Joint CSCE Construction Specialty & CRC Conference 2025
Conférence conjointe spécialisée en construction de la SCGC et CRC-2025

Montreal, Quebec
July 28-31, 2025 / *28-31 juillet 2025*

Self-Evaluation of Single and Multi-Agent LLMs as Assistants in Inspection Reporting Workflows

S.H. Hsu[1*†], Y. Jung[1†], J. Fu[1], and M. Golparvar-Fard[1]

[1] Dept of Civil & Environmental Engineering, University of Illinois Urbana-Champaign, IL, USA
† Authors contributed equally

ABSTRACT: Recently, ChatGPT and various large language models (LLMs) have demonstrated remarkable success in natural language understanding, enhancing inspection reporting workflows as assistants. By leveraging the robust text semantics of LLMs, the development of multi-modal models has further expanded their applications to a wide range of scenarios, including visual inspection of structure defects and construction progress. Current off-the-shelf LLMs can be categorized into two types: commercial models and open-weights models. While commercial models typically require cloud computing and necessitate sharing private data with service providers, open-weight models can be deployed locally without privacy concerns. In the construction industry, where project data is often highly confidential, local LLMs present a more promising option. However, few research focuses on the deployment of open-weights LLMs because of their relatively lower performance. Without additional engineering optimizations, this performance gap can become even more pronounced, making them less capable for the assigned tasks. This paper proposes a multi-agent system based on Llama 3.2 vision model and compares its performance with a single-agent system based on pseudo ground-truth reports from OpenAI's GPT-4o for inspection reporting workflows. The evaluated tasks include bridge inspection, building inspection, indoor progress reporting, outdoor progress reporting, and facility management inspection. For each task, 20 images are collected to facilitate the comparison. A self-evaluation strategy is proposed to assess the model capability from different perspectives.

1. INTRODUCTION

Inspection reporting workflows are often labor-intensive and error-prone, as many firms continue to rely on traditional manual methods. Labor shortages further exacerbate this issue, leading to skill gaps among workers, which result in inconsistent standards and misunderstanding of actual conditions. Large language models (LLMs) have demonstrated promising performance in assisting inspection reporting workflows with the high reasoning capability (Pu et al. 2024; Chen et al. 2025). In recent years, multi-modal LLMs have expanded applications in a wide range of tasks. However, potential hallucinations in image-based inspection reports can mislead engineers' judgements, causing incorrect countermeasures. These uncertainties can increase communication costs, cause scheduling delays, and prompt passive responses (Jung et al. 2024). Retrieval-augmented generation (RAG) has been proven effective in addressing the issue by retrieving the most relevant examples based on user requests and incorporating them into prompts. Several applications, including commercial LLMs, have adopted this technique to mitigate hallucinations (Asai et al. 2023; Hoshi et al. 2023).

While RAG can enhance LLMs in inspection reporting workflows by integrating external knowledge bases, the project data are usually considered confidential, and the access is limited. Companies are reluctant to share information with commercial LLMs, limiting the potential benefits of RAG. In that way, applications must rely solely on the inherent capabilities of commercial LLMs, which may not be tailored to specific inspection scenarios. Few research focuses on evaluating the multi-modal understanding on those scenarios, and the risks of hallucinations remains unknown. Even minor inaccuracies can lead to significant safety concerns and increased costs. Due to the reasoning abilities of LLMs, detecting these errors can be challenging, potentially resulting in additional time and resources spent on verifying outputs.

In addition to commercial LLMs, open-weight models that can be served as local LLMs have emerged as promising alternative for safely incorporating RAG. Interest in these models has grown following the release of DeepSeek-R1 (Guo et al., 2025). Nevertheless, open-weight models generally have relatively lower performance than commercial LLMs, which benefit from the most up-to-date training data and extensive engineering optimizations, including carefully designed prompts and advanced RAG strategies. Furthermore, unlocking the full potential of LLMs, particularly larger variants, demands significantly greater computing resources. To address these challenges, this paper explores the feasibility of a multi-agent system as a local LLM, where multiple LLM instances collaborate on a single task through a communication protocol and finally conclude a final response to a user. Studies have shown that the systems composed of weaker LLMs can even compete with commercial LLMs in complex tasks (Lee et al. 2023, McAleese et al. 2024).

