EG-ICE 2025
GLASGOW

Towards Automated BIM Conflict Resolution Using Reinforcement Learning

YUYE JIANG[1,2], CHANGYU DU[1,2], JIABIN WU[1,2], STAVROS NOUSIAS[1,2], ANDRÉ BORRMANN[1,2]

[1]Chair of Computing in Civil and Building Engineering, Technical University of Munich, Germany
[2]TUM Georg Nemetschek Institute, Munich, Germany

ABSTRACT:
The established practice in construction planning is based on design activities of the different disciplines being conducted in parallel, leading to potential conflicts that need to be resolved in coordination sessions. In BIM projects, this principle is termed federated modeling approach when referring to the process of combining multiple individual discipline-specific models into a single, coordinated model for reference, clash detection, and decision-making. Although BIM coordination tools can detect conflicts automatically, resolving them remains a time-consuming manual process requiring iterative design coordination. To address this challenge, this paper proposes a Reinforcement Learning (RL)-based method for automated geometric conflict resolution. We implement a Proximal Policy Optimization (PPO) algorithm in a BIM environment, training the RL agent to resolve conflicts using real-time feedback from a rule-based model checker. The method's feasibility is evaluated across scenarios of varying complexity. The results demonstrate that the agent learns effective conflict resolution strategies, offering a valuable step beyond model checking towards automatic conflict resolution. The code is available at https://github.com/YuyeJ48/Towards-Automated-BIM-Conflict-Resolution-Using-Reinforcement-Learning.

KEYWORDS:
Building Information Modeling, Design Conflict Resolution, Model Checking, Reinforcement Learning, Proximal Policy Optimization

1. INTRODUCTION

Building Information Modeling (BIM) offers a holistic approach to design, construction, and facility management, in which a digital representation of the building product and process is used to facilitate the exchange and interoperability of information (Borrmann *et al.*, 2018). The BIM-based building design process relies on collaboration across different disciplines, including architectural engineering, structural engineering, and mechanical, electrical, plumbing (MEP) engineering. Designers from different disciplines create their own model and conduct their design process independently, often using different BIM authoring software. The BIM coordinator of each discipline communicates and exchanges information at frequent intervals and merges sub-models into a single federated model. Design components of different disciplines are particularly prone to conflict during this process given the high complexity of building models (Chen and Hou, 2014). These conflicts are recognized as a critical cause of deficiencies and poor collaboration performance in building projects (Charehzehi *et al.*, 2017). While conflicts can be identified by model-checking software, they are typically resolved manually, which is a cumbersome process. An automated conflict resolution is hence desirable; however, it is currently underexplored.

A number of previous studies investigated utilizing various Artificial Intelligence (AI) technologies to automate conflict resolution. For instance, Supervised Learning (SL) techniques have been employed to collect existing clash-resolution data and experts' opinions to train a model to resolve clashes automatically (Hsu *et al*., 2020). However, these methods learned from a set of provided labeled examples, which presents a major disadvantage considering the lack of sufficient data within the Architectural, Engineering, and Construction (AEC) sector. Compared to SL approaches, Reinforcement Learning (RL) is advantageous when examples of desired behavior are sparse or unavailable, but it is possible to evaluate examples of behavior according to some performance criterion (Si *et al*., 2009).

In response to this challenge, our research consequently explores the application of using RL techniques to automate the BIM geometric conflict resolution process. We propose a framework for training an RL agent by employing the Proximal Policy Optimization (PPO) (Schulman *et al*., 2017) algorithm in a BIM environment integrated with a rule-based model checker. The RL model is trained in three use cases, each representing a typical geometric conflict within or between disciplines. The experiments are conducted in a single-family house BIM model to investigate the proposed framework's feasibility across increasingly complex scenarios.

2. BACKGROUND AND RELATED WORKS

2.1 Preliminaries

The Industry Foundation Classes (IFC) serves as a vendor-neutral and industry-specific data model schema for multidisciplinary BIM workflow by facilitating information exchange throughout the project lifecycle (Kubicki *et al*., 2019). It allows the representation of geometrical and semantic structures of a building model using an object-oriented approach (Borrmann *et al*., 2018). Each object type is part of a class hierarchy that defines its specialization and generalization relationships.

RL is the problem faced by an intelligent agent that learns behavior through trial-and-error interactions with a dynamic environment (Kaelbling, Littman, and Moore, 1996). In BIM-based conflict resolution workflows, computing the optimal design directly is often infeasible. However, alternative designs can be evaluated and scored based on domain knowledge or specialized software, aligning with RL's ability to learn optimal strategy through iterative feedback. Even more importantly, unlike SL approach, RL does not require pre-existing training data.

The RL agent explores its environment autonomously, observes the current state (s_t) of the environment at a given time t and takes an action (a_t) that is determined by the policy (π). The policy is a mapping that represents the probability of taking a specific action given the state (s_t), denoted as $\pi(a|s_t)$. As a consequence of the action, the state of the environment transitions from s_t to s_{t+1}. The agent also receives a numerical feedback signal, namely the reward (r_t) based on the success or failure of the action. The self-collected knowledge is applied to adjust its policy, thereby enhancing its accuracy in prediction and interaction with its environment. This one iteration of the agent-environment interaction is defined as a step and the sequence of steps that starts with the initial state and ends with the end state is defined as an episode. During the training phase, the agent refines the policy and learns how to respond to states with appropriate actions that maximize the total reward.

