the overall process into discrete subprocess, that are supported with suitable AI technologies, that offer the most leverage in the respective subprocess. A key feature is the inclusion of human validation at critical points to ensure quality and a reliable data flow. This means that each subprocess can be improved and applied independently, as human validation and targeted adjustments ensure the correctness of input data at each stage. In the AEC industry, where projects are highly variable and domain-specific expertise is indispensable, the integration of human knowledge into AI systems is a crucial enabler for successful implementation. Human feedback also enables the generation of validated training data, which is essential for improving data availability, a key factor in developing reliable ML-based systems. By providing this data as feedback, the workflow develops into a self-improving, closed-loop system that becomes more robust over time simply through continued use. Rather than aiming for full automation from the outset, the conceptual model emphasizes complementarity between human expertise and ML-supported decision-making, enabling iterative design optimization while ensuring compatibility with existing industry practices through the possibility of modular implementation.

### Input information and preprocessing

The methodological workflow is applicable at the stage of a tender design, where technical details require further engineering refinement to ensure the design is buildable and suitable. 2D architectural drawings serve as a descriptive representation of the intended facade design. In contrast to a manual analysis process, where facade elements are identified and matched to standard components by a human planner, the proposed workflow uses a CV system to automate this process. It extracts semantic and geometric information to identify catalog-compatible parts as well as parts that need to be custom tailored. The process is divided into two basic steps, which are exemplarily illustrated in Figure 2: detecting and extracting specific building elements that represent certain components through object detection (providing semantic information) and extracting boundary lines of the detected elements (providing geometric information). The bills of quantities, which are typically provided alongside the 2D architectural drawings, collectively form the baseline for subsequent planning processes. With additional easy accessible information provided by the prepossessed drawing input the analysis of the bill of quantities can be crosschecked. The identified elements are listed in terms of quantity and key dimensions, and noted in a human-readable standardized spreadsheet file, serving as initial data for downstream processes. The structured data forms the basis for further automation and contains the relevant core information needed to supply a parametric model with the required parameters. This approach preserves existing standard planning procedures but improves them by incorporating CV-based information

