2025 European Conference on Computing in Construction

CIB W78 Conference on IT in Construction

Porto, Portugal
July 14-17, 2025

A PRELIMINARY STUDY ON DESIGN RULE DERIVATION FROM GRAPHICAL REPRESENTATIONS USING MULTIMODAL LARGE LANGUAGE MODELS

Yonghan Kim[1], André Borrmann[1], and Ghang Lee[1,2]
[1]Technical University of Munich, Munich, Germany
[2]Yonsei University, Seoul, South Korea

Abstract

Automated design compliance checking has traditionally focused on interpreting natural language clauses, often neglecting accompanying graphical representations of requirements. These visual elements are crucial for accurately understanding requirements but remain underutilized in automated rule derivation. This study explores the potential of multimodal large language models (MLLMs) to generate machine-readable rules from textual and graphical representations. Focusing on accessibility regulations, 23 clause-graphic pairs were collected, and corresponding ground truth rules were manually generated in Prolog. The MLLM's outputs under three input conditions - text-only, graphic-only, and combined - were evaluated against this ground truth. Results show that the combined input case yields the highest F1 score of 0.96, while the text-only and graphic-only cases yield 0.74 and 0.43, respectively. The study demonstrates the potential of MLLMs to interpret multimodal regulatory inputs for automated rule derivation for design compliance checking.

Introduction

Automated derivation of machine-readable rules has been a key component of design compliance checking (Eastman et al., 2009). Most research in this field has focused on natural language processing (NLP) techniques to extract and interpret regulations or guidelines in textual format (Fuchs and Amor, 2021). However, many design guidelines and requirements involve textual descriptions along with tabular and graphic representations (Fuchs and Amor, 2021). Those visual elements often accompany textual clauses to clarify spatial constraints and assist expert understanding, as shown in Figure 1.

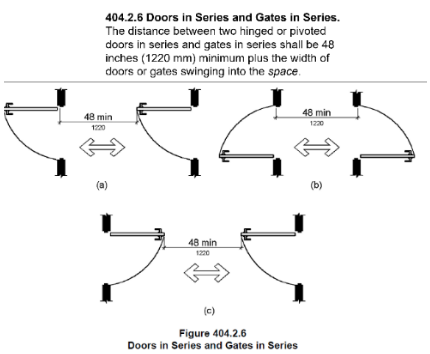


Figure 1: An example of graphical representations with a corresponding clause (U.S. Department of Justice, 2010)

Even recent rule derivation approaches remain primarily text-centric, with limited use of visual elements (Yang and Zhang, 2024; Zhang, 2023). With the advancement of multimodal large language models (MLLMs) that can process both textual and visual data, new opportunities have emerged. While some studies have explored the integration of figures and tables through MLLMs (Zentgraf and König, 2025) and multimodal retrieval-augmented generation (Ying and Sacks, 2024), the direct use of graphical representations for generating machine-readable rules has not been quantitatively evaluated.

To address this gap, this study presents a preliminary study on MLLM-based rule derivation that incorporates graphical representations. A dataset of 23 regulatory clauses and their associated visual elements, including dimensional, annotated drawings, and implementation examples, was collected from accessibility design regulations. Three clause-graphic pairs were used for in-context learning, and the remaining 20 were used for testing. Three different input modalities were compared: text-only, graphic-only, and a combination of both. The MLLM outputs were formalized in Prolog, a logic programming language, and compared against manually constructed ground truth to assess rule derivation performance.

The remainder of this paper is structured as follows: Section 2 reviews related work on LLM-based automated rule derivation and MLLM applications using image input. Section 3 details the research methodology and experimental framework. Section 4 presents the experimental results, followed by Section 5, which provides a detailed discussion of the findings, limitations, and potential improvements. Finally, Section 6 concludes with key insights and future research directions.

Related works

Automated rule derivation is considered as a promising approach to reduce the labor and error-prone nature of manual rule interpretation. The emergence of large language models (LLMs) has recently introduced a new paradigm for automated rule derivation and compliance checking, offering improved scalability and adaptability. However, despite significant advancements, existing rule derivation approaches remain predominantly text-centric, overlooking the role of graphical representations in regulatory standards or design requirements.

