MARC-V: Multi-Agent Rendering for CAD-VQA

Ken Miyamoto[1] and Shotaro Miwa[2]

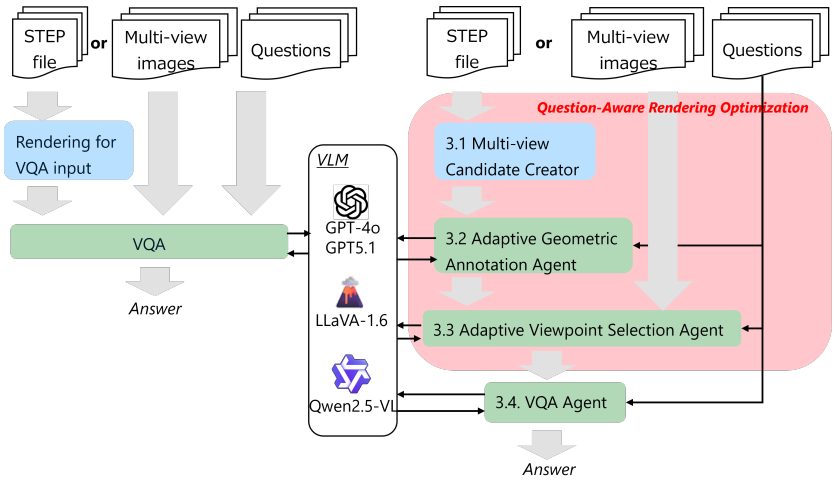
Abstract. Although visual question answering for CAD models (CAD-VQA) serves a wide range of applications, including in-progress design review and manufacturing feature recognition, the existing approaches face two key limitations: reliance on images rendered from fixed, predefined viewpoints and the lack of informative geometric annotations in these rendered images. Fixed viewpoints are not designed for arbitrary questions, and may not satisfy the requirements of a given question. To address this issue, we introduce a question-conditioned adaptive viewpoint selection agent, inspired by CAD design review workflows in which designers carefully select viewpoints for targeted evaluation. However, viewpoint selection by itself is not sufficient to accurately answer questions about part shapes and dimensions, because rendered images do not provide explicit geometric measurements. To address this limitation, we augment our approach with an adaptive geometric annotation agent that overlays question-relevant measurements on rendered images in a question-conditioned manner. To accommodate multiple input modalities—namely CAD models and multi-view rendered images—our method is implemented as a composition of specialized agents. Extensive experiments across multiple vision–language models demonstrate consistent performance improvements on both a CAD model dataset and a multi-view rendered image dataset.

Keywords: Computer Aided Design (CAD) · Vision-Language Model (VLM) · Visual Question Answering (VQA)

1 Introduction

Computer Aided Design (CAD) is widely used for civil engineering and manufacturing. To support CAD designers, two approaches have been proposed: CAD-Generation and CAD-based Visual Question Answering (CAD-VQA). CAD-Generation primarily supports the model-creation phase by generating parametric operation sequences or final geometry from textual, visual, or point-cloud conditions.

CAD-VQA [5,6] aims to answer questions about CAD models or their rendered views. Typical examples include manufacturing-oriented counting questions such as "How many circular holes are present on the target part?" as well as visual reasoning questions over rendered assemblies such as "Which component would be directly connected to the rotary actuator's shaft?" These tasks arise in design review, feature recognition, and post-hoc inspection/documentation,



(a) Existing methods [2,5] (b) MARC-V (ours)

Fig. 1. Architectural comparison between existing methods [2,5] and our MARC-V.

where a user often needs question-specific visual evidence rather than a fixed set of generic views.

With recent advances in Vision–Language Models (VLMs) [13,8,19], most existing CAD-VQA approaches are built upon VLMs. Due to the input modality constraints of VLMs, existing methods commonly employ multi-view images rendered from a set of fixed, predefined viewpoints.

However, such rendered images present three major limitations. First, the use of fixed viewpoints may not be suitable due to occlusions or limited visibility. To address this issue, we introduce a question-conditioned adaptive viewpoint selection agent, inspired by CAD design review workflows in which designers select viewpoints tailored to specific evaluation aspects.

Second, existing CAD-VQA approaches typically lack explicit geometric measurements, which further hinders the accurate identification and interpretation of part shapes and sizes within CAD models. To address this limitation, our adaptive geometric annotation agent overlays geometric measurements onto a rendered image, thereby enabling improved reasoning over CAD models.

Third, most existing CAD-VQA approaches are tailored to either CAD models or rendered images, but not both. Multi-modal input capability extends applicability across different stages of the product lifecycle, with CAD models primarily used during design and rendered images serving as the dominant representation during real-world adoption. Therefore, our method supports both data modalities by composing multiple specialized agents. This modular design enables integration into multi-agent frameworks such as LangGraph [7].

Fig. 1 shows the architectures existing methods and our proposed method. The key component of our method is the question-aware rendering optimiza-

tion, which elaborately chooses viewpoints and annotations to enhance VQA performance.

The contributions of this paper are the following.