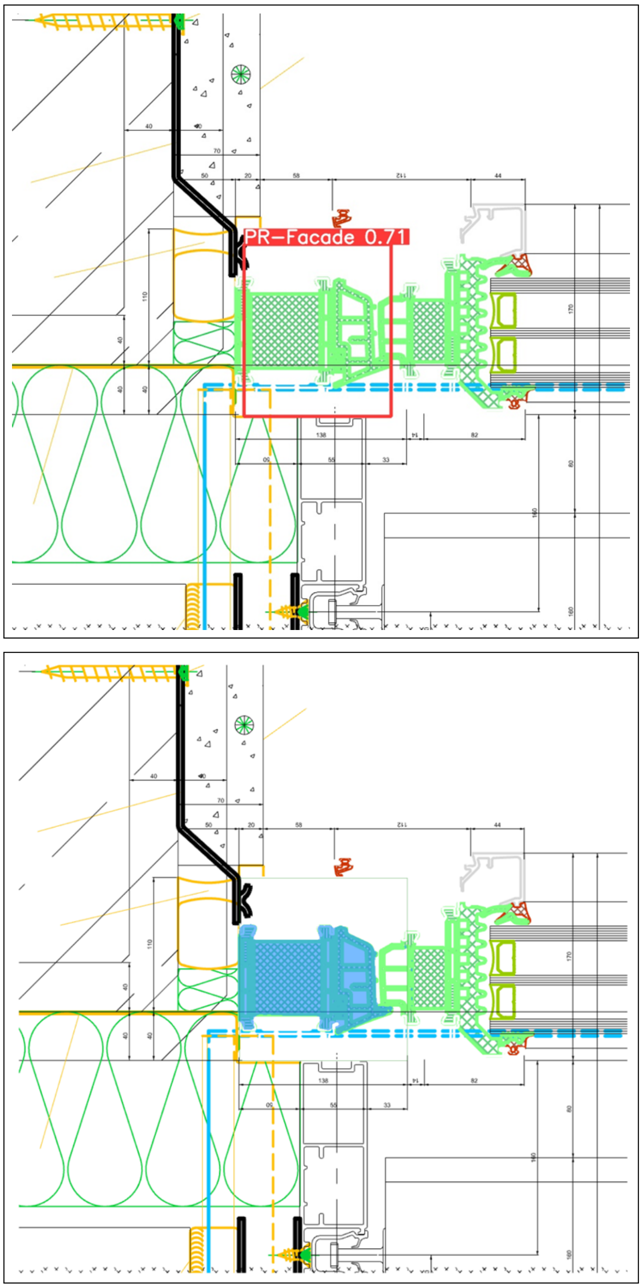


*Figure 2: CV-based plan analysis of architectural facade detail Top: 1.) Object detection of facade element with YOLOv8 Bottom: 2.) Boundary extraction with SAM*

processing, requiring no major changes in the established workflow while still allowing for human quality control.

### Human-LLM interaction for model control

The pre-processed table-based data is fed into a customized LLM designed to work with a predefined JSON data schema. The LLM not only transforms the input data into the defined JSON structure, which later controls a parametric model, but also offers the possibility to include additional specifications in natural language. Instead of manually editing the JSON control file, users can therefore issue natural language commands to refine, adjust, or extend the facade structure. The LLM interprets these commands and translates them into a valid JSON format, ensuring all modifications remain consistent with the required schema used to control the parametric facade model. For this study, a pre-prompted and few-shot-tuned GPT model was used, based on OpenAI's Custom GPTs. A state-of-the-art open source LLM, Meta's LLaMA, was also initially tested for comparison but adhered less consistently to the required output format. While LLMs can produce hallucinations, the impact of such errors is mitigated by the nature of the workflow: since the input data is not expected to be already optimal, subsequent stages include validation and refinement.

### Parametric 3D facade model

To bridge the transition from 2D planning to 3D modeling, a highly parameterized facade model was developed within Rhino Grasshopper. The model is based on an adjustable 2D facade grid, which serves as the foundational layout for placing components, as grid-like base patterns are common practice in facade planning. The JSON control file references cataloged 3D facade components, which are then loaded into Rhino Grasshopper and systematically arranged on to the 2D grid according to the provided parameters like position, dimensions, and orientation. These control parameters not only define the placement and modification of each element but also enable complete control over the model directly through the JSON file. Figure 3 illustrates a sample facade segment generated by the parametric model via such a JSON control file. This structured approach allows for straightforward interaction with the parametric model by using predefined parts and associated input data. The use of standardized components simplifies the process of creating facade models while maintaining flexibility for customization when additional input parameters for those parts are provided. The JSON-driven workflow ensures efficient integration between the initial planning stages and the automated generation of 3D facade designs.

### AI based model optimization

An automated evaluation process is integrated into Rhino Grasshopper to assess the current version of the parametric model. This evaluation provides structured feedback, forming the basis for an optimization loop that enables iterative improvements. When planning a facade system, several key factors must be considered, including thermal performance, structural integrity, material efficiency, and geometric feasibility. By leveraging automated checks directly within the modeling workflow, these factors can be systematically analyzed and optimized. Rhino Grasshopper offers powerful plugin extensions that enable automated evaluations of key performance indicators such as U-Value calculations to assess facade insulation and energy efficiency, structural load checks using tools like Karamba to verify the bearing capacity of facade elements, or geometric constraints ensuring compliance with predefined design boundaries and assembly feasibility. Additionally, it is conceivable to export the Rhino Grasshopper model automatically for external validation using specialized analysis tools, such as finite element analysis software for in-depth structural assessments, energy simulation software for dynamic performance evaluations, cost estimation tools for financial feasibility analysis, and life cycle assessment software to evaluate the environmental impact of facade materials and design choices. To fully harness the power of this AI-enhanced workflow, it is essential that the evaluation process is automated. This requires seamless integration with internal and external software tools, ideally via powerful application programming interfaces, enabling real-time data exchange and iterative optimization without manual intervention. This structured feedback can then be leveraged within an AI-based optimization loop, where RL has emerged as a powerful approach. RL can iteratively refine the design by adjusting model parameters based on reward-driven learning, progressively enhancing facade performance while considering multiple constraints. Further details on this process are demonstrated in the Validation Section of this work.