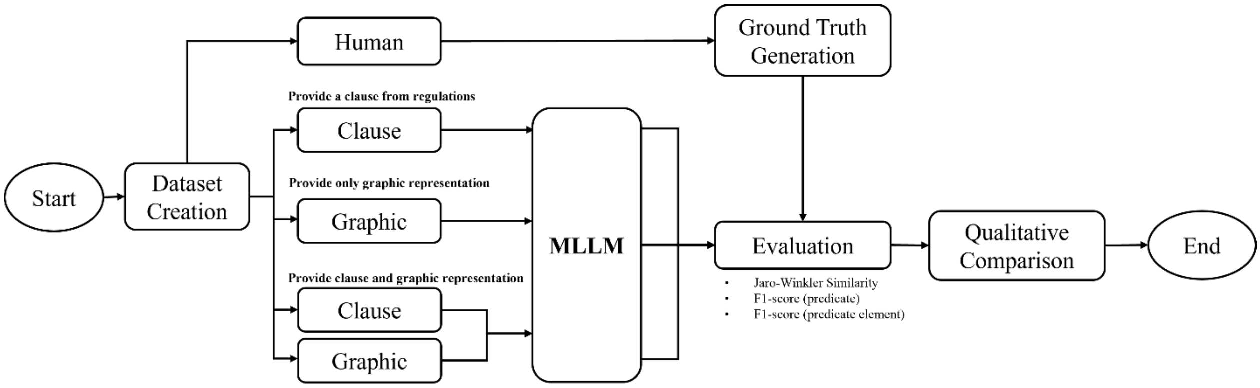


Figure 2: Research Framework

LLM-based automated rule derivation

Recent research has explored LLM-based approaches for extracting, transforming, and applying regulatory rules. Zhang investigated using LLMs to extract and transform textual regulations into Python code, highlighting several limitations, including inefficiencies in processing time compared to NLP-based methods and the need for domain knowledge to refine the generated code for practical use (Zhang, 2023). Zheng et al. proposed a framework leveraging LLMs to map predefined compliance functions, utilizing LLM-extracted information to mitigate the challenge of domain expertise dependency in compliance checking. (Zheng et al., n.d.). Yang et al. proposed a prompt-based framework for transforming building codes into Prolog format (Yang and Zhang, 2024). Fuchs et al. evaluate the performance of LLMs in translating regulation to LegalRuleML format using in-context learning (Fuchs et al., 2024). However, their study focused solely on textual rules, neglecting graphical representations integral to regulatory interpretation.

MLLM-based graphic interpretation

With the advancement of MLLMs, which can process text and images, there is growing interest in applying these models to visual reasoning tasks and logically coherent source code. MLLMs have shown strong image comprehension capabilities in fields such as document processing, visual question answering, and interpretation of engineering drawings (Doris et al., 2024). For generating source code from the image input, Mu et al. utilized visual observation to make MLLMs generate codes that control robotic behavior (Mu et al., 2024). Wu et al. show that MLLMs can generate matplotlib code from scientific visual figures with proper prompt engineering (Wu et al., 2024). Generating compilable user interface code with MLLM from screenshots also proposed offering an alternative solution for design-to-code (Wan et al., 2024).

These studies indicate that MLLMs can extract properties and logical structures from graphical representations to create machine-interpretable representations.

Integration of graphical representations in compliance checking

A recent study has attempted to incorporate graphical representations into information extraction and compliance checking. Ying and Sacks integrated figures and tables by embedding their textual descriptions and retrieving relevant regulations based on user queries (Ying and Sacks, 2024). Zentgraf and König demonstrated the feasibility of using MLLMs to extract structured information from regulatory figures and associated texts (Zentgraf and König, 2025). However, lacked quantitative evaluation, and the direct transformation of graphical representations into a machine-readable rule format remains unexplored.

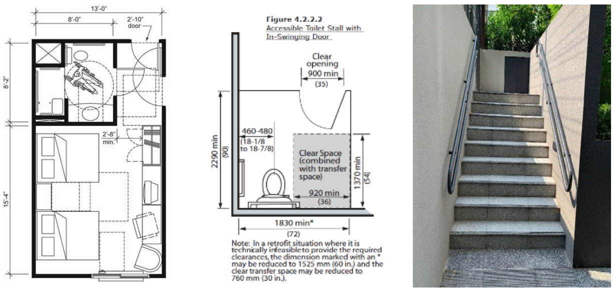
Therefore, this study aims to bridge this gap by investigating the utilization of graphical representations in MLLM-based rule derivation. With the integrated inputs, the authors hypothesize that graphical representations improve the accuracy and completeness of automated rule derivation, similar to how graphical representations assist human experts in interpreting regulations and design guides.

Research method

The research framework consists of dataset creation, ground truth generation, followed by rule creation using MLLM, and evaluation, as illustrated in Figure 2.

Dataset creation

To assess the feasibility of MLLM-based rule derivation, a dataset was created using the universal design guidelines, with a specific focus on accessibility regulations. This domain was chosen due to the significant presence of graphical representations, such as dimensional drawings, annotated drawings, and implementation examples, which are used to clarify requirements. The dataset was compiled from three different guideline documents, resulting in 23 clause-graphical guideline pairs. The clauses are selectively extracted, containing quantifiable regulations without any exemptions. The graphical representations consist of 11 dimensional drawings, 6 annotated drawings, and 6 implementation examples, each serving a specific function in regulatory interpretation. Dimensional drawings provide precise measurements and spatial relationships necessary for compliance verification, such as minimum clearance spaces, turning radius, and door width requirements. Annotated drawings include not only dimensional information but also text descriptions. Implementation examples illustrate real-world applications of regulatory requirements, demonstrating compliant cases to clarify best practices. The examples of each graphic requirement are shown in Figure 3.