- **Rendering Optimization for CAD-VQA** We jointly optimize rendering by combining adaptive viewpoint selection with adaptive geometric annotation, producing question-aware visual inputs that yield state-of-the-art performance.
- **Modular Multi-modal Input Design** By composing multiple specialized agents, our method supports both CAD models and multi-view images as inputs. This modular architecture is compatible with existing multi-agent frameworks.

2 Related Works

2.1 CAD Design Support

CAD-Generation methods focus on synthesizing parametric CAD models from multi-modal inputs. For example, DeepCAD [16] represents CAD models as sequences of parametric operations, while CAD-MLLM [17] generates CAD models directly from inputs such as point clouds, images, and text.

This paper focuses on CAD-VQA, as it involves a broader scope than CAD-Generation, such as design review and manufacturing-feature recognition.
CAD-VQA formulates question answering tasks over CAD models using VLMs [13,19,9,10]. Owing to the input restrictions of VLMs, CAD models are typically rendered into multi-view images prior to inference. VLM CAD Feature Recognition (VCFR) [5] performs VQA on such multi-view rendered images with the aid of few-shot examples. CAD Query [6] incorporates part-level segmentation before executing the VQA process. CAD-Image-VQA [2] uses rendered images, where multiple parts are integrated. A primary part and the surrounding parts are rendered by red and transparent, respectively.

These rendering-based approaches [5,6,2] suffer from two major limitations: fixed, predefined viewpoints and the absence of explicit geometric cues. This paper addresses both limitations in existing CAD-VQA methods [5,6,2].

2.2 3D Visual Understanding

Viewpoint is a critical factor in vision-based tasks. Multi-view convolutional neural network (MVCNN) [12] enhances object detection by selecting an optimal viewpoint, where a max-pooling layer aggregates predictions and retains the most confident result across views. RotationNet [4] mitigates the winner-takes-all strategy of MVCNN [12] . Detection results from multiple viewpoints are merged with learned priorities. ViewActive [15] iteratively updates a viewpoint to obtain high-confidence prediction. Although these studies demonstrate the importance of viewpoint selection, they do not consider question-dependent reasoning.

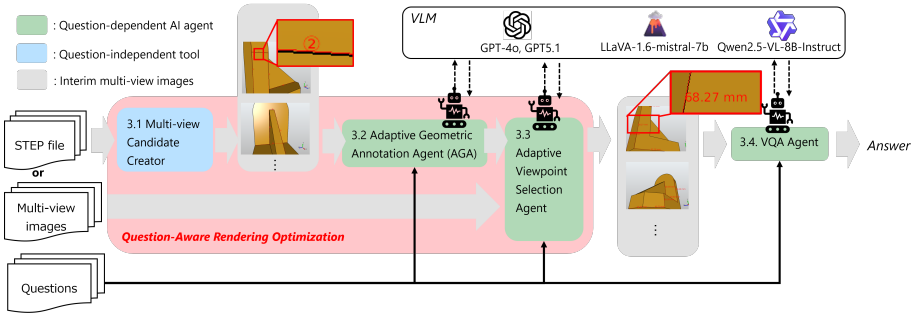


Fig. 2. Multi-Agentic CAD-VQA(MARC-V) pipeline. Both AGA and adaptive viewpoint selection agent are question-aware prompt-based procedures to optimize rendered images.

Visual Prompt provides a mechanism for embedding textual or symbolic information directly into images. For example, Set-of-marks [20] encodes numerical values to enhance detection performance, and GPT4Scene [11] establishes correspondences across views using embedded markers. These works suggest that overlaying visual markers can significantly enhance 3D scene understanding, and a similar strategy may also be effective for CAD models.

3D Foundation Models operate on various 3D representations, including point clouds and multi-view images. PointCLIPv2 [21] projects point clouds onto an image plane, and the resulting projected images are subsequently processed by a VLM. To unify multi-modal data within a shared embedding space, ULIP2 [18] learns an adaptor that aligns the embeddings of point clouds, images, and text.

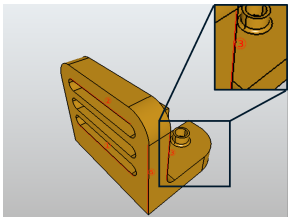
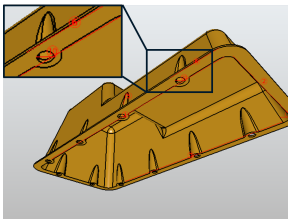
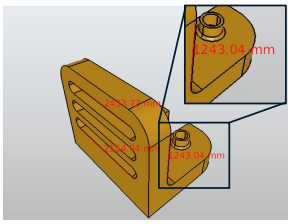
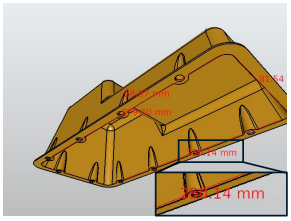
Although the original CAD model may be available in some settings, directly operating on raw 3D CAD representations requires a different model class than the image-centric VLMs used in current CAD-VQA practice. Our goal in this paper is therefore not to replace CAD-native reasoning with a new 3D foundation model, but to improve the rendered evidence presented to strong general-purpose VLMs.

