ASME Journal of Computing and Information Science in Engineering
Online journal at:
https://asmedigitalcollection.asme.org/computingengineering

Leveraging Vision-Language Models for Manufacturing Feature Recognition in Computer-Aided Designs

Muhammad Tayyab Khan
Singapore Institute of Manufacturing Technology (SIMTech),
Agency for Science, Technology and Research (A*STAR),
5 CleanTech Loop, #01-01 CleanTech Two Block B,
Singapore 636732;
School of Mechanical and Aerospace Engineering,
Nanyang Technological University,
Singapore 639798
e-mail: khan0022@e.ntu.edu.sg

Lequn Chen[1]
Advanced Remanufacturing and Technology Centre (ARTC),
Agency for Science, Technology and Research (A*STAR),
3 CleanTech Loop, #01-01 CleanTech Two,
Singapore 637143
e-mail: chen1470@e.ntu.edu.sg

Ye Han Ng
Advanced Remanufacturing and Technology Centre (ARTC),
Agency for Science, Technology and Research (A*STAR),
3 CleanTech Loop, #01-01 CleanTech Two,
Singapore 637143
e-mail: ng_ye_han@artc.a-star.edu.sg

Wenhe Feng
Singapore Institute of Manufacturing Technology (SIMTech),
Agency for Science, Technology and Research (A*STAR),
5 CleanTech Loop, #01-01 CleanTech Two Block B,
Singapore 636732
e-mail: fengwh@simtech.a-star.edu.sg

Nicholas Yew Jin Tan
Singapore Institute of Manufacturing Technology (SIMTech),
Agency for Science, Technology and Research (A*STAR),
5 CleanTech Loop, #01-01 CleanTech Two Block B,
Singapore 636732
e-mail: nicholas_tan@simtech.a-star.edu.sg

Seung Ki Moon[1]
School of Mechanical and Aerospace Engineering,
Nanyang Technological University,
Singapore 639798
e-mail: skmoon@ntu.edu.sg

Automatic feature recognition (AFR) is essential for transforming design knowledge into actionable manufacturing information. Traditional AFR methods, which rely on predefined geometric rules and large datasets, are often time-consuming and lack generalizability across various manufacturing features. To address these challenges, this study investigates vision-language models (VLMs) for automating the recognition of a wide range of manufacturing features in computer-aided designs (CAD) without extensive training datasets or predefined rules. Instead, prompt engineering techniques, such as multiview query images, few-shot learning, sequential reasoning, and chain-of-thought, are applied to enable recognition. The approach is evaluated on the proposed CAD dataset containing designs of varying complexity relevant to machining, additive manufacturing, sheet metal forming, molding, and casting. Five VLMs, including three closed-source models (GPT-4o, Claude-3.5-Sonnet, and Claude-3.0-Opus) and two open-source models (LLava and MiniCPM), are evaluated on this dataset with ground truth features labeled by experts. Key metrics include feature quantity accuracy, feature name matching accuracy, hallucination rate, and mean absolute error (MAE). Results show that Claude-3.5-Sonnet achieves the highest feature quantity accuracy (74%) and name matching accuracy (75%) with the lowest MAE (3.2), while GPT-4o records the lowest hallucination rate (8%). In contrast, open-source models have higher hallucination rates (larger than 30%) and lower accuracies (less than 40%). This study demonstrates the potential of VLMs to automate feature recognition in CAD within diverse manufacturing scenarios. [DOI: 10.1115/1.4069266]

Keywords: automatic feature recognition, vision-language models, computer-aided design, prompt engineering, advanced manufacturing, computer-aided manufacturing, engineering informatics, knowledge engineering, manufacturing automation

1 Introduction

The integration of computer-aided design (CAD), computer-aided process planning (CAPP), and computer-aided manufacturing (CAM) is crucial for seamless operations in modern manufacturing. CAD enables the creation of digital designs that serve as the foundation for the manufacturing process [1]. CAPP acts as the bridge between CAD and CAM by analyzing the necessary manufacturing processes and conditions [2]. Subsequently, CAM uses the insights from CAPP to control and automate manufacturing equipment, transforming a CAD model into a physical product [3]. This integration not only facilitates a smooth transition from digital blueprints to physical products but also significantly improves manufacturing efficiency.