(a) Dimensional drawing *(b) Annotated drawing* *(c) Photographic example*

Figure 3: Three cases of graphical representations (U.S. Department of Justice, 2010; City of London, 2007; Federal Ministry for the Environment, Nature Conservation, Building and Nuclear Safety, 2015)

Ground truth generation

For the output format for machine-readable rules, we have chosen the Prolog language, following the approach used in Yang and Zhang's research (Yang and Zhang, 2024). Since the primary goal of this study is to evaluate the rule derivation capability of MLLMs, it was assumed that all necessary compliance checking functions and attribute information for compliance checking were available. Figure 4 illustrates the process of generating ground truth Prolog code.

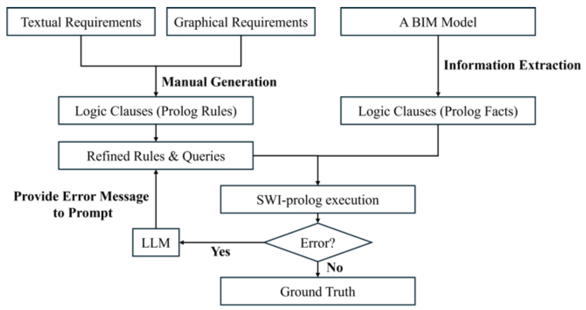


Figure 4: Process of generating ground truth Prolog code from textual and graphic design requirements

Manual Prolog rule and query creation were conducted, considering textual clauses and graphical representations to generate the ground truth. The rules were structured according to Prolog-based compliance reasoning principles to ensure compatibility with logic-based compliance checking. Key attribute information necessary for compliance verification was manually input to support this process. However, we assumed such information would automatically be extracted from a BIM model in full-scale text. This included spatial dimensions, door clearances, and other relevant parameters extracted from the regulations. The manually entered data was then used to generate Prolog facts, as shown in Figure 5, forming the basis for logical reasoning and compliance evaluation.

```
% Fact: Maximum gradient based on length constraints
max_gradient(Length, 4) :- Length =< 10.  % If length ≤ 10m, max gradient is 4%
max_gradient(Length, 3) :- Length > 10.   % If length > 10m, max gradient is 3%

% Fact: Entrance circulation area dimensions (Example)
circulation_area(entrance1, 150, 150, 9, 3.5).
circulation_area(entrance2, 150, 150, 12, 3).
circulation_area(entrance3, 150, 150, 12, 3.5).
```

Figure 5: Example of generated Prolog facts

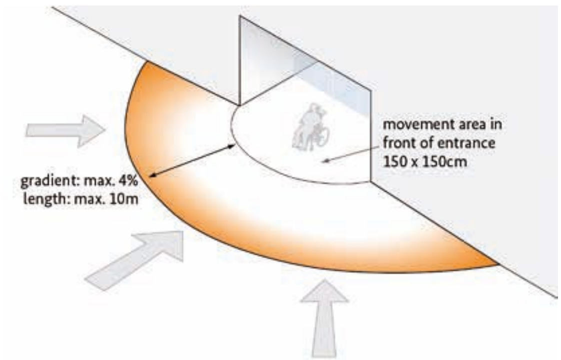
The extracted facts were tested within the SWI-Prolog interpreter (Wielemaker et al., 2012) to secure the executability of generated rules. If errors occurred, such as syntax issues or logical inconsistencies, the rules and queries were iteratively refined using LLM until they were executed correctly. Once a rule was successfully processed without errors, it was stored as ground truth for further evaluation. The examples of rules derived from graphical representations are presented in Figure 6.

```
% Rule: Checking if entrance circulation area meets compliance requirements
compliant_entrance(Entrance) :-
    circulation_area(Entrance, Width, Height, Length, Gradient),
    Width >= 150, Height >= 150,  % Ensure minimum movement area of 150x150 cm
    max_gradient(Length, MaxGrad),
    Gradient =< MaxGrad.
```

(a) Prolog rule

Circulation area at entrance ≤ 4% at a maximum length of 10 m; if length >10 m maximum gradient 3%

(b) Text requirement



(c) Graphic requirement

Figure 6: Example of Prolog rules derived from text and graphical representations (Ministry for the Environment, Nature Conservation, Building and Nuclear Safety, 2015)

Rule derivation using MLLM

To evaluate the MLLM's ability to interpret regulatory rules and generate machine-readable rules, three input conditions were tested: text-only, graphic-only, and text + graphic. In the text-only condition, the model was provided with regulatory clauses in textual requirements without any accompanying graphical representations. In the graphic-only condition, the model received only graphical representations with no textual descriptions, simulating cases where regulatory information is conveyed exclusively through visual elements. In the text

+ graphic condition, the MLLM was given both textual and graphical representations, such as dimensional drawings, annotated drawings, and implementation examples, to determine whether visual support enhanced the model's ability to interpret compliance rules more accurately compared to text alone.