This paper examines five critical scenarios for inspection reporting: bridge inspection, building inspection, indoor construction progress reporting, outdoor construction progress reporting, and built-environment facility management inspection (Hsu et al. 2023; Núñez-Morales et al. 2023). An open-weight LLM is used to implement a single-agent system with RAG (RAG-SAS) and a multi-agent system (MAS). Among the available multi-modal models, *Llama-3.2-11B-Vision-Instruct* (Meta AI 2024) from the Llama 3.2 series is selected to meet the limitation of computing resources because the model can be hosted on a single Nvidia GeForce RTX 3090 GPU. Two self-evaluation strategies are proposed to evaluate and compare the performance of these systems. All the results and datasets mentioned in this paper are available in the shared link: (https://drive.google.com/drive/folders/1U4sncLvA_pI5HgqQ2eSMWby9BtFv7NCJ?usp=sharing).

2. RELATED WORK

2.1 LLMs in Inspection Reporting Workflows

LLMs are increasingly being applied in the construction industry for planning, information management, and automation. For construction scheduling, Amer et al. (2023) fine-tuned GPT-2 to identify sequencing constraints, while He et al. (2024) employed GPT-4 for constraint classification. In Building Information Modeling (BIM), LLM-driven frameworks facilitate question classification, information retrieval, and model updates through natural language processing (Du et al. 2024; Jang et al. 2024; Autodesk University 2024). Automated reporting systems leverage OpenAI's GPT models to generate reports from visual data and voice logs, either through fine-tuning or direct use (Pu et al. 2024; Chen et al. 2025). Additionally, Jung et al. (2024) introduced a vision-language model designed to automate the generation of textual descriptions from visual data for daily reporting. Heo (2025) proposed a theoretical framework for LLM adoption in the Architecture, Engineering, and Construction (AEC) industry.

Despite these advancements, LLM applications in construction remain limited. Current implementations often fail to analyze domain-specific knowledge in general domain LLMs, reducing practical applicability. Systematic evaluation of LLM outputs is lacking, raising concerns about the reliability of LLM applications. Moreover, integration efforts of LLMs are typically confined to isolated tasks, posing challenges in interpretability and validation in various real-world scenarios. Lastly, most studies rely on commercial LLMs, incurring significant costs without exploring their foundational knowledge across diverse construction-related tasks. Addressing these limitations is essential for optimizing LLM applications in construction.

2.2 Retrieval Augmented Generation (RAG) for LLMs

While LLMs are first pre-trained on web-scale datasets and then fine-tuned to follow instructions, the learned knowledge is not unlimited and may generate hallucinations when handling requests outside their domain. To mitigate this issue, RAG has been extensively studied as a method to enhance LLMs by providing relevant demonstrations (Asai et al. 2023; Hoshi et al. 2023). RAG works via several components: a knowledge base serving as an example pool, semantic representations used as search keys, similarity functions for ranking examples, and prompt engineering to integrate retrieved information. Among these, embedding models play a crucial role in the pipeline. The more accurately examples from a knowledge base are represented, the more useful the incorporated retrievals become, resulting in higher-quality responses. Additionally, several supporting applications, such as vector databases and search engines, have become increasingly available. This paper examines the impact of RAG on a single-agent system by comparing it to a multi-agent system and explores the feasibility of incorporating expert reports to ensure alignment with the desired reporting style.

2.3 Multi-Agent System and Self-Evaluation

As LLMs become superintelligent, evaluating their factual accuracy and reasoning in inference-only settings has become increasingly challenging. Kenton et al. (2024) explored various communication protocols like debate to create multi-agent systems and investigated oversight methods for accurately supervising superhuman AI. Their findings showed that weaker LLMs can effectively evaluate more advanced models and provide feedback that surpasses human annotations. Furthermore, reinforcement learning from AI feedback has been shown to be an effective self-reflection mechanism for determining reward of actions (Lee et al. 2023; McAleese et al. 2024). Collaboration-based communication protocols, which mimic human teamwork, have been studied to enhance weaker LLMs and expand their potential applications (Guo et al. 2024).

Multi-agent systems offer a promising solution to overcome computational bottlenecks, enabling them to compete with commercial LLMs after multiple iterations. The success of these protocols highlights the potential of LLM self-evaluation, providing insightful interpretations of their capabilities. Building on this idea, this paper proposes two self-evaluation strategies that compare LLM-generated reports with expert reports to reach automated feedback on reporting accuracy and reliability. Through this approach, this paper explores and validates the feasibility of multi-agent systems as local LLMs, demonstrating a cost-effective alternative to commercial solutions.