### Human validations, refinements and creation of final output

After AI-based optimization, a human reviewer engages with the resulting 3D parametric model to evaluate, refine, and finalize it. Adjustments can be made directly within the modeling environment (in this case, Rhino Grasshopper), allowing users to modify the proposed solution using familiar tools and interfaces. Since automated checks remain active within the modeling software, the human reviewer benefits from continuous feedback on design validity, supporting informed and confident decision-making throughout the refinement process. For example, a reviewer with deep process knowledge may choose to replace an automatically placed facade component that is known to cause issues during assembly. The finalized model is then used to generate downstream outputs such as technical detailing drawings, complemented with necessary annotations for production or construction. By placing final refinement and output generation in the hands of human users, the workflow ensures that potential AI-generated errors are identified and corrected. This ensures the quality of the final facade design and maintains full human control over the process.

### Generation of synthetic training data

The final results serve as a foundation for generating synthetic training data, addressing two key areas that benefit from such data. Since the parametric model is built up from cataloged building parts, loaded in Rhino Grasshopper, the information about which geometry corresponds to which building element can be passed back to the CV system. Thereby the CV system can be provided with automatically labeled plan data, which is used to expand the training dataset for the initial information extraction process from architectural drawings. This addresses one of

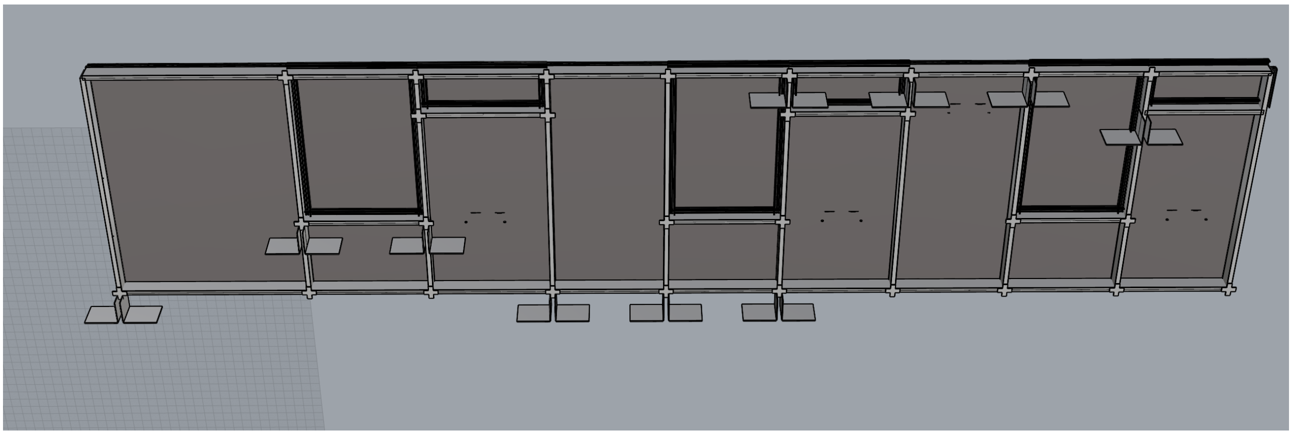

*Figure 3: Facade segment created by parametric model in Grasshopper controlled with a JSON file*

the major challenges faced when implementing AI systems in the AEC sector: the limited availability of reliable, domain-specific training data. Prior work showed how such synthetic data can be derived from existing or parametrically controlled BIM Models and fed into an AI system (Hoeng et al., 2025). This data ensures that the CV system becomes more robust and accurate in extracting semantic and geometric information from the input data. Additionally human-verified optimal model parameters, can be used to enhance the model optimization process by providing additional feedback to the RL system. These human-adjusted parameters serve as high-quality benchmarks, guiding the RL system in refining its decision-making process and aligning it more closely with human expectations. Similar to RL with human feedback, this approach leverages the expertise of human verification to improve the system's performance, ensuring that subsequent iterations of the workflow yield more accurate and practical results. This closed-loop system enables a self-improving process, meaning that the workflow itself becomes more reliable over time. By continuously generating and integrating synthetic training data, the system is meant to reduce the need for human adjustments and manual efforts, ultimately streamlining the entire workflow.