System prompts were designed to ensure consistent MLLM behavior across input types using structured prompting strategies such as role assignment, in-context learning, task definition, input specification, chain-of-thought (Wei et al., 2022), output format, and heuristic prompting, as shown in Table 1.

Table 1: Prompt design for each input case

Component	Text-only	Text + Graphic	Graphic-only
Role Assignment	"You are an expert in Prolog-based rule generation for automated code compliance checking."		
In Context Learning (3 examples)	"Here are the examples of generation: #1 {Clause} #1 {Graphic guideline} #1 {Ground truth Prolog rules} …"		
Task Definition	"Analyze a given clause, generate a Prolog rule."	"Analyze both the clause and graphic guideline, generate Prolog rule."	"Analyze a graphic guideline, generate a Prolog rule."
Input Specification	"Input: Regulatory Clause"	"Input: Regulatory Clause + Graphic guideline"	"Input: Graphic guideline"
Chains of Thought	"Provide a step-by-step explanation of how the rule is derived."		
Output Format	"Generate a finalized rule in a single cell."		
Heuristic Prompt	-	"If multiple rules exist in the graphic guideline, focus on the dominant or relevant single rule and generate a single set of rules."	

The MLLM was assigned the role of an expert in Prolog-based rule generation for automated compliance checking and provided three examples as in-context learning references. Among 23 pairs, a pair from each of 3 different graphic guideline types with ground truth was selected. The model was then instructed to analyze the regulatory input and generate machine-readable rules accordingly. In the text-only case, the model was expected to interpret compliance rules solely from regulatory clauses. In the text + graphic case, both textual and visual descriptions were prompted to be integrated to improve accuracy. In the graphic-only case, the model had to infer logical rule representations directly from graphical representations, testing its ability to process information when textual descriptions were unavailable. To enhance logical consistency, the model was instructed to provide a step-by-step explanation of how the Prolog rule was derived before generating the final output. The response format was structured to produce a finalized Prolog rule in a single output cell, ensuring uniformity across different input conditions. Additionally, in scenarios where multiple rules could be inferred, the additional prompt from the heuristic lessons guides MLLM to focus on the dominant rule and generate a single structured set of Prolog rules. This was particularly relevant in the graphic-only case, where multiple compliance constraints might exist within a single graphic requirement.

Evaluation metrics

To evaluate the accuracy and effectiveness of MLLM-generated rule derivations, this study employed Jaro-Winkler similarity and F1-score metrics, specifically assessing performance at both the predicate and the predicate-element levels. These evaluation methods were adapted from Yang and Zhang's research (Yang and Zhang, 2024) and were chosen to measure how well the generated Prolog rules align with the manually curated ground truth data, both in terms of structural similarity and logical correctness.

Jaro-Winkler similarity was used to assess the string-based similarity between the generated Prolog and ground truth rules. Since Prolog rules can have multiple correct representations that may differ slightly in wording or formatting, Jaro-Winkler similarity helps determine whether the generated rules are semantically close to the expected output, even if minor variations exist. A similarity score closer to 1 indicates a higher match between the generated and reference rule, while a score closer to 0 suggests significant differences.

In addition, this study employed the F1-score to evaluate the correctness of the generated logical components in the Prolog rules. The evaluation was conducted at the predicate level and the predicate-element level. The predicate-level F1-score assesses whether the model correctly identifies key rule components, such as logical conditions and constraints. The predicate-element-level F1-score, on the other hand, evaluates how accurately the model extracts detailed elements within predicates, such as numerical values, dimensions, and compliance parameters. The F1-score is derived from precision and recall.

Experimental results

To evaluate the performance of the MLLM-based rule derivation, experiments were conducted across three different input conditions: text-only, text + graphic, and graphic-only. The overall experimental results are shown in Table 2.

Table 2: Overall experimental results

Metrics	Text-only	Graphic-only	Text + Graphic
Jaro-Winkler Similarity	0.78	0.53	0.82
F1-score (predicate)	0.85	0.65	0.79
F1-score (element)	0.74	0.43	0.96

The text + graphic case consistently outperformed both the text-only and graphic-only cases in the Jaro-Winkler similarity and predicate-element-level F1 score, demonstrating that combining textual and graphic information significantly improves rule derivation accuracy. The graphic-only case performed the worst across all metrics, indicating that the MLLM struggles to infer compliance rules accurately when only graphic information is provided, without textual context.