3. METHODS

This paper develops both single- and multi-agent systems based on the open-weight LLM, *Llama-3.2-11B-Vision-Instruct* (Meta AI 2024), for deployment as local LLMs. In the single-agent system, RAG is incorporated to enhance response quality while the multi-agent system relies entirely on a collaboration mechanism between agents. A knowledge base is constructed by collecting images of selected tasks for inspection reporting workflows and generating corresponding ground truth data with the assistance of the commercial LLM, GPT-4o from OpenAI. In this framework, agents are defined as LLM instances assigned with different role profiles and instruction prompts. Self-evaluation strategies are conducted using GPT-4o, which scores and compares the responses of each system based on pre-defined criteria. Figure 1 presents an overview of the proposed methods and pipeline.

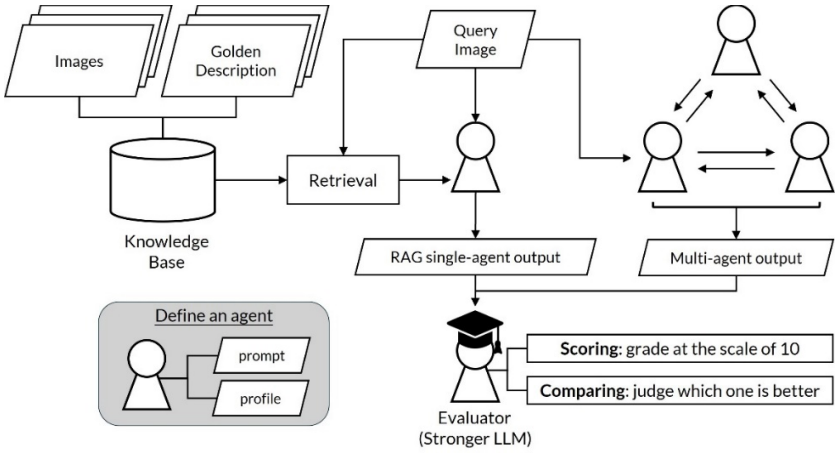

Figure 1: The method overview

3.1 RAG Single-agent System (RAG-SAS)

This paper employs *Llama-3.2-11B-Vision-Instruct* (Meta AI 2024) to implement the single-agent system and integrate it with RAG using the proposed knowledge base. Because of the limited size of the knowledge base and the risk of long-context inputs distracting the model from its original instructions, the system considers only the top-1 retrieval as a demonstration. Cosine similarity is utilized to rank relevant examples of each task. To decide the most effective embedding model, the retrieval pool is initially not divided into smaller subsets for intentionally challenging the models with additional noises from different tasks. In the experiment, retrievals are restricted to the same task category, as output formats vary across tasks. Also, retrieval is conducted offline, eliminating the need for a vector database or a searching mechanism within the scope of this paper. The details of the dataset and a comparison analysis of multiple embedding models are elaborated in Section 4.

3.2 Multi-agent System (MAS)

This paper employs a multi-agent system (MAS) approach to enhance the accuracy, consistency, and reliability of LLM applications in construction-related tasks. By integrating specialized LLM-based agents, the system enables structured data assessment, validation, and refinement, improving report quality through iterative collaboration. Because of computational constraints, the MAS framework is implemented to have two primary agents: Field Engineer (FE) and Site Superintendent (SS). Each agent is assigned a distinct role with profiling:

- FE: Conduct the initial technical assessment based on given visual information and predefined assessment criteria
- SS: Verify and refine the FE's assessment, identifying risks, compliance concerns, and scheduling issues while providing actionable insights.

This sequential collaboration process mirrors real-world construction workflows, where field engineers conduct inspections and site superintendents ensure site operations, safety, quality control, and team supervision. To accommodate the limited space of this paper, role descriptions and prompts used for each agent's assessment process are provided in a shared link.

3.3 Self-evaluation Agent

Because of the superintelligence of LLMs, comparing responses from two models is challenging and requires significant human labor. In line with the common practice of self-evaluation in the research community, this paper employs GPT-4o as the evaluation agent to compare the outputs of RAG-SAS and MAS. On one hand, a scoring-based evaluation is conducted. In addition to an overall score (OVR), five evaluation criteria—accuracy (ACC), completeness (COM), consistency (CON), explainability (EXPL), and alignment (ALN)—are proposed. These criteria help break down reasoning capabilities into varied perspectives, allowing the evaluator to assign scores on a 10-point scale. The criteria are adapted to

different tasks, incorporating task-specific instructions where necessary. On the other hand, another prompt for a direct comparison is also developed. Instead of assigning scores, which may introduce ambiguity due to style preferences and score variability, the evaluator is instructed to determine which response is superior based on the same evaluation criteria. Note that no image is attached to simplify the evaluation task, and only the generated responses are evaluated against the pseudo ground truth by GPT-4o. As a result, two single-modality evaluation strategies are applied. The detailed evaluation prompts are available via the shared drive link.