3 Multi-Agent Rendering for CAD-VQA

To enhance CAD-VQA through adaptive viewpoint selection and informative geometric annotations, we propose *Multi-Agent Rendering for CAD-VQA* (MARC-V). MARC-V does not introduce additional task-specific training for either the adaptive geometric annotation agent (AGA) or the adaptive viewpoint selection agent. The geometric primitive detectors are off-the-shelf components, and both AGA and viewpoint selection are implemented as prompt-based selection procedures over candidate primitives and candidate views, respectively.

Furthermore, to accommodate both image and CAD-model input modalities, our MARC-V bypasses AGA when the input is provided as an image. An overview of the MARC-V pipeline is illustrated in Fig. 2.

Table 1. The resultant images of subsection 3.1 and 3.2.

	S1	S2
3.1 Multi-view candidate creator outputs in $\mathcal{I}$		
3.2 Adaptive geometric annotation agent outputs in $\mathcal{I}'$		

3.1 Multi-view Candidate Creator

We render the CAD model from multiple viewpoints and augment each image with indexed markers corresponding to detected geometric primitives (lines, circles, and ellipses). Primitive detection is performed using non-VLM methods [14,1], which are more efficient and reliable for this task.

Line segments are detected using LSD [14], yielding $L_i = \{[\mathbf{s}_{i,j}, \mathbf{e}_{i,j}] \mid j = 1, \ldots, N_i\}$, where $\mathbf{s}_{i,j}$ and $\mathbf{e}_{i,j}$ denote the endpoints of the j-th line in view i. Ellipses and circles are detected using Ellipse R-CNN [3], producing $E_i = \{(\mathbf{c}_{i,k}, \phi_{i,k}, a_{i,k}, b_{i,k}) \mid k = 1, \ldots, M_i\}$, where $\mathbf{c}_{i,k}$, $\phi_{i,k}$, and $(a_{i,k}, b_{i,k})$ denote the center, rotation, and semi-axis lengths, respectively. A circle corresponds to the special case $a_{i,k} = b_{i,k}$. Since Ellipse R-CNN is trained on images captured from diverse viewpoints, it provides more robust detection compared to analytical methods [1].

The detected primitives are overlaid on each image with indexed markers placed at line midpoints and ellipse centers, producing the candidate image set $\mathcal{I} = \{I_i\}$.

3.2 Adaptive Geometric Annotation Agent

The Adaptive Geometric Agent (AGA) selects question-relevant labels from detected primitives. Subsequently, the selected labels are converted into geometric measurements.

Let the set of detected primitives in the i-th view be $P_i = \{p_{i,1}, ..., p_{i,N_i+M_i}\}$, where each primitive is either a line segment or an ellipse/circle candidate. Rather than overlaying all detected primitives, AGA retains only those that are likely to be useful for answering a given question q. We assign each primitive

```
Detect the necessary labels to conduct the
following task. The labels are attached on
the input image.
<question_dependent_prompt>
<format_restriction_prompt>
```

(a) Adaptive geometric annotation agent prompt to obtain $\hat{P}_i$.

```
You are provided with input images
containing multiple views of a 3D CAD model
taken from different angles.
Your task is to identify and describe the
most optimal image to capture manufacturing
features in the input images.
  The guideline to select only one image is
the following.
<question_dependent_prompt>
<format_restriction_prompt>
```

(b) Adaptive viewpoint selection agent prompt to obtain S^*.

Fig. 3. Prompts designs. Both `<question_dependent_prompt>` and `<format_restriction_prompt>` depend on a question q. The concrete prompts are described in Fig. 6 and Fig. 7.

a question-conditioned relevance score $s_{rel}(p_{i,l}, q, I_i)$, which reflects whether the primitive is relevant to the queried attribute, sufficiently visible in the current projection, and suitable for geometric estimation. The retained primitive subset is then $\hat{P}_i = \{p_{i,l} \in P_i \mid s_{rel}(p_{i,l}, q, I_i) \geq \tau\}$ or equivalently the top-k primitives ranked by s_{rel}. A VLM automatically determines the threshold τ, and $\hat{P}_i$ is selected accordingly. The prompt to select the retained subset $\hat{P}_i$ is illustrated in Fig.3 (a).

For a line segment with endpoints $\mathbf{s}_{i,j}$ and $\mathbf{e}_{i,j}$, the recovered 3D length estimate is as follows:

$$\left\| \mathbf{r}\left(t_{\mathbf{s}_{i,j}}, \mathbf{v}_i, \mathbf{K}^{-1}\left[\mathbf{s}_{i,j}\ 1\right]^\top\right) - \mathbf{r}\left(t_{\mathbf{e}_{i,j}}, \mathbf{v}_i, \mathbf{K}^{-1}\left[\mathbf{e}_{i,j}\ 1\right]^\top\right) \right\| \quad p_{i,j}^{line} \in \hat{P}_i \tag{1}$$