For Jaro-Winkler similarity, which measures lexical and structural similarity between the generated rules and the ground truth, the text + graphic case achieved the highest score of 0.82, surpassing the text-only case at 0.78 by +0.04. The graphic-only (0.48) was significantly lower, suggesting that textual descriptions ensure structural consistency in rule generation. While graphical representations enhance interpretation, they are not sufficient on their own for accurately structuring compliance rules.

For the predicate-level F1-score, which evaluates the correct identification of key rule components, the text-only case achieved the highest score at 0.85, slightly outperforming the text + graphic case (0.79) by +0.06. The graphic-only case (0.65) lagged, indicating that graphic information alone is less effective in capturing overall rule structures. The text-only case performed slightly better than the text + graphic case at the predicate level, which suggests that introducing graphical representations may slightly affect the model's ability to generalize rule structures, possibly due to inconsistencies in how graphic data is interpreted in combination with text.

However, the predicate-element-level F1-score, which evaluates how well the model captures specific rule details such as numerical values, dimensions, and compliance attributes, demonstrated a clear advantage for the text + graphic case, achieving the highest score of 0.96. This was a significant +0.22 improvement over the text-only case at 0.74 and a significant +0.53 increase over the graphic-only case at 0.43. These results indicate that graphical representations provide supplementary information, particularly for capturing detailed numerical constraints and spatial relationships that are often difficult to extract from text alone.

To further investigate the influence of graphical representations, performance was analyzed separately based on three types of graphical representations: dimensional drawings, annotated drawings, and implementation examples. The results are shown in Table 3.

Table 3: Similarity differences based on type of graphic guideline

Case	Dimensional drawing	Annotated drawing	Implementation example
Text-only	0.77	0.79	0.78
Graphic-only	0.78 (+0.03)	0.72 (-0.07)	0.12 (-0.66)
Text+ Graphic	0.90 (+0.13)	0.92 (+0.13)	0.70 (-0.08)

Among the three graphic guideline types, dimensional and annotated drawings contributed most significantly to performance improvements, whereas implementation examples had limited impact.

Dimensional drawings improved similarity (+0.13 in the text + graphic case compared to the text-only case). This suggests that precise numerical annotations in drawings enhance the MLLM's ability to extract compliance-related attributes and improve structural accuracy.

Annotated drawings also improved performance significantly (+0.13 in the text + graphic case compared to the text-only case). Text descriptions and directional cues in annotated drawings likely helped the model better associate graphical representations with compliance requirements.

Implementation examples, however, did not provide a noticeable advantage and even led to a slight decrease in similarity (-0.08 in the text + graphic case compared to the text-only case). This suggests that real-world images or implementation examples may be more ambiguous and more complex for the MLLM to interpret in the context of regulatory rule generation.

In the graphic-only case, the dimensional drawings achieved the highest similarity score (0.78), while implementation examples performed the worst (0.12, a -0.66 compared to case 1). This further highlights that annotated graphics contribute meaningfully to regulatory rule derivation, whereas real-world implementation images are more challenging for the model to process.

Discussions

This section presents qualitative observations and specific examples to analyze the role of graphical representations in MLLM-based automated rule derivation. While graphical representations can enhance compliance checking by providing spatial and semantic clarity, they can also introduce challenges, such as ambiguity, verbosity, and unintended rule expansion when multiple objects are represented within a single drawing.

Effectiveness of graphical representations in clarifying rule derivation

In some cases, graphical representations contribute to extracting more precise and desirable parameters for rule derivation. For instance, consider the clause: "The side wall grab bar shall be 42 inches (1065 mm) long minimum, located 12 inches (305 mm) maximum from the rear wall and extending 54 inches (1370 mm) minimum from the rear wall." and its corresponding graphic guideline as shown in Figure 6. The drawing graphically represents the grab bar placement requirements, providing a clear spatial understanding of dimensions.

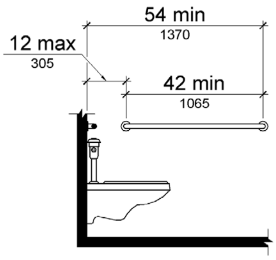


Figure 6: Corresponding dimensional drawing (U.S. Department of Justice, 2010)

As shown in Figure 7, when interpreting the clause using text alone, while this correctly captures the fundamental constraints of the regulation, it lacks the explicit distinction between the grab bar's starting position and ending position relative to the rear wall. This could lead to ambiguity when implementing compliance checks.

```
% Rule to check compliance of side wall grab bar placement
side_wall_grab_bar(Length, DistanceRearWall, Extension) :-
    Length >= 42,
    DistanceRearWall =< 12,
    Extension >= 54.
```