## Validation: Data Interaction and Test of RL-Based System

The individual subprocesses have been successfully implemented at a prototypical level. The CV component combined a trained YOLOv8 model for object detection with Meta's Segment Anything Model (SAM) for boundary extraction, both integrated into a graphical interface. The setup supported model switching between diffrent YOLO models for flexibility in future development. Bounding boxes from YOLOv8 were passed to SAM, and small fragments from complex profiles were removed through simple post-processing. The object detection model was initially trained on a small manually labeled dataset and later extended with synthetically generated drawings from the parametric model. For validating the human–LLM interaction, a pre-prompted and few-shot-tuned GPT model was used, based on OpenAI's Custom GPTs. The customization focused on enforcing a predefined JSON schema, enabling structured control over parametric modeling. This can be achieved by using the structured output formatting feature provided by OpenAI's Application Programming Interface. To fully harness the potential of such automated workflows, seamless interaction and efficient data flow are essential. This paper therefore focuses on testing the interaction between an AI system utilizing RL for optimizing target parameters and modeling software. The control file is automatically processed within Rhino Grasshopper, with evaluation triggered only upon the generation of new input data. The evaluation is then exported in the form of a structured machine-readable report file, which can be used by an AI agent to calculate an immediate reward for changes made to the model. Figure 4 illustrates the interaction between the AI agent and Rhino Grasshopper via structured JSON files. The RL system is programmed to continuously monitor changes in the output file, triggering the next step in its sequence whenever a new version is detected. The same principle applies to the Rhino Grasshopper script, which responds dynamically whenever the RL-modified JSON control file is updated. This setup ensures a seamless yet independent interaction between the RL system and Rhino Grasshopper, enabling parallel improvements to both the RL system and the Rhino Grasshopper script. Several evaluation checks are conducted to assess the facade design. These include: Thermal Analysis (calculation of the U-value of the facade system), Structural Integrity Checks (performed using the Rhino Grasshopper plugin Karamba to ensure load-bearing stability) and Geometric Validation (ensuring all components remain within predefined bounding constraints). All evaluation results are exported into a structured report in JSON format. This JSON report file serves as input for calculating rewards in the RL system. A combination of binary rewards (pass/fail for rule-based checks) and incremental rewards (continuous scoring for numerical evaluations) is used to guide the RL optimization process. An exemplary machine-readable report that serves as the foundation for calculating the agent's rewards

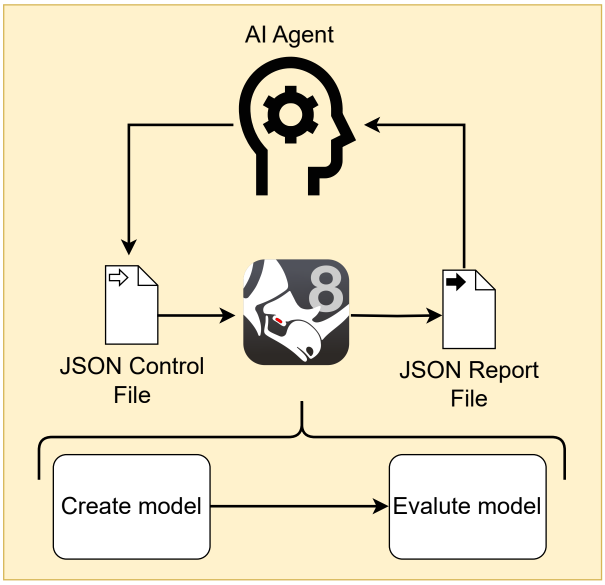


*Figure 4: AI interaction with model through structured files*

is shown below:

```
{
 "model_id": "Model_Version_6.2",
 "date_time": "2025-01-14 17:14:39",
 "checks": {
  "Bounding_box_check_passed": true,
   "bearing_capacity_value": 1,
   "threshold": 0.9
  },
  "Thermal_analysis": {
   "U_value": 0.25,
   "threshold": 0.4
  }
 }
}
```