The PPO algorithm has become one of the most widely applied algorithms in RL. PPO is a policy gradient-based algorithm designed to offer a more stable and efficient approach to policy optimization. The central innovation of the PPO algorithm is its approach to making the learning process more stable by constraining the extent of policy updates.

2.2 Automated conflict resolution

Several noteworthy studies have been conducted to develop automated solutions for BIM-based clash resolution. Hsu *et al.* (2020) proposed an AI solution incorporating knowledge-based ML and heuristic optimizing techniques was developed to address BIM design clashes. In the paper, five experienced constructors completed questionnaires, and their responses were collected to train a neural network to identify underlying knowledge patterns. These patterns subsequently served as the basis for optimization. Liu *et al*. (2024) considered the clash resolution as a multi-objective optimization problem and proposed a genetic algorithm approach to balance the optimization of multiple objectives. The Design Healing framework was proposed to automatically address design issues identified in code compliance checking (Wu, Nousias and Borrmann, 2025). By integrating the design information with compliance-checking outputs, it employs a graph-based topological algorithm and sensitivity analysis to identify non-compliant components and generate similar code-compliant design alternatives. Du *et* *al*. (2024) proposed a Large Language Model (LLM)-based agentic framework for generating BIM models, in which two agents write imperative code to address issues identified in the generated BIM model by interpreting the BIM Collaboration Format (BCF) files exported from a rule-based model checker.

Harode *et al*. (2021; 2022) investigated the general application of SL for clash resolution, identifying its limitations and introducing a combined SL and RL methodology, in which the authors hypothesize that RL will help mitigate the dependency of SL on the dataset, while SL acts as a pre-training to the RL model and reduce the number of training steps. The follow-up study explored the common strategy by adopting a neural network that could predict possible clash resolution options (Harode, Thabet and Gao, 2024).

Most of these approaches involved SL and optimization rely on existing examples or manually labeled data that require input from experienced experts, which poses challenges in the data-sparse AEC domain. This also restricts their applicability in different scenarios, as the inputs are typically constrained to a single discipline or type. Harode *et al*. (2021) have pointed out the limitations of SL and put forth the concept of integrating SL and RL, developing a framework to facilitate the integration. Nevertheless, in subsequent and more in-depth research, they devoted a significant portion of their efforts to SL, while the research on the RL aspect has remained conceptual and relatively stagnant.

2.2 Application of RL for BIM-Based Design

The rapid advancement of RL has demonstrated its potential for solving complex problems across related domains. Sharbaf *et al*. (2022) presented an RL approach to automatically resolve model merging conflicts based on quality characteristics, as introduced by language modeling engineers as preferences. Yang *et al*. (2023) proposed a Deep Reinforcement Learning (DRL) method for generating three-dimensional pipeline layouts.

The task of rebar design in BIM has been studied extensively using RL. Liu *et al*. (2019) presented a framework employing a targeted Multi-Agent Reinforcement Learning (MARL) system for the automated reinforcement concrete (RC) joints design in BIM, where each rebar is regarded as an intelligent RL agent, allowing the rebar design problem to be formulated as a three-dimensional path-planning problem. This work was further extended, and a typical RL algorithm Q-learning was implemented for a more realistic real-world design (Liu *et al*., 2020). The same framework was also extended to automatically generate clash-free rebar designs in prefabricated concrete wall panels, integrating a Generative Adversarial Network (GAN) to learn from designers' experiences with existing design drawings and generate 2D preliminary rebar designs (Liu *et al*., 2023). Notably, the utilization of RL has achieved remarkable outcomes in related fields, implying the potential for the practical application of RL in the context of automated BIM conflict resolution.

2.3 Research gaps

In summary, we identify the existing research gaps in literature as follows:

- Limited application of RL: The use of RL for general BIM conflict resolution has been scarcely studied. Existing research has predominantly focused on automatic design within BIM, particularly in RC design.
- High data demand for SL: Numerous studies have proposed a range of SL algorithms to resolve conflicts, which require a large quantity of training data to achieve satisfactory results (Sutton and Barto, 1998). However, the dearth of adequate datasets presents a significant barrier for implementing SL in this field. While the input from several experts may be sufficient for specific use cases, this approach is evidently limited by the number and expertise of the experts involved and is difficult to generalize to other scenarios.
- Oversimplification of BIM environments: Due to the complexity of real BIM model, most research studies employed significant simplifications, extracting only the essential information to create a simulated 3D non-BIM model and environment for RL training. However, a BIM model is an integrated whole, with complex interrelations among its objects, making simplifications less than ideal.

3. METHODOLOGY

The proposed methodology integrates the federated BIM model (merging architectural, structural, and MEP disciplines) and the rule-based model checker into an RL environment, enabling training within a real BIM context to automatically resolve geometric conflicts.