Figure 7: Interpretation results with clause only

By incorporating the dimensional drawing into the rule derivation, a more refined representation emerges, as shown in Figure 8.

```
% Compliance check for side wall grab bar placement
compliant_grab_bar(GrabBarLength, StartDistanceFromRearWall, EndDistanceFromRearWall) :-
    GrabBarLength >= 42,
    StartDistanceFromRearWall =< 12,
    EndDistanceFromRearWall >= 54.
```

Figure 8: Interpretation results integrating clause and graphic guideline

Here, integrating the graphic guideline helps clarify the semantic-rich properties for compliance checking. The StartDistanceFromRearWall (≤ 12 inches) represents where the grab bar begins. The EndDistanceFromRearWall (≥ 54 inches) specifies the total extension of the bar. The GrabBarLength (≥ 42 inches) ensures compliance with the minimum required length.

Challenge of graphical representations in rule derivation

While graphics can be beneficial, they do not always contribute effectively to automated rule derivation. In cases where multiple objects and annotations are present in a single drawing, the model may generate verbose or overly complex Prolog rules that exceed the scope of the regulatory clause.

For instance, consider the clause: "A turning radius of 1,800 mm by 1,800 mm should be available in at least one accessible bathroom in the dwelling unit." And its corresponding annotated drawing, as shown in Figure 9.

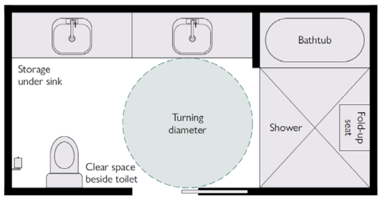


Figure 9: Corresponding annotated drawings (Nature Conservation, Building and Nuclear Safety, 2015)

The corresponding graphic guideline includes an annotated drawing specifying the required turning space with multiple notations of objects.

When the MLLM interprets the rule solely on the clause, this approach effectively captures the essential compliance check, as shown in Figure 10.

```
% Rule to check if a dwelling unit has at least one accessible bathroom with the required turning radius
has_accessible_bathroom(DwellingUnit) :-
    accessible_bathroom(DwellingUnit, _, RadiusX, RadiusY),
    RadiusX >= 1800,
    RadiusY >= 1800.
```

Figure 10: Interpretation results with clause only

When the annotated drawing is incorporated without heuristic guidance, as shown in Table 1, the extracted Prolog rules become more detailed but excessively verbose, as shown in Figure 11. This approach introduces additional conditions inferred from the drawing (e.g., clear space beside the toilet, fold-up seat), which may or may not be relevant to the original clause. The lack of specificity in rule prioritization results in an unnecessarily complex interpretation, making compliance validation cumbersome.

```
% Rule: A bathroom is accessible if it meets the turning radius requirement
accessible_bathroom(Bathroom) :-
    turning_radius(Bathroom, Width, Length),
    min_turning_radius(MinWidth, MinLength),
    Width >= MinWidth,
    Length >= MinLength,
    has_clear_space_beside_toilet(Bathroom),
    has_fold_up_seat(Bathroom).
```

Figure 11: Interpretation result integrating clause and graphic requirement without heuristic prompt

To refine rule extraction, a heuristic prompt is applied, guiding the model to prioritize key compliance criteria while avoiding excessive rule expansion. This ensures that the primary requirement (turning radius) is correctly extracted without introducing extraneous conditions, as shown in Figure 12. This case highlights the importance of structured prompting techniques when integrating graphical representations to avoid unintended rule expansion.

```
% Compliance Rule: The accessible bathroom must provide a turning radius of at least 1800 mm.
compliant_turning_radius(BathroomID) :-
    bathroom_turning_radius(BathroomID, Diameter),
    Diameter >= 1800.
```

Figure 12: Interpretation result integrating clause and graphic requirement with heuristic prompt

Impact of graphical representations without numeric values

This study primarily focused on regulatory clauses that define explicit, quantifiable variables for compliance checking. However, when implementation examples lack numerical information, automated rule derivation tends to degrade in accuracy.

For instance, consider the clause: "Staircases shall be provided with a minimum tread width of 300mm and maximum riser height of 150mm." and its corresponding implementation example, as shown in Figure 13.

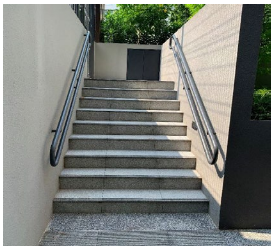


Figure 13: Corresponding implementation example (Nature Conservation, Building and Nuclear Safety, 2015)

This requirement is illustrated in Figure 13, which depicts an implemented staircase example. While this visual representation aids human understanding, the information extracted from the image may lead to unnecessary rule generation due to including non-regulatory elements.