To test RL integration within this system, a simplified learning scenario was created using Q-learning, a widely applied RL technique. The test focused on optimizing the placement of base facade elements (transoms and mullions) on a grid while adhering to predefined rules: each grid line had to be fully occupied by a fitting element, with only one element per line. Correct placements were rewarded, while incorrect ones resulted in penalties. The RL system could choose to place an element on a specific grid line, with each grid line representing a distinct action. Figure 5 shows such a facade grid created by the RL agent. A training episode continued until the RL agent selected a designated stop action, signaling that placement is considered complete. Correct stopping was highly rewarded to reinforce optimal behavior, while early or late stopping resulted in significant negative penalties. The system is thereby intended to learn autonomously when the placement process should end, ensuring efficiency. To prevent infinite loops, a maximum number of actions was set per episode as a fallback. After each training episode the grid was reset respectively emptied so on the RL agent could start placing elements on an empty grid with the resulting corresponding rewards. In this way a training environment was created in which the RL agent can perform its chosen actions to learn a policy that aims to maximize its overall reward. Due to the large action space involved, which can pose challenges for RL systems, action masking was implemented to improve learning efficiency by eliminating irrelevant choices and reducing unnecessary computations. As it is typical for this kind of RL-based system, hyperparameters play a crucial role in system behavior (Eimer et al., 2023). Figure 6 illustrates the varying behavior of the RL system during training as a result of hyperparameter modifications. In this case, different exploration rates (Epsilon) and minimum exploration rates (Epsilon min) were applied, while the decay rate remained the same, leading to divergent learning outcomes. In RL, the exploration rate determines how often the agent takes random actions instead of following its learned strategy. A higher initial exploration rate means the agent explores more at the beginning, testing different placements before settling on an optimal approach. However, excessive exploration can slow down learning, as the agent may continue making random decisions instead of refining a strategy. The minimum exploration rate sets a lower limit on how much exploration remains after many training steps. If the minimum exploration rate is high, the agent will always retain some randomness in its choices, which can help avoid getting stuck in suboptimal solutions but can prevent full convergence to the best strategy. A lower minimum exploration rate allows the agent to rely more on learned knowledge toward the end of training, leading to more stable decisions. A higher initial exploration rate allowed broader testing of different placements but slowed convergence since random actions persisted longer. In contrast, a lower initial exploration rate or lower minimum exploration rate made the agent settle into a pattern more quickly. While this can speed up training, it also risks the agent failing to find the best solution if it has not explored enough possibilities. Even within 200 training

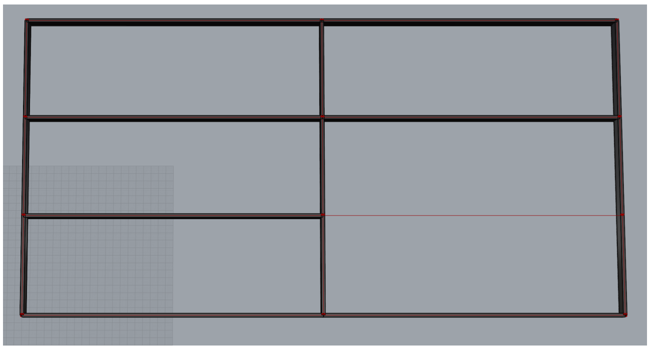

*Figure 5: Simplified mullion-transom grid filled by RL agent with a missing transom*

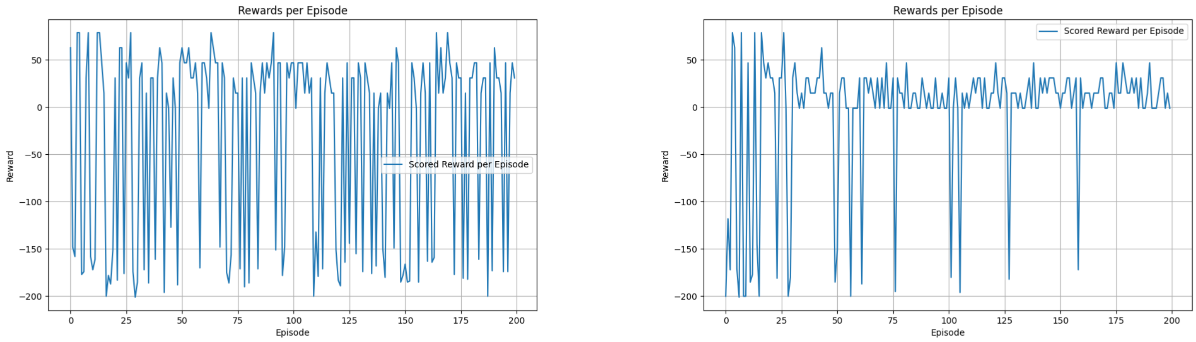


*Figure 6: Different RL outcomes due to varying hyperparameters*
*Left: Learning diverges due to a minimal exploration rate of 0.3 in each episode*
*Right: Learning converges to a stable yet suboptimal reward due to a smaller minimal exploration rate of 0.1*