Fig. 1 illustrates the overall pipeline of our approach. The process begins with a BIM model containing multiple geometric conflicts, imported into the model checker as a vendor-neutral IFC file. The model checker contains a wide range of comprehensive checking rules and is integrated into the established RL environment where the agent operates. The RL training loop proceeds as follows: The model checker first checks the model and outputs the results. Employing the PPO algorithm, the agent receives rewards or penalties based on the checking results, updates its policy accordingly, and performs appropriate actions to reposition the conflicted building components within the IFC. The updated IFC model is re-checked by the model checker, and the loop continues until a terminated state is reached, so that all conflicts are resolved. Finally, the conflict-free BIM model is exported as the resolution. The RL agent learns the optimal conflict resolution strategy through this training progress, with the objective of minimizing both the number and the severity of conflicts reported by the model checker. The trained RL model is saved for further evaluation and testing. In the following subsections, we describe several key modules of the proposed framework in detail.

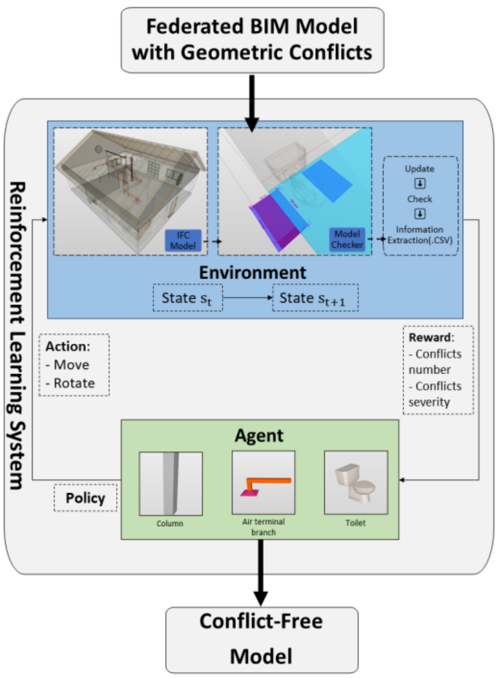


Figure 1: The proposed framework of the RL system based on IFC model and a rule-based model checker.

3.1 Observation space

In contrast to other studies that construct an abstracted and highly simplified building model, our research integrates a complete IFC model into the RL environment. In real-world design, relocating a single element may affect numerous adjacent components. By employing the whole IFC model, our approach incorporates a degree of the intricacies and interdependencies intrinsic to actual building designs.

The IFC component properties and checking feedback from the model checker are encoded as RL observations, i.e., the information the agent receives from the environment at each time step to guide its decision-making. The parameters of the observation space design are summarized in Table 1.

Table 1: Summary of the observation space

Parameters	Definition
Number of conflicts	The total number of conflicts that we aim to resolve in the BIM model.
Severity of conflicts	The accumulative severity indicator for conflicts that we aim to resolve.
Number of created conflicts	The total number of conflicts that we do not intend to cause but arise during the iteration due to inappropriate action.
Severity of created conflicts	The accumulative severity indicator for created conflicts.
Element 1 type	The IFC class of the element.
Element 2 type	
Element 1 rotation	The indicator for the current direction of the element.
Element 2 rotation	
Element 1 vertices	The calculated world-coordinates of eight vertices of the bounding box of the element.
Element 2 vertices	

A conflict in the BIM model may involve one or more elements. When a conflict involves only a single element, it typically results from the elements' properties not meeting relevant requirements, which is outside the scope of this research. For most other cases, geometric conflicts typically involve two elements. While some conflicts may include multiple elements simultaneously, these can usually be broken down into a series of pairwise geometric conflicts. Therefore, this study focuses on analyzing geometric conflicts at the element-pair level.

3.2 Action space

To effectively resolve geometric conflicts in BIM models, actions must be designed to enable the agent to directly and practically manipulate model elements, thereby addressing the identified conflicts. Based on the literature research and practical experience, the primary actions involved in resolving geometric conflicts include moving and rotating. In practice, there might be other feasible actions. However, due to implementation constraints and the need for adaptability across different use cases, the available actions in this method are focused on movement and rotation. In addition, the voids or niches that may be created by the actions are out of the scope of this study.

Eight available actions are designed in the RL environment to construct a discrete action space, including forward and reverse movement (10 mm) in the x, y, z axis, and rotate 90 degrees clockwise or anticlockwise. The value of 10 mm was determined by the fact that it is the typical tolerance for clash detection. Each possible action is indexed and encoded as part of the discrete action space.

It is worth noting that the RL terminology *action* here only includes the movement or rotation of the main conflicted elements. Because the BIM model integrates both geometric and semantic information, relocating certain elements, particularly those in the MEP system, often requires corresponding adjustments to connected neighboring components. For example, the movement of an air terminal can result in length changes of ducts and movement of duct fittings. These consequential changes are also considered and designed in the RL environment to guarantee the integrity of the BIM model, but they do not belong to the agent action module.