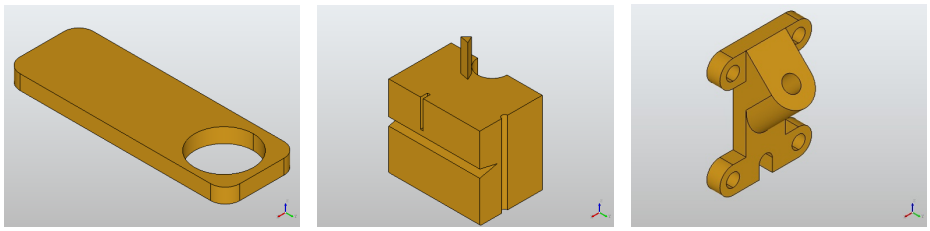
For an ellipse candidate parameterized by center $\mathbf{c}_{i,k}$, rotation $\phi_{i,k}$, and semi-axis lengths $(a_{i,k}, b_{i,k})$, we compute a diameter-like estimate using two opposite points on the fitted ellipse:

$$\left\| \mathbf{r}\left(t_{\mathbf{z}_{i,k}(0)}, \mathbf{v}_i, \mathbf{K}^{-1}\left[\mathbf{z}_{i,k}(0)\ 1\right]^\top\right) - \mathbf{r}\left(t_{\mathbf{z}_{i,k}(\pi)}, \mathbf{v}_i, \mathbf{K}^{-1}\left[\mathbf{z}_{i,k}(\pi)\ 1\right]^\top\right) \right\| \quad p_{i,k}^{ellipse} \in \hat{P}_i \tag{2}$$

where $\mathbf{K}$ is the camera intrinsic matrix, and $\mathbf{r}(t, \mathbf{v}, \mathbf{d}) = \mathbf{v} + t\mathbf{d}$ denotes ray-casting. $\mathbf{v}$ denotes the viewpoint position and $\mathbf{d}$ represents the ray direction from the viewpoint. t corresponds to the distance from the viewpoint to a point on an object along the ray. Ray-triangle intersection tests are performed to solve for t. $\mathbf{z}_{i,k}(\theta)$ denotes a point on a circle or an ellipse, as below:

$$\mathbf{z}_{i,k}(\theta) = \mathbf{c}_{i,k} + \begin{pmatrix} \cos\phi_{i,k} & -\sin\phi_{i,k} \\ \sin\phi_{i,k} & \cos\phi_{i,k} \end{pmatrix} \begin{pmatrix} a_{i,k}\cos\theta \\ b_{i,k}\sin\theta \end{pmatrix} \quad \theta \in [0, 2\pi) \tag{3}$$

Examples of the resultant images $I'_i \in \mathcal{I}'$ are shown in the last row of Table 1.



(a) Easy (b) Middle (c) Hard

Fig. 4. Sample images in VCFR [5].

To account for the reliability of image-based primitive detection in section 3.1, we employ two selections in AGA: question-relevant selection and ray-casted point existence selection. Moreover, even if a primitive mistakenly passes the above two selections, the corresponding label still presents actual size, because ray-casted points follow the CAD model.

3.3 Adaptive Viewpoint Selection Agent

Given the annotated candidate views, the adaptive viewpoint selection agent performs question-conditioned view ranking. Let the candidate rendered views be $V = \{v_1, ..., v_N\}$. For a given question q, a view utility score $u(v_i, q, I'_i)$ is assigned on each view v_i, which reflects three factors: (1) whether the view exposes question-relevant geometry, (2) how clearly the relevant primitives are visible, and (3) whether the view contributes evidence that is non-redundant with already selected views. The selected view subset is then defined as

$$S^*(q) = \arg \max_{S \subseteq V, v_i \in S} u(v_i, q, I'_i). \tag{4}$$

A VLM implicitly determines the view utility score u, from which $S^*(q)$ is derived. When CAD models are available, we partition the candidate views V by eight octants of $\mathbb{R}^3$; when only rendered images are available, we partition the candidate views V into six subsets randomly. The prompt to select views is shown in Fig.3 (b).

The subsequent VQA agent is the same as conventional approaches [2,5]. The agent answers on a given question q based on selected images S^*.

4 Evaluation

4.1 Dataset

VCFR [5] dataset consists of 100 CAD models in STEP format, categorized into three complexity levels. The dataset formulates CAD-VQA as a counting-based manufacturing feature recognition task and encompasses six question types that vary in prompting strategy and input modality. Following prior work, we evaluate

performance using accuracy and feature matching rate. While VCFR originally uses three fixed viewpoints, we additionally evaluate an eight-view setting for fair comparison with our adaptive rendering framework.

Sample images are shown in Fig. 4. While the easy and middle levels differ noticeably in terms of structural features, such as holes, bosses, and slots, the hard level does not exhibit significant structural differences from the middle level.

CAD-Image-VQA [2] dataset consists of rendered multi-view images arranged into tiled inputs, where each tiled input includes six views depicting a target CAD part and its surrounding parts. The surrounding parts are rendered transparently to emphasize the target CAD part. The task is formulated as multiple-choice visual question answering, and evaluation is performed using classification accuracy.