When the compliance rule is derived solely from the clause, it is structured as shown in Figure 14. This rule strictly follows the regulatory clause, ensuring only the required parameters (tread width and riser height) are considered.

```
% Rule to check compliance of staircases based on tread width and riser height
staircase_compliant(TreadWidth, RiserHeight) :-
    TreadWidth >= 300,
    RiserHeight =< 150.
```

Figure 14: Interpretation result with clause only

When incorporating the implementation example, the automated system detects additional features in the image, leading to a more complex but undesirable rule as shown in Figure 15. Here, handrails, which were not mentioned in the original regulation, have been incorrectly inferred as a compliance requirement.

```
compliant_staircase(StairID, TreadWidth, RiserHeight, Handrails) :-
    TreadWidth >= 300,
    RiserHeight =< 150,
    Handrails >= 2.
```

Figure 15: Interpretation results integrating clause and graphic requirement

Graphical representations representing multiple rules

A significant portion of graphical representations in regulations represent multiple rules simultaneously. Some graphical representations stand alone as rules without corresponding text-based clauses. MLLMs have demonstrated the ability to distinguish individual rules within such graphical representations and interpret them separately.

For instance, Figure 16 shows the graphical representations supporting the clause: "*Accessible* toilet stall doors shall provide a clear opening of at least 900 mm (35 in.) with the door in the open position.

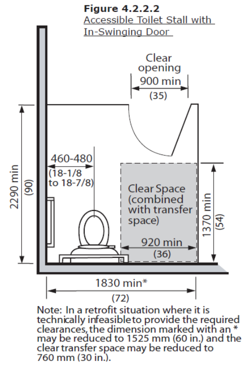


Figure 16: Example of graphical representations containing multiple rules (City of London, 2007)

This annotated drawing represents six rules at once, including: Door width compliance, Minimum stall dimensions, Grab bar placement, Toilet clearance space, Sink positioning, and Turning radius requirements. The presence of multiple rules affects the accuracy of interpretation in specific use cases, as the model must determine which parameters are relevant to the current query. When interpreting the graphic guideline alone, the model extracts all six rules, as shown in Figure 17, which may be unnecessary for a targeted compliance check.

```
% Rule 1: Accessible Toilet Stall Door Clearance
% Ensures the stall door provides a minimum clear opening of 900 mm (35 in.)
toilet_stall_door_clearance(Width) :-
    Width >= 900.

% Rule 2: Minimum Stall Depth Requirement
% The stall must have a minimum depth of 1830 mm (72 in.), but may be reduced to 1525 mm (60 in.) in retrofit situations
toilet_stall_depth(Depth, retrofit(false)) :-
    Depth >= 1830.

toilet_stall_depth(Depth, retrofit(true)) :-
    Depth >= 1525.

% Rule 3: Minimum Stall Width Requirement
% The stall must have a minimum width of 920 mm (36 in.)
toilet_stall_width(Width) :-
    Width >= 920.

% Rule 4: Toilet Positioning
% The toilet must be placed 460-480 mm (18-1/8 in. to 18-7/8 in.) from the nearest side wall.
toilet_position(Distance) :-
    Distance >= 460, Distance =< 480.

% Rule 5: Clear Transfer Space Requirement
% The transfer space must be at least 920 mm (36 in.), but may be reduced to 760 mm (30 in.) in retrofit situations.
clear_transfer_space(Space, retrofit(false)) :-
    Space >= 920.

clear_transfer_space(Space, retrofit(true)) :-
    Space >= 760.

% Rule 6: Retrofit Exception Compliance
% Determines whether a stall qualifies for a retrofit exception based on feasibility
retrofit_exception(Feasibility) :-
    Feasibility = true.

% Compliance Check Aggregator
% Verifies if the toilet stall meets all accessibility requirements
toilet_stall_compliance(Width, Depth, DoorWidth, ToiletDist, TransferSpace, Retrofit) :-
    toilet_stall_width(Width),
    toilet_stall_depth(Depth, Retrofit),
    toilet_stall_door_clearance(DoorWidth),
    toilet_position(ToiletDist),
    clear_transfer_space(TransferSpace, Retrofit).
```

Figure 17: Interpretation result with visual only

Conclusions

This study conducted a preliminary exploration of integrating graphical representations into MLLM-based automated rule derivation for design compliance checking. Experimental results demonstrated that combining textual and visual inputs enhances rule derivation accuracy, particularly in extracting specific compliance parameters such as numerical constraints and spatial relationships. However, the graphic-only condition resulted in significantly lower performance.

Despite these findings, several limitations remain. The dataset was relatively small and focused on accessibility regulations, limiting the generalizability of the findings.