3.3 Rewards and checking rules

The reward system functions as the primary feedback mechanism, indicating to the agent whether its actions are leading to improvements or deteriorations in the state of the environment. To better guide the agent in learning to make decisions that lead to better states, the reward is designed with three primary considerations based on the state of the current BIM model, mainly related to the output of the model checker, as summarized in Table 2.

Table 2: Summary of the reward module

Key parameters	Reward or penalty
The change in the number of conflicts	± 1
The change of created conflicts	± 1
The change in conflicts' severity	± 0.2

To conduct comprehensive checking of the model and to better reflect the state of the model within the reward system, the rulesets in the model checker are defined to align with the reward module. The configuration of the rulesets takes three aspects into account:

- The primary rule to check specific types of conflicts that the agent aims to resolve.
- The auxiliary rules to detect the specific conflicts that may arise during the relocation of the element according to its type and properties.
- The general BIM validation rules ensure that the complete model is comprehensively checked for semantic, structural, and topological rationality.

4. IMPLEMENTATION DETAILS

The developed RL system is illustrated in Figure 2. The RL environment is built on the OpenAI Gym framework (Brockman *et al.*, 2016), using the PPO implementation from Stable Baselines3 (Raffin *et al.*, 2021). IfcOpenShell [1] is used to extract geometric and semantic information and to perform movement and rotation of building components in the IFC model.

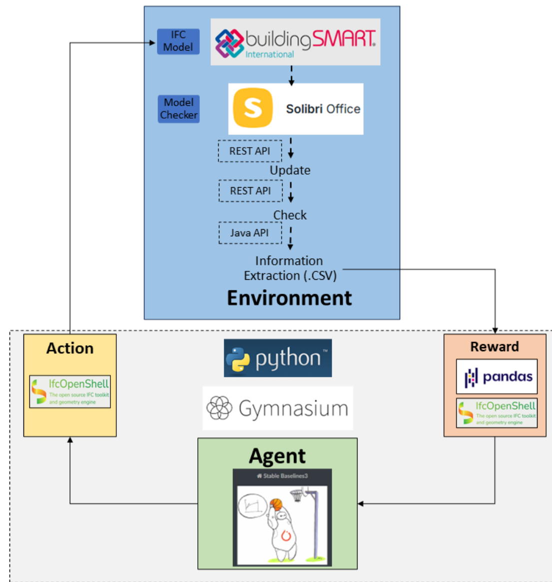


Figure 2: Implementation of the RL system

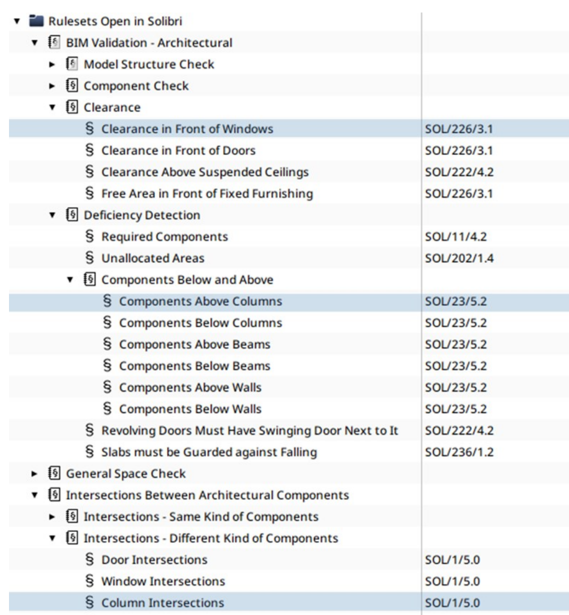


Figure 3: An example of the selected rules in Solibri for column-window conflict following the principles defined in Section 3.3 (primary and auxiliary rules highlighted)

[1] https://ifcopenshell.org/

We use Solibri Office[2] as the rule-based model checker. Compared to design check functionalities in a self-formulated environment, Solibri offers a high degree of comprehensiveness and adaptability in incorporating domain knowledge into its rule-based checking system. This helps identify unintended conflicts from element relocation during training, preserving BIM model integrity. The completed and accurate rule selections and setup can maximize the benefits of applying the model checker and ensure the optimal functioning of the RL reward system. Conversely, the rules should not be excessively repetitive and should be tailored to the specific objective, in alignment with the principles in Section 3.3. Figure 3 provides an example of selected rulesets in Solibri for the column-window conflict in Section 5.2. We utilized Solibri's REST and JAVA APIs to update and check the modified IFC model continuously, as well as extract checking results for subsequent interpretation.

5. EXPERIMENTS AND EVALUATION

Three distinct experiments were conducted within an IFC model of a representative single-family house (Figure 4). Each experiment represents a characteristic conflict scenario with increasing complexity within or across different disciplines (Table 3). The training outcomes were collected and analyzed separately.