4.2 Implementation

VLMs We evaluate general performance using multiple VLMs, including GPT-4o [9], GPT-5.1 [10], LLaVA-1.6-Mistral-7B (LLaVA-1.6) [8], and Qwen2.5-VL-8B-Instruct (Qwen2.5-VL) [19]. GPT-4o is accessed via the Azure OpenAI API with version *2024-10-21*. GPT-5.1 is accessed through the OpenAI API with the reasoning effort set to *low*. For the open models LLaVA-1.6 and Qwen2.5-VL, the temperature is set to zero, resulting in deterministic outputs and a zero standard deviation. Unless otherwise specified, all experiments are conducted using the default inference settings of each model. Due to output variability in closed models, all reported results for GPT-4o and GPT-5.1 are averaged over five independent runs.

VCFR We use pythonOCC to render CAD models in VCFR dataset, following the conventional rendering pipeline adopted in VCFR method. The multi-view candidate creator generates images from 78 distinct viewpoints. The detail viewpoint setting is depicted in Fig. 8. While the original VCFR method uses only three viewpoints as inputs for a VLM, we also conduct VCFR method with eight viewpoints to ensure comprehensive coverage of the CAD model.

CAD-Image-VQA When employing MARC-V, only the adaptive viewpoint selection agent is employed, as the dataset provides only pre-rendered images. Specifically, six images are selected from the 72 rendered images in the CAD-Image-VQA dataset following the strategy described in subsection 3.3. The selected views are concatenated and downsampled to a resolution of 362×256, consistent with the preprocessing procedure used in the original CAD-Image-VQA method. The temperature of a VLM is adjusted to 0.01.

4.3 Result

VCFR The quantitative results are presented in Table 2. The predefined VQA prompts in VCFR method are used across all experiments; no test-time prompt

Table 2. Experiment on VCFR [5] dataset. *Easy*, *Middle*, and *Hard* represent structual complexity. *FMR* stands for Feature Matching Rate.

Category	VLM	Method	Views	Easy		Middle		Hard		Overall	
				Acc. (%)	FMR (%)	Acc. (%)	FMR (%)	Acc. (%)	FMR (%)	Acc. (%)	FMR (%)
Open	LLaVA-1.6	VCFR	3	50.43±0.00	83.58±0.00	22.32±0.00	73.55±0.00	43.57±0.00	**89.42±0.00**	32.71±0.00	81.51±0.00
		VCFR	8	**63.48±0.00**	74.63±0.00	**27.46±0.00**	61.16±0.00	42.32±0.00	73.08±0.00	37.06±0.00	68.49±0.00
		MARC-V	8	61.04±0.00	**88.06±0.00**	27.28±0.00	**77.19±0.00**	**45.89±0.00**	88.65±0.00	**37.56±0.00**	**83.56±0.00**
	Qwen2.5-VL	VCFR	3	**72.17±0.00**	67.16±0.00	52.68±0.00	69.42±0.00	65.15±0.00	**75.00±0.00**	59.20±0.00	70.89±0.00
		VCFR	8	70.43±0.00	68.66±0.00	53.57±0.00	68.60±0.00	63.90±0.00	70.19±0.00	59.08±0.00	69.18±0.00
		MARC-V	8	68.87±0.00	**72.84±0.00**	**73.66±0.00**	**75.04±0.00**	**65.98±0.00**	73.08±0.00	**70.27±0.00**	**73.29±0.00**
Closed	GPT-4o	VCFR	3	73.91±1.06	74.63±1.83	56.47±3.85	72.56±1.79	63.73±0.81	65.38±1.52	61.14±2.14	70.48±1.04
		VCFR	8	78.43±1.29	75.22±1.70	60.40±2.00	70.08±1.08	66.64±1.27	68.08±1.25	64.85±0.78	70.55±0.34
		MARC-V	8	**85.22±1.06**	**88.06±1.83**	**73.39±8.09**	**78.68±1.08**	**79.00±1.43**	**84.42±1.25**	**76.77±4.61**	**82.88±0.42**
	GPT-5.1	VCFR	3	78.61±2.09	84.48±2.26	**89.20±2.39**	83.97±0.74	82.41±2.23	84.81±1.85	85.65±1.56	84.38±0.89
		VCFR	8	83.30±1.56	85.07±2.79	74.96±2.26	81.82±1.31	**86.47±1.96**	88.27±1.25	79.60±1.51	84.86±1.12
		MARC-V	8	**84.70±4.11**	**85.67±2.91**	86.96±1.08	**86.94±1.48**	85.64±1.39	**88.85±1.99**	**86.24±0.72**	**87.33±1.39**

Table 3. Ablation study on VCFR [5] dataset. Overall performance is presented.