4. EXPERIMENTS AND RESULTS

4.1 Datasets

A new image dataset is collected to validate and compare the proposed systems and self-evaluation method. As shown in Figure 2, this paper focuses on inspection reporting workflows of five tasks: bridge inspection, building inspection, indoor construction progress reporting, outdoor construction progress reporting, and built-environment facility management. For each task, 20 images are collected from our own source, public datasets (Liu et al. 2020, Xiao et al. 2022, Zhai et al. 2023), and web searches that are free of use. In total, 100 images are collected, and the corresponding descriptions are first generated by GPT4-o using carefully designed prompts and then manually reviewed to serve as the ground truth for RAG and self-evaluation. The dataset and prompts are publicly available via the shared link.

Figure 2: The demonstration of collected images for the selected civil engineering tasks

4.2 Embedding Model for RAG

For RAG, embeddings from deep learning models are widely used to represent the semantics of examples, and similarity searching can be applied to retrieve the most relevant ones. After reviewing different embedding models, this paper selects Visualized BAAI General Embedding (BGE) (Zhou et al. 2024), which have deep integration of image and text to better encode multimodal inputs. To compare the performance, visualized BGE are used to encode images and images with corresponding text descriptions, respectively. The text descriptions for each image are generated using *Llama-3.2-11B-Vision-Instruct* with the simple prompt, "*Describe the image in detail.*" The output is limited to 1024 tokens. As a baseline, image embeddings are obtained using the CLS token from the base Llama vision model. These three retrieval strategies are evaluated on 100 images using task categories as labels. Figure 3 shows the top-k accuracy results, where k ranges from 1 to 20. Since each task category contains 20 images, each image can have up to 19 correct retrievals. The results show that visualized BGE outperforms the baseline, and encoding images along with their generated text descriptions provides richer semantics, improving retrieval accuracy—particularly for scenes with visual similarities. Figure 4 illustrates that additional descriptions generated by Llama enhance retrieval robustness. Consequently, the retrieval approach based on images and their generated text descriptions is adopted for RAG in SAS to finally generate outputs for the comparison with MAS.

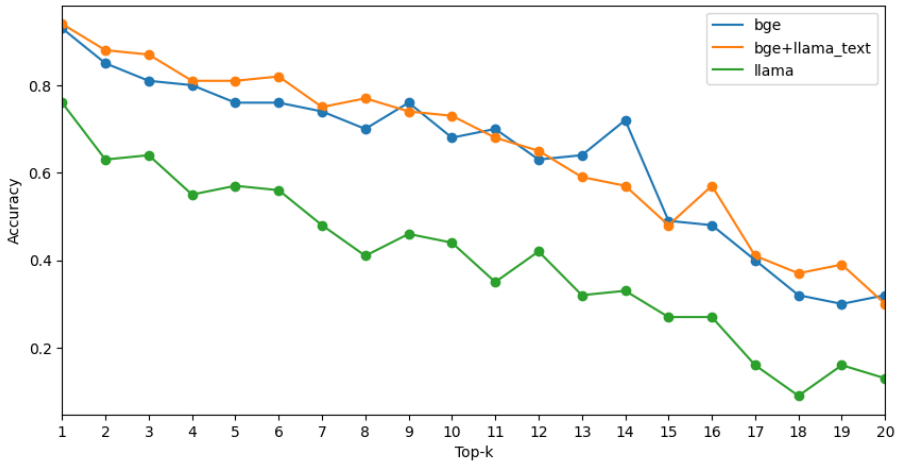

Figure 3: Top-k retrieval accuracy of different embeddings

Figure 4: Retrieval results of embeddings from image-only and image-with-text inputs

4.3 Self-evaluation of RAG-SAS and MAS

Table 1 and Table 2 present the evaluation results using (1) scoring- and (2) comparing-based prompts. As seen, the scoring-based evaluation (Table 1) indicates that RAG-SAS consistently outperforms MAS across all tasks, particularly in progress reporting and facility management. A deeper analysis evaluation (Table 2) further shows this trend, showing that RAG-SAS dominates in both outdoor and indoor progress reporting and facility management while being competitive with MAS in bridge and building defect inspection tasks.