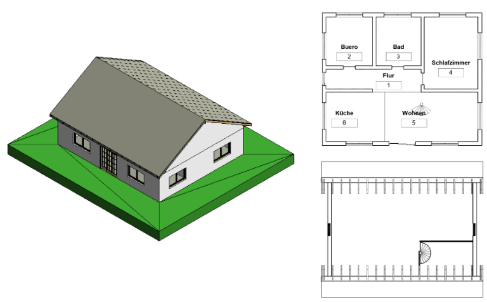
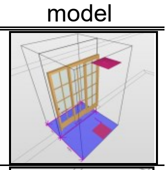
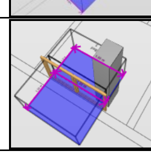
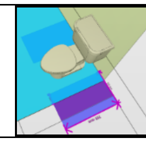


Figure 4: The IFC model used in experiments

Table 3: Three conducted experiments

Conflict	Description	Type	Illustration in model
Air Terminal vs. Door	MEP terminals intrude into spaces for architectural doors.	Architectural vs. MEP	
Column-Window Conflicts	Structural columns are positioned too close to where architectural design specifies windows.	Structural vs. Architectural	
Space for Toilet Seat	Inadequate room layout for toilet seat placement or accessibility.	Architectural Sub-Aspects	

[2] https://www.solibri.com/

5.1 The air terminal-door conflict

A common conflict that arises in the integration of MEP systems with architectural design is the discouraged placement of air terminals intruding into spaces for door swings. In practical BIM design workflows, architectural models are typically provided prior to MEP design, and modifications to architectural elements are generally avoided unless no feasible HVAC adjustments can be made, making repositioning the air terminal a more realistic first approach. As mentioned in Section 3.2, the movement of MEP objects is generally more challenging than that of other disciplines since they are typically situated in specific systems and interconnected. The movement of a single element often necessitates the coordinated movement of numerous other related elements. In this case study, four air terminals with different directions were purposely positioned in front of different doors, creating four specific conflicts for the agent to resolve. The air terminal and the door are considered as conflicting elements 1 and 2, respectively, with only the air terminal subject to be moved during training. The hierarchical relationships of the IFC file enable the retrieval of all air terminal-related elements' GUID. In addition to the air terminal itself, the connected short duct and duct fitting should also be relocated, and the long duct should be shortened or lengthened accordingly. The primary rule for this use case is *the Distance Between Doors and MEP components*. As illustrated in Figure 5, the agent needs around 3000 training steps to learn to resolve all four fixed conflicts and converge on the maximum possible reward. The plotted curve shows the rollout mean episode reward against the training steps, which represents the average total reward obtained over several episodes during each evaluation phase.

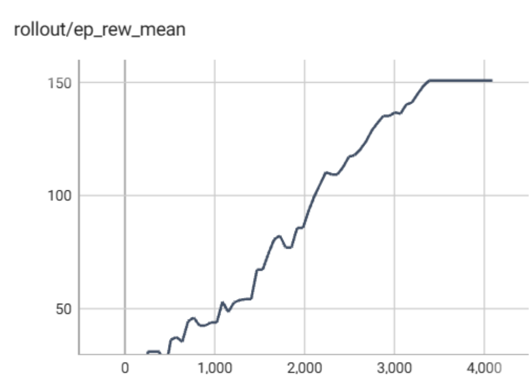


Figure 5: Rollout mean episode reward plotted against steps for air terminal-door case

5.2 The column-window conflict

In architectural and structural design workflows, conflicts may arise when structural elements such as a column is placed too close to key architectural features like a window which is crucial for natural lighting and ventilation. In practice, the resolution of this type of conflict requires collaboration between architects and structural engineers, taking into account a number of factors, such as whether the column is load-bearing. For training, the conflict resolution process was simplified to entail the relocation of the column. In this use case, the reset of the training environment varies for each training episode. The column is randomly assigned to one of the nine regions distributed in front of each window in the house, with its exact position within the chosen region determined by a continuous spatial distribution, as illustrated in Figure 6. The environment is guaranteed to be different for each initialization, but at the same time, a desired conflict is present.

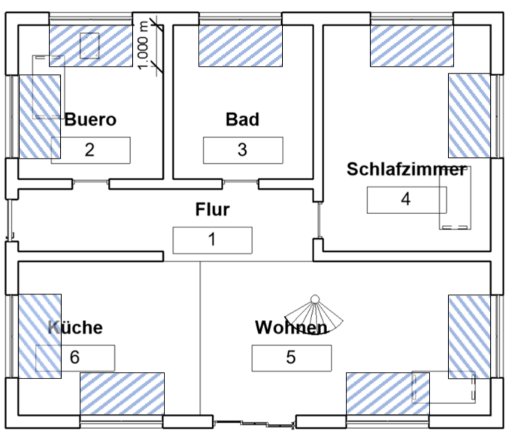


Figure 6: The reset placement area of the column

In order to better emulate the real-world scenario, additional architectural furniture (e.g., tables) was placed in the IFC model This augmentation increases the complexity of the environment and requires the agent to avoid conflicts with other architectural elements while resolving the main conflict. Three distinct types of rules are selected and adjusted for this particular instance accordingly. The *Clearness in Front of Windows* rule is the primary means of detecting the focused conflict the agent aims to resolve. Subsequently, the general rule *Column Intersections* is employed to detect whether the relocated column creates new conflicts with other elements in the entire building. The *Components Above Columns* rule ensures that when the column is relocated outside of the building, the model checker can indicate that a conflict exists, even when the clearness of the window is guaranteed and there is no collision between the column and other components.