episodes, balancing exploration and exploitation is crucial. Too much randomness delays optimization, while too little may lead the agent to a premature, suboptimal solution. Another notable observation is the RL system's tendency to select an early stop action when the penalty for incorrect placements is set too high. In such cases, the system learns that avoiding placement altogether is preferable to risking negative rewards, ultimately discouraging exploration and proper optimization. This behavior highlights the importance of carefully balancing reward functions and exploration settings to ensure that the agent remains incentivized to search for correct placements rather than prematurely terminating the episode.

## Discussion of Validation Results

The provided results demonstrate that the RL system can successfully interact with the parametric model, enabling a functional and automated optimization loop while showcasing a seamless yet simple integration of RL techniques into standard architectural design tools. This foundational step highlights the feasibility of embedding AI-driven workflows into conventional planning processes, setting the stage for future advancements in automated design optimization. By employing straightforward data interactions through structured JSON files, complex RL methodologies can be effectively incorporated into architectural design processes, enabling more adaptive and intelligent workflows without disrupting established industry practices. While the proposed framework demonstrates potential for enhancing facade planning efficiency, limitations exist in terms of real-world practicality due to the task's high complexity, especially when using powerful yet fragile technology like RL. Therefore, the parametric facade model was simplified to achieve a somewhat stable performance. The initial CV-based approach was limited due to a lack of labeled training data. To tackle this, the final 2D drawings derived from the 3D model can be used as auto-labeled training data to improve the CV-based information extraction. Meaning that the use of the AI-based workflow will improve the overall quality of the workflow. While certain standards aim to ensure consistent quality and layout in input drawings, the actual quality can vary significantly, ranging from rough hand sketches to detailed plans created with modeling software using generic components. This variability poses a challenge for achieving consistently reliable analysis results and hinders full automation without human validation. Future work will explore additional potential benefits of the proposed closed-loop approach such as development of a custom agent controlling the whole process, as well as possibilities for scalability and adaptability in practical applications. Further alternative optimization technologies like Genetic Programming could be considered as an alternative to RL. Xu et al. (2024) compared these two approaches in specific settings and, depending on the boundary conditions, one or the other resulted in more reliable solutions. Additionally, the importance of automation and digitally supported plan analysis has become evident in this research. While researchers have proposed solutions using continuous model-based data workflows that avoid traditional 2D plan sheets, real-world workflows in the AEC sector still heavily rely on such 2D-based data. To bridge this gap, CV-based plan analysis can serve as a viable intermediary solution until fully model-based or comparable continuous data workflows become standard in the industry. In the depicted workflow, there is an additional manual step of transforming the bill of quantities. With proper prompting and fine-tuning, it is conceivable that an LLM could extract and transfer the needed information into a structured format for subsequent processes. One noteworthy insight regarding the customized LLM used for transforming the calculation input data was its ability to adapt to and self-correct minor issues such as typos or formatting inconsistencies in the input data. This demonstrated the potential of LLM-based data handling compared to conventional data processing, which relies on strictly structured input.

## Summary and Conclusions

This study demonstrates how integrating multiple AI technologies within a closed-loop data workflow can enhance facade planning by reducing manual effort and improving design quality. In particular, the proposed method leverages parametric modeling, CV, LLMs, and RL to automate key tasks and enhance decision-making. An important feature of this workflow is its human-in-the-loop design, which ensures quality control, aims to build trust in AI supported design decisions, and supports real-world adaptability. By incorporating human expertise at critical junctures—such as data validation and model refinement—the workflow maintains alignment with practical constraints while still benefiting from AI-driven automation. Despite this potential, certain limitations exist. RL, while powerful, still requires extensive hyperparameter tuning and requiries refined reward design reward when applied to more complex tasks. Beyond facade planning, the core principles of this AI-driven workflow—such as automated feedback loops, structured data exchange, and model parameter optimization—could be generalized to a variety of tasks in construction. By tailoring the underlying models and analytical checks, the same framework can adapt to different project requirements and constraints. Promising fields for future research include refining how human-in-the-loop validation can guide AI systems toward more robust, trustworthy solutions, enhancing synergy between human expertise and automated processes. Additionally, as continuous, model-based data workflows become more prevalent, fully automated AI agent controlled pipelines that bypass the need for manual interventions could unlock even greater efficiencies. Overall, through a prototypical implementation in facade planning, this work demonstrates the feasibility of a cohesive, AI-enhanced planning methodology, contributing to the advancement of data-driven design processes in the AEC industry.