This performance difference may stem from the underlying model's capabilities. The Llama-3.2-Vision-Instruct model appears to have strong visual grounding for bridge and building images, leading to comparable performance between the two methods in defect inspection. This enables multi-agent discussions that refine inspection results. However, for indoor and outdoor progress reporting and facility management, MAS struggles significantly, as reflected in both scoring (Table 1) and win-based comparison (Table 2). This disparity could be attributed to the model's lack of inherent visual grounding for various built environments, which RAG-SAS mitigates through domain knowledge injection via RAG. Despite being based on the lightweight open-source Llama-3.2-11B-Vision-Instruct model, RAG-SAS effectively compensates for these limitations, enabling stronger performance in tasks requiring deeper contextual understanding beyond visual processing. More details are discussed in Section 5.2.

Additionally, Table 1 highlights that *Llama-3.2-11B-Vision-Instruct* performs well in consistency (CON) and explainability (EXPL) but struggles relatively more with alignment (ALN), suggesting that while the model provides coherent and interpretable outputs, there may be challenges in ensuring its responses align well with ground truth or expected reasoning pathways described in the given prompts.

Table 1: Results of scoring evaluation by GPT-4o

Task	RAG-SAS						MAS					
	OVR	ACC	COM	CONS	EXPL	ALN	OVR	ACC	COM	CONS	EXPL	ALN
Bridge insp.	5.8	5.2	5.0	6.7	7.7	4.8	5.4	5.0	4.5	6.8	6.7	4.2
Building insp.	5.4	5.0	4.8	6.7	7.6	4.4	4.8	3.9	4.3	5.8	6.4	3.6
Indoor prog.	6.0	5.7	5.3	6.8	7.0	5.2	4.3	3.6	4.1	5.4	4.8	3.4
Outdoor prog.	7.7	8.0	7.8	8.0	8.2	6.7	4.9	4.2	4.6	5.9	5.8	3.6
Facility Mgmt.	6.0	5.8	5.2	7.0	7.3	4.8	4.6	4.7	3.6	5.8	5.0	4.0
Avg.	**6.2**	**5.9**	**5.7**	**7.0**	**7.5**	**5.2**	**4.8**	**4.3**	**4.2**	**5.9**	**5.8**	**3.8**

Table 2: Results of comparing evaluation by GPT-4o

Task	Wins	OVR (#)	ACC (#)	COM (#)	CONS (#)	EXPL (#)	ALN (#)
Bridge insp.	RAG-SAS	11	11	11	10	5	11
	MAS	9	9	8	9	7	9
	Tie	-	-	1	1	8	-
Building insp.	RAG-SAS	8	8	9	8	9	8
	MAS	7	5	4	5	5	5
	Tie	5	7	7	7	6	7
Indoor prog.	RAG-SAS	17	14	16	15	14	17
	MAS	3	3	4	5	5	3
	Tie	-	3	-	-	1	-
Outdoor prog.	RAG-SAS	19	19	19	19	16	19
	MAS	1	1	1	1	-	1
	Tie	-	-	-	-	4	-
Facility Mgmt.	RAG-SAS	16	16	17	15	17	16
	MAS	3	4	3	2	2	4
	Tie	1	-	-	3	1	-
Avg.	RAG-SAS	**14.2**	**13.6**	**14.4**	**13.4**	**12.2**	**14.2**
	MAS	**4.6**	**4.4**	**4.0**	**4.4**	**3.8**	**4.4**
	Tie	**1.2**	**2.0**	**1.6**	**2.2**	**4.0**	**1.4**

5. DISCUSSION

5.1 Retrieval Analysis

In the experiment, a limitation was introduced to ensure that retrievals come from the same task category. To assess the impact of retrieval quality, this paper removes the limitation to use originally incorrect retrievals when using all the data as one whole pool, and more retrievals instead of just one are used to generate another batch of outputs for qualitative analysis. The generated outputs are available via the shared link. For incorrect retrievals, errors led the agent to generate irrelevant content alongside the report, distracting the agent from filling valid values in the report columns. Similarly, when more retrievals were inserted, the lack of a memory mechanism in the model caused long-context prompts to confuse the agent, making it forget the original instructions. As a result, the generated reports often replicated one of the retrieved examples rather than producing a properly reasoned output. Since open-weight LLMs are currently less capable than commercial LLMs, the limitations of RAG usage must be further examined to optimize performance for building local LLMs.