The algorithm initiates its operation without knowing the environment. In the early stage of training, the algorithm randomly attempts the available operations and gradually improves. After about 10000 training steps, the reward received by the agent stabilized, and the loss curve also converged, as illustrated in Figure 7.

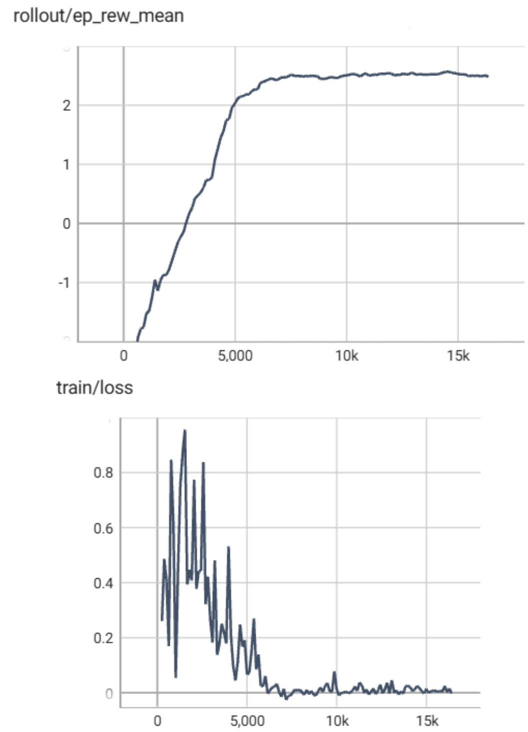


Figure 7: Rollout mean episode reward and training loss plotted against steps for column-window case

5.3 The toilet-wall conflict

A common architectural sub-aspect conflict in bathrooms arises when the placement of a toilet seat in close proximity to the wall, as the minimum lateral distance of 30 cm is necessary to ensure accessibility and comfort. Additionally, the toilet needs to be installed on the rear wall and the misplacement could lead to alignment issues with sanitary pipes of the plumbing system. This issue can be extended to many of the conflicts of insufficient distance between furniture, and inadequate accessibility of spaces, which are very common interdisciplinary building design. Similar to the previous cases, the reset of the training environment for this case varies for each training episode. The toilet seat in the bathroom is randomly assigned, with its exact position within two corner areas determined by a continuous spatial distribution, but always too close to one of the walls. The solution to resolve the conflict is to reposition the toilet so that its free space on both sides is more than 500 mm, and its distance from the rear wall does not exceed 10 mm. Given these considerations, four distinct rules were especially selected from the Solibri rulesets and adapted to align with the training requirements: (1) *Shower and Bathrooms* rule was adjusted to check the free space on both sides of the toilet seat. (2) *Component Distance* rule is created to ascertain the proximity of any wall elements to the toilet seat, ensuring that the toilet is not positioned in the center of the room. (3) *Object Intersections* are used to check if the toilet clashes with other components in the bathroom. (4) *Space Intersections* to ensure the toilet is placed in the bathroom space, not outside.

Figure 8: Rollout mean episode reward and training loss plotted against steps for toilet-wall case

Compared to the previous use cases, the terminated state of this training is more difficult to reach since it requires the toilet seat abutting to a wall rather than being placed anywhere in the room. Due to the increasing complexity, the total number of training steps was set to 32,768. The training process spanned more than five days. The rolling average of the episode mean reward received by the agent is depicted in Figure 8. As the training progresses, the reward exhibits a gradual upward trend, suggesting the agent has acquired a certain level of knowledge to resolve the conflict. However, it has not yet reached convergence. The loss curve exhibits an overall downward trend but has not yet converged, indicating that the model is still undergoing optimization.

6. DISCUSSION AND FUTURE WORKS

As pioneering research into the use of RL for BIM conflict resolution automation, the results of the experiments applying the proposed approach did achieve certain expectations. However, the study still has some limitations.

6.1 Training efficiency

Model-free RL algorithms, including PPO, are relatively sample-inefficient. They require a substantial number of samples, often millions of interactions to achieve some meaningful results, which is the key reason why most of the successes in RL were achieved on games or in simulation only (Stable Baselines3, 2024). The insufficient training step could be the primary factor contributing to the unsuccessful training outcomes in the toilet use case. The integration with Solibri limits the training speed by the speed at which the Solibri software executes. Although the training efficiency has been significantly improved by shutting down and restarting Solibri after several steps to clear the cache, the speed still decreases as the step increases as exemplified in Figure 9.

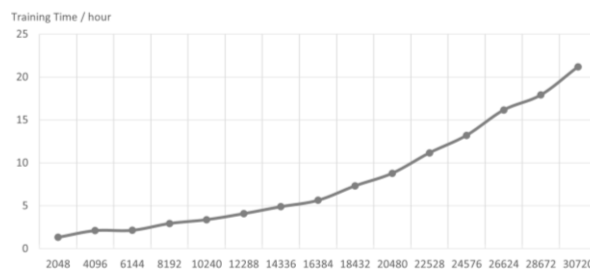


Figure 9: Training time in hour per 2048 time steps over toilet-wall case training process.