Expanding the dataset to include a broader range of regulatory domains and incorporating real-world BIM data will be essential for more robust validation. Additionally, while Prolog was chosen for its simplicity and suitability for early-stage experiments, adopting more interoperable rule representation formats like LegalRuleML could improve integration with existing compliance checking systems. Moreover, the limited performance of the graphic-only case also suggests that relying solely on MLLM-generated text descriptions may fail to capture the full spatial and contextual richness of graphical representations. Future research should explore advanced techniques such as scene-graph extraction, object detection, and structured semantic parsing to enhance visual interpretation. Lastly, validating the execution of generated rules within BIM-based compliance workflows will be critical to ensure their practical applicability in real-world settings.

Acknowledgments

This work was supported by the Hans Fischer Senior Fellowship program at the Technical University of Munich - Institute for Advanced Studies (TUM-IAS) in Germany and by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant, funded by the Ministry of Land, Infrastructure and Transport (No. RS-2024-00407028).

References

City of London (2007) Facility Accessibility Design Standards. City of London, Ontario. https://www.oxfordcounty.ca/en/yourgovernment/resources/Accessibility/FADS_2007_final.pdf

Doris, A.C., Grandi, D., Tomich, R., Alam, M.F., Ataei, M., Cheong, H. & Ahmed, F. (2024) DesignQA: A Multimodal Benchmark for Evaluating Large Language Models' Understanding of Engineering Documentation. *Journal of Computing and Information Science in Engineering*, 25(2), 021009.

Eastman, C., Lee, Jae-min, Jeong, Y. & Lee, Jin-kook, (2009) Automatic rule-based checking of building designs. *Automation in Construction*, 18, 1011–1033.

Federal Ministry for the Environment, Nature Conservation, Building and Nuclear Safety (2015) *Guideline Accessibility in Building Design*. BMUB, Berlin, Germany. https://www.leitfadenbarrierefreiesbauen.de/fileadmin/downloads/archiv/barrierefreies_bauen_leitfaden_en_bf_version2.pdf

Fuchs, S. & Amor, R. (2021) Natural Language Processing for Building Code Interpretation: A Systematic Literature Review. *Proceedings of the 38th International Conference of CIB W78, 294-303.*

Fuchs, S., Witbrock, M., Dimyadi, J. & Amor, R. (2024) Using Large Language Models for the Interpretation of Building Regulations. *Journal of Engineering, Project, and Production Management*, *14*(4), 1-12

Mu, Y., Chen, J., Zhang, Q., Chen, S., Yu, Q., Ge, C., Chen, R., Liang, Z., Hu, M., Tao, C., Sun, P., Yu, H., Yang, C., Shao, W., Wang, W., Dai, J., Qiao, Y., Ding, M. & Luo, P. (2024) RoboCodeX: Multimodal Code Generation for Robotic Behavior Synthesis. *arXiv preprint arXiv:2402.16117*.

U.S. Department of Justice (2010) ADA Standards for Accessible Design. U.S. Department of Justice. https://www.ada.gov/2010ADAstandards_index.htm.

Wan, Y., Wang, C., Dong, Y., Wang, W., Li, S., Huo, Y. & Lyu, M.R. (2024) Automatically Generating UI Code from Screenshot: A Divide-and-Conquer-Based Approach. *arXiv preprint arXiv:2406.16386*.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V. & Zhou, D. (2022) Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in neural information processing systems*, *35*, 24824-24837.

Wielemaker, J., Schrijvers, T., Triska, M. &Lager, T. (2012) SWI-Prolog. *Theory and Practice of Logic Programming*, *12*(1-2), 67-96.

Wu, C., Ge, Y., Guo, Q., Wang, J., Liang, Z., Lu, Z., Shan, Y. & Luo, P. (2024) Plot2Code: A Comprehensive Benchmark for Evaluating Multi-modal Large Language Models in Code Generation from Scientific Plots. *arXiv preprint arXiv:2405.07990*.

Yang, F. & Zhang, J. (2024) Prompt-based automation of building code information transformation for compliance checking. *Automation in Construction*, *168*, 105817.

Ying, H. & Sacks, R. (2024) From Automatic to Autonomous: A Large Language Model-driven Approach for Generic Building Compliance Checking. *Proceedings of the 41st International Conference of CIB W78.*

Zhang, J. (2023) How Can ChatGPT Help in Automated Building Code Compliance Checking? *Proceedings of the 40th International Symposium on Automation and Robotics in Construction,* 63-70

Zheng, Z., Chen, K.-Y., Cao, X.-Y., Lu, X.-Z. & Lin, J.-R. (2023) LLM-FuncMapper: Function Identification for Interpreting Complex Clauses in Building Codes via LLM. *arXiv preprint arXiv:2308.08728*.

Zentgraf, S., König, M. (2025) Enhancing Information Extraction from Building Standards with ChatGPT-4: A Multimodal Approach*, International Conference on Computing in Civil and Building Engineering*, 234–247.