Category	VLM	Method	Views	Acc. (%) ↑	FMR (%) ↑
Open	LLaVA-1.6	VCFR [5]	8	37.06±0.00	68.49±0.00
		MARC-V (w/o AGA)		**37.81±0.00**	**85.27±0.00**
		MARC-V (w/ AGA)		37.56±0.00	83.56±0.00
	Qwen2.5-VL	VCFR [5]	8	59.08±0.00	69.18±0.00
		MARC-V (w/o AGA)		67.54±0.00	68.84±0.00
		MARC-V (w/ AGA)		**70.27±0.00**	**73.29±0.00**
Closed	GPT-4o	VCFR [5]	8	64.85±0.78	70.55±0.34
		MARC-V (w/o AGA)		69.28±2.23	74.79±1.62
		MARC-V (w/ AGA)		**76.77±4.61**	**82.88±0.42**
	GPT-5.1	VCFR [5]	8	79.60±1.51	84.86±1.12
		MARC-V (w/o AGA)		82.79±1.95	85.00±0.74
		MARC-V (w/ AGA)		**86.24±6.01**	**87.33±1.65**

adaptation is applied. The results demonstrate that MARC-V achieves performance improvements in most evaluation settings. Notably, more than 10% improvements are observed on GPT-4o and Qwen2.5-VL.

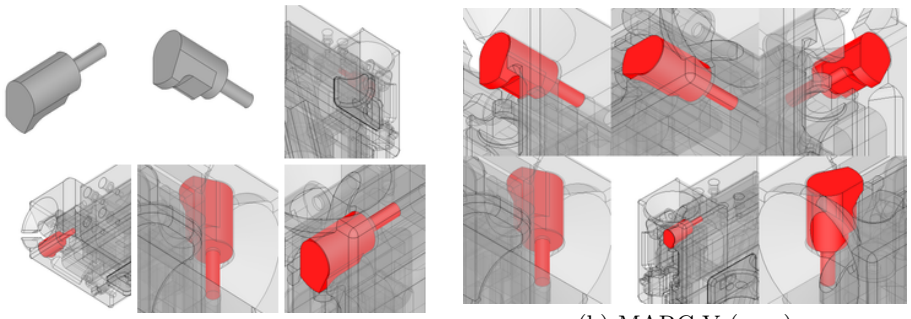
The sources of performance improvement vary across complexity levels. In the easy category, structural features are sparse, and question-relevant information is often hidden by the limited three viewpoints. Consequently, eight views lead to better performance than three views. In contrast, the middle and hard categories contain denser structural features, where question-aware rendering—particularly adaptive viewpoint selection and geometric annotation—more effectively supports model understanding. This benefit is especially found for stronger VLMs such as Qwen2.5-VL and GPT-4o.

As shown in Table 3, AGA improves performance for GPT-4o, GPT-5.1, and Qwen2.5-VL, suggesting that these models can make effective use of added geometric overlays as question-relevant cues. In contrast, LLaVA-1.6 shows slight degradation. This difference is consistent with prior findings [5] that VLMs vary in their robustness to dense scenes, visualized text, and fine-grained spatial

Table 4. Experiment on CAD-Image-VQA [2] dataset.

Category	VLM	Method	Views	Acc. (%) ↑
Open	Qwen2.5-VL	CAD-Image-VQA [2]	6	57.65±0.00
		MARC-V (ours)		**60.94±2.41**
Closed	GPT-4o	CAD-Image-VQA [2]	6	44.23±2.14
		MARC-V (ours)		**44.47±6.52**
	GPT-5.1	CAD-Image-VQA [2]	6	58.82±1.18
		MARC-V (ours)		**60.71±1.97**

Question: Based on the 2D image, which of the following components would you expect to see directly connected to the rotary actuator's shaft in an assembled system?



(a) CAD-Image-VQA [2] (b) MARC-V (ours)

Fig. 5. Selected multi-view images in CAD-Image-VQA [2] experiment. (a) uses predefined multi-view images. (b) uses the adaptive viewpoint selection agent in subsection 3.3.

grounding. We therefore hypothesize that AGA is beneficial when a model can reliably associate overlaid markers and measurements with the queried feature, but may be detrimental when the added annotations act as visual interference.

CAD-Image-VQA The quantitative results on the CAD-Image-VQA dataset are summarized in Table 4. Even without AGA, MARC-V outperforms the CAD-Image-VQA baseline on all the cases. This improvement is primarily attributed to the effectiveness of the proposed view selection strategy. As illustrated in Fig. 5, MARC-V selects more question-relevant views. Such targeted view selection leads to higher accuracy in the VQA task. We could not evaluate the results on LLaVA-1.6, due to its hallucinated responses. We observe a discrepancy between our rerun GPT-4o result in Table 4 and the result reported in the original paper [2]. Though the original paper reports an accuracy of 54.11 with GPT-4o, our rerun result achieves 44.23. We attribute this discrepancy to differences in the deployment environment: the original paper uses OpenAI GPT-4o, whereas our experiments are conducted using Azure OpenAI.

5 Limitation

Our current evaluation should be interpreted as an initial validation on two complementary CAD-VQA settings rather than as a final large-scale robustness study. In addition, the geometric annotations in MARC-V rely on the quality of projected primitive detection and ray-based recovery from rendered views. Under severe occlusion, truncation, overlapping geometry, or highly ambiguous projections, the recovered annotations may become unreliable. We will expand the evaluation with finer-grained analysis and qualitative failure cases, and we view the construction of a larger benchmark with question-level view relevance and primitive relevance annotations as an important direction for future work.