Future work will explore enhancing the integration of Solibri to accelerate the training process, forming a foundation for further improvements. For instance, a cloud-based solution potentially offers faster execution compared to the local checking. In addition, the PPO algorithm's support for parallel environments offers the potential for more efficient training and CPU utilization. Since Solibri permits the inclusion of multiple models within a single checking file, it may be feasible to envision a scenario where multiple agents are simultaneously controlled across distinct models.

6.2 Generalizability

A further limitation of the current research lies in its generalizability, as the RL model was trained independently across three scenarios. From a practical perspective, it would be advantageous to develop a unified model capable of resolving all conflicts simultaneously. This is theoretically feasible, as the three models share a common framework. Nevertheless, in current trainings, adjustments were made to movement distances, available actions, the number of IFC types, and the corresponding rules to improve the agent's learning efficiency and enable optimal policy development. Moreover, to solve more types of conflicts, action space should also be extended to include other design actions such as resizing the elements and creating a void for MEP pipes and ducts.

However, training a single model across all use cases with more available actions would significantly increase the number of steps required for conflict resolution, making the training process more time-consuming. Therefore, unifying the separately trained RL would require meticulous calibration of the agent's learning parameters. For example, breaking different rules could be subject to disparate penalties, which would be determined according to the applicable building regulations. Future research could focus on refining the RL parameters to balance generalizability and training efficiency. Additionally, exploring alternative RL algorithms, such as DRL, may provide insights into potentially more suitable methods.

Another limitation is associated with its applicability across different BIM models. During the training process, only a standard one-family house with relatively simple geometry was utilized. Although conflicting components are randomly placed during the initialization of the RL environment and additional elements are added to ensure a certain degree of diversity and complexity in the conflicts, further testing and evaluation are necessary before applying the trained model to larger and more complex systems. The complexity of the IFC model should be incrementally augmented, commencing with the successful column-window conflict resolution of relatively simple environments and subsequently progressing to further training in more complex IFCs.

In addition, the conflicted components' type and dimension are the same in the current experiments, and only the placement and orientation are randomly generated. Implementing generative design techniques for the automated generation of conflict scenarios could further enrich the diversity of training data.

6.3 Hyperparameter tuning

The selection of hyperparameters in RL significantly impacts the rate of convergence, the stability of the learning process, and the overall success of the learning task. However, tuning RL hyperparameters does not have clear and sufficient scientific principles to work with (Li, 2018), and the process of tuning these hyperparameters is notoriously challenging due to the high dimensionality of the hyperparameter space of PPO and the stochastic nature of RL environments. Incorporating automated hyperparameter optimization tools, such as the Bayesian optimization strategy or the Optuna optimization framework, for the systematic tuning of the parameters of the PPO algorithm should be beneficial.

6.4 Enhancing domain knowledge integration

By incorporating a rule-based model checker into the RL environment, fundamental design knowledge from the building domain has been embedded into the checking mechanism. This integration helps ensure that the RL model adheres to clash-free design principles and essential design criteria. However, achieving more comprehensive knowledge integration remains a challenge. Integrating more design logic has the potential to improve training efficiency as it would reduce the agent's exploration of the wrong actions. For instance, spatial alignment grids can be integrated to guide the RL agent, thereby constraining its action to more plausible configurations by following parametric dependencies (Wu *et al.*, 2025). As a concrete example, incorporating the parametric constraint, *a toilet must be adjacent to a wall* could potentially facilitate successful convergence within the same training steps in the third use case discussed. However, applying such parametric dependencies and constraints would require interaction with a BIM modeling tool capable of handling parametric relationships, rather than relying solely on a vendor-neutral IFC model.

While the resolved federated BIM model may be clash-free, real-world design decisions must also consider factors such as cost, material availability, and alignment with design preferences. These aspects, which are currently not captured by the model checker, could be further encoded into the RL reward system to guide the agent toward more practical and holistic design solutions. It is noteworthy that the rules in Solibri are highly extensible and customizable. The *Information Take-off* feature enables detailed extraction of building component data, which could contribute to quantifying the material and cost factors in the RL environment. Future work can focus on incorporating more complex knowledge into the RL framework. Furthermore, leveraging LLMs presents a potential avenue for extracting and integrating semantic knowledge from design regulations into the RL environment.

7. CONCLUSION

This paper presets a preliminary study on the application of a PPO-based RL algorithm for automated BIM geometric conflict resolution by integrating a complete IFC model and a rule-based model checker into a custom RL environment.

Our methodology does not require initial labeled data but rather embeds the domain knowledge into the RL environment to estimate the current model state, thus addressing identified research gaps. To evaluate the feasibility of the proposed framework, the RL agent was trained separately in three different use cases, demonstrating the adaptability of the approach. The experiments have yielded preliminary promising results, showing the potential of utilizing RL for automated BIM conflict resolution. However, further research is required to refine the proposed method.