5.2 RAG-SAS vs. MAS

The quantitative results in Tables 1 and 2 show that RAG-SAS outperforms MAS across all tasks, while remaining competitive in bridge and building defect inspection tasks. Building on these findings, this section provides a qualitative comparison of RAG-SAS and MAS, especially for bridge inspection and indoor progress monitoring. Figure 5 illustrates representative examples of the outputs from both methods,

alongside the ground truth generated by GPT4-o and photographs of the inspection scenes.

Bridge Defect Inspection: As shown in Figure 5, both RAG-SAS and MAS methods capture key defects, but RAG-SAS provides a more structured report. It correctly identifies corrosion and spalling but omits deterioration and underestimates severity as moderate instead of critical. MAS correctly assigns critical severity but fails to mention spalling and deterioration, reducing completeness.

Indoor Progress Monitoring: RAG-SAS outperforms MAS in indoor progress monitoring by producing more consistent and structured reports. While MAS correctly identifies certain elements (e.g., wall framing and electrical wiring), it also introduces clear inconsistencies, such as listing flooring laying as an ongoing activity, which is not present in the given image. On the other hand, RAG-SAS follows the GPT-4o ground truth more closely, mentioning ceiling work, which may have been inferred from partial or planned tasks. However, the inclusion of ceiling grid installation in GPT-4o's pseudo ground truth is debatable, as the image does not clearly show this activity. Instead, it appears that the ceiling grid was already installed and awaiting ceiling finishes. This raises the possibility that GPT-4o's ground truth itself contains a hallucinated element, which RAG-SAS then incorporates as supplementary information.