## Acknowledgments

This work is supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag as part of the KIFa-Project. We thank our collaborating partners from No Doubts GmbH respective Hoelscher GmbH, Dr. Michael Drass and Prof. Dr. Michael Kraus.

## References

Abioye, S. O., Oyedele, L. O., Akanbi, L., Ajayi, A., Davila Delgado, J. M., Bilal, M., Akinade, O. O., and Ahmed, A. (2021). Artificial intelligence in the construction industry: A review of present status, opportunities and future challenges. Journal of Building Engineering, 44:103299.

Ahmad, A., Bande, L., Ahmed, W., Young, K., and Jha, M. (2024). AI application in architecture in UAE: Application of an advanced optimized shading structure as a retrofit strategy of a midrise residential building façade in downtown Abu Dhabi. Energy and Buildings, 325:114995.

Du, C., Nousias, S., and Borrmann, A. (2024). Towards a copilot in BIM authoring tool using large language model based agent for intelligent human-machine interaction. In Proc. of the 31st Int. Workshop on Intelligent Computing in Engineering. European Group for Intelligent Computing in Engineering.

Eimer, T., Lindauer, M., and Raileanu, R. (2023). Hyperparameters in Reinforcement Learning and How To Tune Them. In Proceedings of the 40th International Conference on Machine Learning, pages 9104–9149. PMLR.

Elmaraghy, A., Montali, J., Restelli, M., Causone, F., and Ruttico, P. (2023). Towards an AI-Based Framework for Autonomous Design and Construction: Learning from Reinforcement Learning Success in RTS Games. In Computer-Aided Architectural Design. INTERCONNECTIONS: Co-computing Beyond Boundaries, pages 376–392. Springer Nature Switzerland.

Hoeng, S. K., Eder, F., Schmailzl, M., and Obergrießer, M. (2025). Exploring the Potential of BIM Models for Deriving Synthetic Training Data for Machine Learning Applications. In Advances in Information Technology in Civil and Building Engineering: Proceedings of ICCCBE 2024 - Volume 2: Simulation and Automation, pages 54–63. Springer Nature Switzerland.

Mahdavinejad, M., Bazazzadeh, H., Mehrvarz, F., Berardi, U., Nasr, T., Pourbagher, S., and Hoseinzadeh, S. (2024). The impact of facade geometry on visual comfort and energy consumption in an office building in different climates. Energy Reports, 11:1–17.

Mohammadpour, A., Karan, E., and Asadi, S. (2019). Artificial intelligence techniques to support design and construction. In Proceedings of the 36th International Symposium on Automation and Robotics in Construction (ISARC), pages 1282–1289. IAARC.

Shen, Y. and Pan, Y. (2023). BIM-supported automatic energy performance analysis for green building design using explainable machine learning and multi-objective optimization. Applied Energy, 333:120575.

Stoitchkov, D., Breier, P., Slepicka, M., Genc, C., Harmsen, F., Köhler, T., Vilgertshofer, S., and Borrmann, A. (2019). Automatic detection of plan symbols in railway equipment engineering using a machine learning approach. In Proceedings of the EC3 Conference 2019, pages 92–99. European Council on Computing in Construction.

Xu, M., Mei, Y., Zhang, F., and Zhang, M. (2024). Genetic Programming and Reinforcement Learning on Learning Heuristics for Dynamic Scheduling: A Preliminary Comparison. IEEE Computational Intelligence Magazine, 19(2):18–33.

Xu, S., Wang, J., Shou, W., Ngo, T., Sadick, A.-M., and Wang, X. (2021). Computer Vision Techniques in Construction: A Critical Review. Archives of Computational Methods in Engineering, 28(5):3383–3397.