ACKNOWLEDGMENT

This work is funded by Nemetschek Group, which is gratefully acknowledged. We sincerely appreciate the licensing support provided by Solibri, Inc.

REFERENCES

Borrmann, A., Beetz, J., Koch, C., Liebich, T. and Muhic, S., (2018). Industry Foundation Classes: A Standardized Data Model for the Vendor-Neutral Exchange of Digital Building Models. In: A. Borrmann, M. König, C. Koch and J. Beetz, eds. *Building Information Modeling*. [online] Cham: Springer International Publishing. pp.81–126. https://doi.org/10.1007/978-3-319-92862-3_5.

Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Hrsg.), (2018). Building Information Modeling – Technology Foundations and Industry Practice. *Springer International*

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J. and Zaremba, W., (2016). *OpenAI Gym*. Available at: http://arxiv.org/abs/1606.01540 [Accessed 28 August 2024].

Charehzehi, A., Chai, C., Md Yusof, A., Chong, H.-Y. and Loo, S.C., 2017. Building information modeling in construction conflict management. *International Journal of Engineering Business Management*, 9, p.184797901774625. https://doi.org/10.1177/1847979017746257.

Chen, H.-M. and Hou, C.-C., (2014). Asynchronous online collaboration in BIM generation using hybrid client-server and P2P network. *Automation in Construction*, 45, pp.72–85. https://doi.org/10.1016/j.autcon.2014.05.007.

Du, C., Esser, S., Nousias, S. and Borrmann, A., (2024). *Text2BIM: Generating Building Models Using a Large Language Model-based Multi-Agent Framework*. https://doi.org/10.48550/arXiv.2408.08054.

Hsu, H.-C., Chang, S., Chen, C.-C. and Wu, I.-C., (2020). Knowledge-based system for resolving design clashes in building information models. *Automation in Construction*, 110, p.103001. https://doi.org/10.1016/j.autcon.2019.103001.

Kaelbling, L.P., Littman, M.L. and Moore, A.W., (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 4, pp.237–285. https://doi.org/10.1613/jair.301.

Kubicki, S., Guerriero, A., Schwartz, L., Daher, E. and Idris, B., (2019). Assessment of synchronous interactive devices for BIM project coordination: Prospective ergonomics approach. *Automation in Construction*, 101, pp.160–178. https://doi.org/10.1016/j.autcon.2018.12.009.

Li, Y., (2018). *Deep Reinforcement Learning: An Overview*. Available at: http://arxiv.org/abs/1701.07274 [Accessed 2 September 2024].

Liu, J., Liu, P., Feng, L., Wu, W. and Lan, H., (2019). Automated Clash Resolution of Rebar Design in RC Joints using Multi-Agent Reinforcement Learning and BIM. [online] 36th International Symposium on Automation and Robotics in Construction. Banff, AB, Canada. https://doi.org/10.22260/ISARC2019/0123.

Liu, X., Zhao, J., Yu, Y. and Ji, Y., (2024). BIM-based multi-objective optimization of clash resolution: A NSGA-II approach. *Journal of Building Engineering*, 89, p.109228. https://doi.org/10.1016/j.jobe.2024.109228.

Mohri, M., Rostamizadeh, A. and Talwalkar, A., (2018). *Foundations of machine learning*. Second edition ed. Cambridge, Massachusetts: The MIT Press.

Raffin, A., Hill A., Gleave A., Kanervisto A., Ernestus M., Dormann N., (2021) *Stable-Baselines3: Reliable Reinforcement Learning Implementations*, 22(268), p. 1–8.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., (2017). *Proximal Policy Optimization Algorithms*. Available at: http://arxiv.org/abs/1707.06347 [Accessed 11 August 2024].

Sharbaf, M., Zamani, B. and Sunyé, G., (2022). Automatic resolution of model merging conflicts using quality-based reinforcement learning. *Journal of Computer Languages*, 71, p.101123. https://doi.org/10.1016/j.cola.2022.101123.

Stable Baselines3. (n.d.). Reinforcement Learning Tips and Tricks. [online] Available at: https://stable-baselines3.readthedocs.io/en/master/guide/rl_tips.html [Accessed 1 Apr. 2025].

Sutton, R.S. and Barto, A.G., (1998). *Reinforcement learning: an introduction*. Cambridge, Mass.: MIT Press.

Wu, J., Esser, S., Nousias, S. and Borrmann, A., (2025). Enriching IFC Models with Spatial Design Logic and Parametrics to Improve Design Adaptability – The Case of Alignment Grids. In: A. Francis, E. Miresco and S. Melhado, eds. *Advances in Information Technology in Civil and Building Engineering, Lecture Notes in Civil Engineering*. [online] Cham: Springer Nature Switzerland. pp.91–105. https://doi.org/10.1007/978-3-031-84208-5_8.

Wu, J., Nousias, S. and Borrmann, A., (2025). Design Healing framework for automated code compliance. *Automation in Construction*, 171, p.106004. https://doi.org/10.1016/j.autcon.2025.106004.

Yang, C., Zheng, Z. and Lin, J.-R., (2023). Automatic Design Method of Building Pipeline Layout Based on Deep Reinforcement Learning.