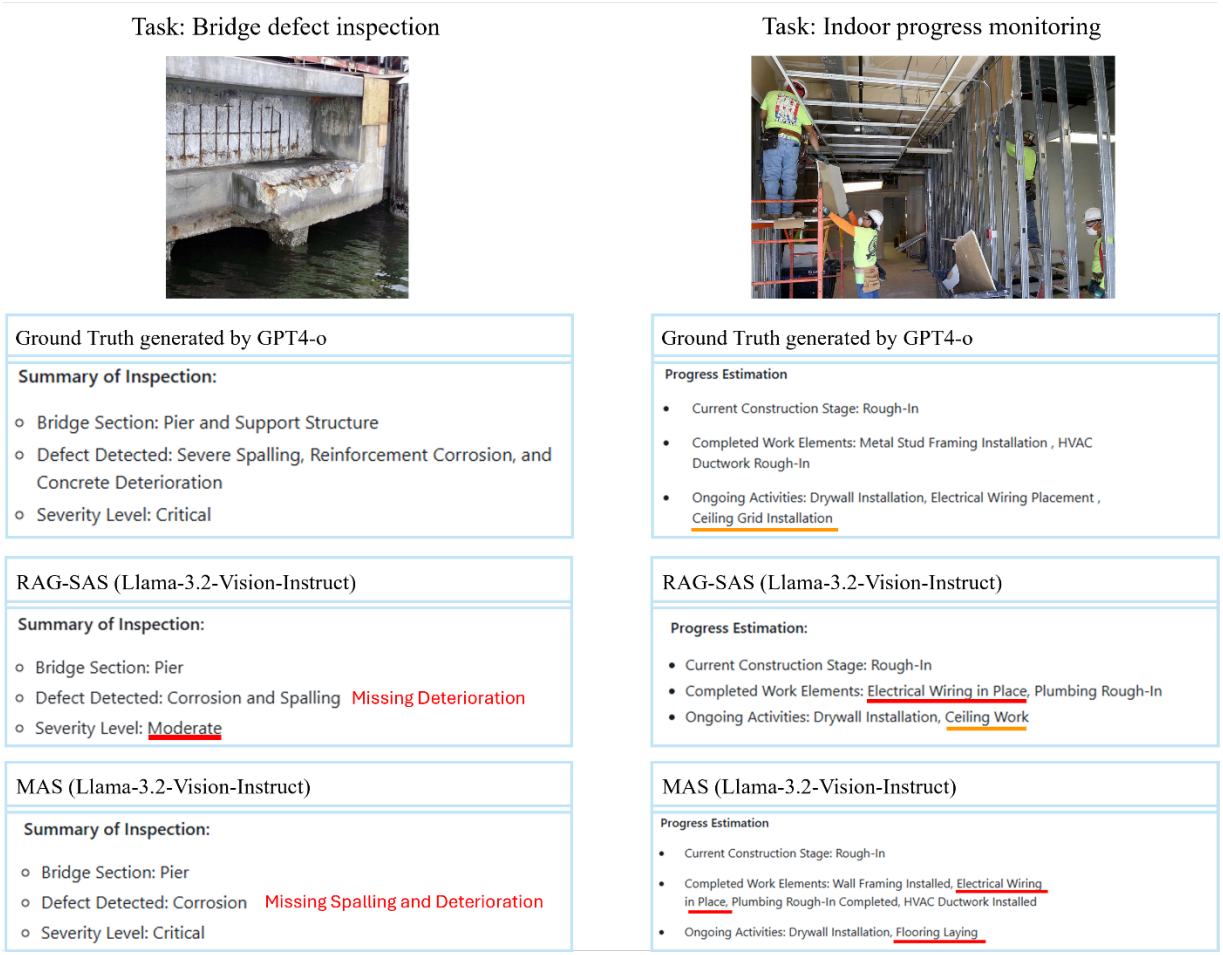

Figure 5: Qualitative results from bridge inspection and indoor progress monitoring reports. The orange color-coded underline indicates uncertain information (likely hallucinated), while the red underline marks incorrect inspection results.

5.3 Scoring- vs. Comparing-based Self-evaluation

As presented in Tables 1 and 2, the evaluation results from both strategies are consistent with each other. Because score differences can sometimes be abstract and ambiguous, a direct comparison task allows the

agent to explain these differences more clearly. For instance, consider an evaluation result from the defect inspection task:

- Scoring:
  - RAG-SAS: *Needs significant improvement. The report should correctly identify the affected sections, focus on water-related deterioration, and align its severity assessment with the reference findings.*
  - MAS: *The report requires major improvements in defect identification, location accuracy, and alignment with the reference report. Correctly identifying water-related issues and aligning recommendations with the reference report would greatly enhance its reliability and usefulness.*
- Comparing:
  - *Generated Report 2* (MAS) *is the better report overall, as it correctly identifies a key defect related to the reference findings. While it does not include all details, it is more aligned with the reference report compared to Report 1* (RAG-SAS)*, which misidentifies defect locations and structural elements.*

Although both lack some important details and received the same score, MAS is considered superior because RAG-SAS failed to follow the instruction and did not even generate its response in the required JSON format. This paper presents the preliminary results from two self-evaluation strategies, offering insights from different perspectives for future improvements. If real-world expert reports become available as reference documents, this evaluation workflow could be used to incorporate the AI feedback into instruction tuning and further align LLM outputs with human-preferred reporting styles.

6. CONCLUSIONS

RAG has been widely used to mitigate the hallucination issue of LLM applications in domain-specific scenarios by leveraging an external knowledge base. However, project data in the industry is often highly confidential, and the capabilities of available LLMs without RAG for civil engineering tasks remain underexplored, causing high risks of using them as-is. To address this challenge, local LLMs based on open-weight models present a promising alternative for safely incorporating RAG. Despite the relatively lower performance of open-weight models than commercial ones, multi-agent systems using collaboration-based communication protocols have demonstrated the ability to enhance generation quality to a comparable level. This paper implements two approaches: RAG-SAS and MAS, both utilizing the *Llama-3.2-11B-Vision-Instruct* model. Their outputs are evaluated by GPT-4o against pseudo-ground truths generated from itself. In the experiments, 100 images across five civil engineering tasks were collected: bridge inspection, building inspection, indoor construction progress reporting, outdoor construction progress reporting, and built-environment facility management.

The results showed that with the aid of retrievals, RAG-SAS consistently outperforms MAS in progress reporting and facility management while both performed comparably in defect inspection tasks, with the base model excelling in visual grounding but facing challenges in alignment. For the effective deployment of local LLMs using smaller model variants, both RAG and the MAS communication protocol require adaptation and refinement to account for the model capability limitations. The correctness of pseudo ground truths also needs to be analyzed. The validation dataset contains only a limited number of images because supervising superintelligent LLMs requires increasingly more labor hours to ensure accuracy and factual correctness. It is crucial to establish a reliable performance before integrating these agents into inspection reporting workflows as incorrect judgements could raise safety concerns and cause confusion. Although self-evaluation methods offer a low-cost solution for performance analyze, they remain under-explored. Future research should emphasize more on clearly defining evaluation metrics and core values for assisting humans with domain-specific tasks. The dataset can be expanded to cover various inspection reporting workflows with different focuses. This paper serves as a pioneering effort in evaluating the feasibility of incorporating LLMs in such workflows. All the results and datasets mentioned in this paper are available in the shared link: (https://drive.google.com/drive/folders/1U4sncLvA_pI5HgqQ2eSMWby9BtFv7NCJ?usp=sharing).