6 Conclusion

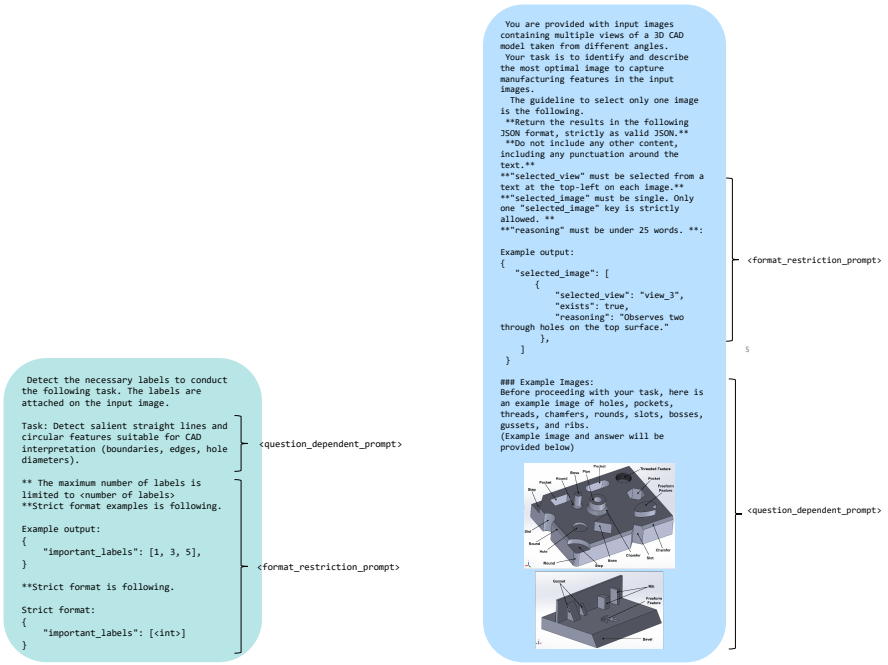
We propose MARC-V, a multi-agent framework for question-aware rendering in CAD-VQA. Experiments show consistent performance gains across most settings, highlighting the importance of question-aware rendered images for 3D-VQA. Moreover, MARC-V is compatible with modular multi-agent frameworks, which enables its application to both CAD model datasets and rendered image datasets.

Acknowledgement We gratefully acknowledge the support and resources provided by our organizations, which enabled the completion of this research.

References

1. Akinlar, C., Topal, C.: Edcircles: A real-time circle detector with a false detection control. Pattern Recognition **46**(3), 725–740 (2013)
2. Asgari, S., Lambourne, J.G., Mongkhounsavath, A.: How to determine the preferred image distribution of a black-box vision-language model? In: Neurips Safe Generative AI Workshop 2024 (2024)
3. Dong, W., Roy, P., Peng, C., Isler, V.: Ellipse r-cnn: Learning to infer elliptical object from clustering and occlusion. IEEE Transactions on Image Processing **30**, 2193–2206 (2021)
4. Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR). pp. 5010–5019 (2018)
5. Khan, M.T., Chen, L., Ng, Y.H., Feng, W., Tan, N.Y.J., Moon, S.K.: Leveraging vision-language models for manufacturing feature recognition in computer-aided designs. Journal of Computing and Information Science in Engineering **25**, 104501 (2025). https://doi.org/10.1115/1.4069266, https://doi.org/10.1115/1.4069266
6. Kienle, C., Alt, B., Katic, D., Jäkel, R., Peters, J.: Querycad: Grounded question answering for cad models. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 5798–5805. IEEE (2025)
7. LangChain: Langgraph (2024), https://github.com/langchain-ai/langgraph, accessed: 2024

8. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition(CVPR). pp. 26296–26306 (2024)
9. OpenAI: Gpt-4o (2024), https://platform.openai.com/docs/models/gpt-4o, accessed: 2025
10. OpenAI: Gpt-5.1 (2025), https://openai.com/index/gpt-5-1/, accessed: 2025
11. Qi, Z., Zhang, Z., Fang, Y., Wang, J., Zhao, H.: Gpt4scene: Understand 3d scenes from videos with vision-language models. arXiv preprint arXiv:2501.01428 (2025)
12. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 945–953 (2015)
13. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
14. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. IEEE transactions on pattern analysis and machine intelligence **32**(4), 722–732 (2008)
15. Wu, J., Lin, X., He, B., Fermuller, C., Aloimonos, Y.: Viewactive: Active viewpoint optimization from a single image. arXiv preprint arXiv:2409.09997 (2024)
16. Wu, R., Xiao, C., Zheng, C.: Deepcad: A deep generative network for computer-aided design models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6772–6782 (2021)
17. Xu, J., Wang, C., Zhao, Z., Liu, W., Ma, Y., Gao, S.: Cad-mllm: Unifying multimodality-conditioned cad generation with mllm. arXiv preprint arXiv:2411.04954 (2024)
18. Xue, L., Yu, N., Zhang, S., Panagopoulou, A., Li, J., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., et al.: Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 27091–27101 (2024)
19. Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al.: Qwen2.5 technical report. arXiv preprint arXiv:2505.09388 (2025)
20. Yang, J., Zhang, H., Li, F., Zou, X., Li, C., Gao, J.: Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v (2023), https://arxiv.org/abs/2310.11441
21. Zhu, X., Zhang, R., He, B., Guo, Z., Zeng, Z., Qin, Z., Zhang, S., Gao, P.: Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 2639–2650 (2023)