REFERENCES

Amer, F., Jung, Y., and Golparvar-Fard, M. 2023. Construction schedule augmentation with implicit dependency constraints and automated generation of lookahead plan revisions. *Automation in Construction*, *152*, 104896.

Autodesk University. 2024. Optimizing Revit Structural Intelligent Models with Large Language Models and Autodesk Platform Services. [online] Available at: https://www.autodesk.com/autodesk-university/class/Optimizing-Revit-Structural-Intelligent-Models-with-Large-Language-Models-and-Autodesk-Platform-Services-2024 [Accessed 24 February 2025].

Chen, G., Alsharef, A., Ovid, A., Albert, A., and Jaselskis, E. 2025. Meet2Mitigate: An LLM-powered framework for real-time issue identification and mitigation from construction meeting discourse. *Advanced Engineering Informatics*, *64*, 103068.

Du, C., Esser, S., Nousias, S., and Borrmann, A. 2024. Text2BIM: Generating Building Models Using a Large Language Model-based Multi-Agent Framework. *arXiv preprint arXiv:2408.08054*.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., ..., and He, Y. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N. V., Wiest, O., and Zhang, X. 2024. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. *arXiv preprint arXiv:2402.01680*.

He, C., Yu, B., Liu, M., Guo, L., Tian, L., and Huang, J. 2024. Utilizing large language models to illustrate constraints for construction planning. *Buildings*, *14*(8), 2511.

Hsu, S. H., Hung, H. T., Lin, Y. Q., and Chang, C. M. 2023. Defect inspection of indoor components in buildings using deep learning object detection and augmented reality. *Earthquake Engineering and Engineering Vibration*, *22*(1), 41-54.

Jung, Y., Cho, I., Hsu, S. H., and Golparvar-Fard, M. 2024. VisualSiteDiary: A detector-free Vision-Language Transformer model for captioning photologs for daily construction reporting and image retrievals. Automation in Construction, 165, 105483.

Jang, S., Lee, G., Oh, J., Lee, J., and Koo, B. 2024. Automated detailing of exterior walls using NADIA: Natural-language-based architectural detailing through interaction with AI. *Advanced Engineering Informatics*, *61*, 102532.

Kenton, Z., Siegel, N. Y., Kramár, J., Brown-Cohen, J., Albanie, S., Bulian, J., ..., and Shah, R. 2024. On scalable oversight with weak LLMs judging strong LLMs. arXiv. *Preprint posted online*.

Lee, H., Phatale, S., Mansoor, H., Mesnard, T., Ferret, J., Lu, K., ..., and Prakash, S. (2023). RLAIF vs. RLHF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. *arXiv preprint arXiv:2309.00267*.

Liu, H., Wang, G., Huang, T., He, P., Skitmore, M., and Luo, X. 2020. Manifesting construction activity scenes via image captioning. *Automation in Construction*, *119*, 103334.

McAleese, N., Pokorny, R. M., Uribe, J. F. C., Nitishinskaya, E., Trebacz, M., and Leike, J. 2024. LLM Critics Help Catch LLM Bugs. *arXiv preprint arXiv:2407.00215*.

Meta AI. 2024. *Llama-3.2-11B-Vision-Instruct*. https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct

Núñez-Morales, J. D., Hsu, S. H., Ibrahim, A., & Golparvar-Fard, M. (2023). Reality-Enhanced Synthetic Image Training Dataset for Computer Vision Construction Monitoring. *Proceedings of International Structural Engineering and Construction*, *10*(1), CON-29.

Pu, H., Yang, X., Li, J. and Guo, R, 2024. AutoRepo: A general framework for multimodal LLM-based automated construction reporting. *Expert Systems with Applications*, *255*, p.124601.

Xiao, B., Wang, Y., and Kang, S. C. 2022. Deep learning image captioning in construction management: a feasibility study. *Journal of Construction Engineering and Management*, *148*(7), 04022049.

Zhai, P., Wang, J., and Zhang, L. 2023. Extracting worker unsafe behaviors from construction images using image captioning with deep learning–based attention mechanism. *Journal of Construction Engineering and Management*, *149*(2), 04022164.

Zhou, J., Liu, Z., Xiao, S., Zhao, B., and Xiong, Y. 2024. VISTA: visualized text embedding for universal multi-modal retrieval. *arXiv preprint arXiv:2406.04292*.