(a) Concrete prompt for adaptive label annotation

(b) Concrete prompt for adaptive viewpoint selection

Fig. 6. The prompts for the adaptive geometric annotation agent and the adaptive viewpoint selection agent, respectively. `<number_of_labels>` is a variable for the maximum number of detecting labels. `<question_dependent_prompt>` and `<format_restriction_prompt>` are placeholders illustrated in Fig. 3.

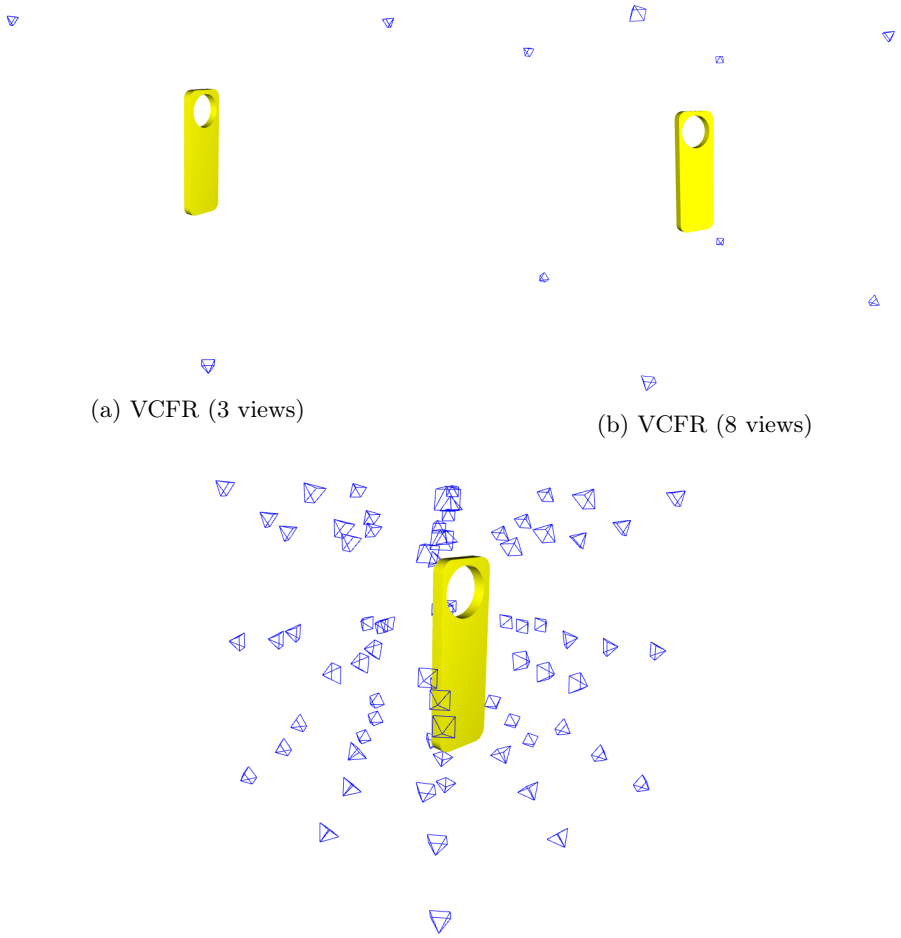
```
 You are provided with input images
containing multiple views of a 3D CAD
model taken from different angles.
 Your task is to identify and describe
the most optimal image to capture
manufacturing features in the input
images.
  The guideline to select only one image
is the following.
The two images are selected by the
strongest relationships with the
following question.
<question>
**Return the results in the following
JSON format, strictly as valid JSON.**
**Do not include any other content,
including any punctuation around the text.
**
**"selected_view" must be selected from a
text at the top-left on each image.**
**"selected_image" must be single. Only
one "selected_image" key is strictly
allowed. **
**"reasoning" must be under 25 words. **:

Example output:
{
   "selected_image": [
       {
           "selected_view": "view_3",
           "exists": true,
           "reasoning": "Observes two
through holes on the top surface."
       },
       {
           "selected_view": "view_5",
           "exists": true,
           "reasoning": "The view is
related with the question."
       }
   ]
}
```

<question_dependent_prompt>

<format_restriction_prompt>

Fig. 7. The prompt for the adaptive viewpoint selection agent on CAD-Image-VQA dataset. `<question>` is a variable to input a given question. `<question_dependent_prompt>` and `<format_restriction_prompt>` are placeholders illustrated in Fig. 3.



(a) VCFR (3 views)

(b) VCFR (8 views)

(c) Multi-view candidate creator of MARC-V (78 views)

Fig. 8. Dedicated viewpoint configurations used in the VCFR dataset experiments. Each blue tetrahedron indicates a camera viewpoint.