Within the integrated workflow of CAD, CAPP, and CAM, automatic feature recognition (AFR) plays a pivotal role in CAPP. AFR converts CAD models into recognizable manufacturing features that CAM systems can utilize [3]. The growing complexity of CAD models, which include features related to various manufacturing

[1]Corresponding authors.

Manuscript received October 10, 2024; final manuscript received July 15, 2025; published online August 18, 2025. Assoc. Editor: Jianxi Luo.

processes such as machining, additive manufacturing, casting, and molding, highlights the necessity of AFR. Traditional manual methods for identifying these features are often time-consuming and prone to errors [4]. By automating this process, AFR reduces the reliance on human intervention, thereby minimizing potential variability.

AFR has a substantial impact on downstream tasks by improving the accuracy of manufacturing time and material requirement calculations, both of which are crucial for cost estimation [5]. In terms of machine tool selection, AFR helps in selecting suitable tools and fixtures. Furthermore, it can be used to optimize tool paths and reduce manufacturing time by identifying efficient strategies that minimize tool changes and repositioning. These capabilities establish AFR as a vital area of research in manufacturing automation [6].

Despite their advantages, traditional AFR methods have significant limitations. Most existing AFR methods [7–15] are primarily focused on machining features and do not adequately address features associated with other manufacturing processes, such as additive manufacturing, sheet metal forming, casting, and molding. This limitation arises from the challenge of adapting predefined geometric patterns to recognize the diverse and complex features present in various manufacturing processes. Furthermore, traditional methods rely heavily on expert knowledge to develop and maintain feature recognition rules [11,16], which can hinder scalability. The datasets used also often lack diversity, making it difficult to manage the wide range of features encountered in real-world manufacturing scenarios. Therefore, advancing feature recognition methods to comprehensively identify features across diverse manufacturing processes is crucial for optimizing the overall manufacturing workflow.

To overcome these limitations, vision-language models (VLMs) [17–19] offer a promising solution for AFR by leveraging their ability to understand and interpret complex visual and textual information. The main contribution of this study is a novel framework that leverages large language model (LLM) for AFR in manufacturing by recognizing complex CAD features using prompt engineering. Additionally, the study aims to develop a customized CAD dataset with varying feature complexities, systematically apply prompt engineering techniques, and propose tailored evaluation metrics to objectively assess model performance in AFR tasks. This approach reduces reliance on predefined geometric patterns and enhances the adaptability of AFR to manage the growing complexity of modern CAD.

To explore the potential of VLMs in manufacturing feature recognition and address the challenges posed by current AFR methods, this study aims to answer the following research questions:

(1) *How effectively can VLMs recognize features in CAD across various manufacturing processes?*
(2) *What prompt engineering techniques can be implemented to optimize VLMs for AFR tasks?*
(3) *What metrics and methodologies can be used to effectively quantify and evaluate the performance of VLMs in AFR?*

The objective of this work is to automate the recognition of a wide range of manufacturing features in CAD using VLMs. To achieve this, a diverse dataset of 100 CAD is developed and categorized into easy, medium, and hard levels based on feature complexity, feature quantity, and expert judgment. The study employs four prompt engineering techniques, such as multiview query images [20], few-shot learning [21], sequential reasoning [22], and chain-of-thought (CoT) reasoning [23], to enhance VLM performance in AFR tasks.

Five state-of-the-art (SOTA) VLMs are evaluated, including three closed-source models (*GPT-4o* [24], *Claude-3.5-Sonnet* [25], and *Claude-3.0-Opus* [26]) and two open-source models (*MiniCPM-Llama3-V2.5* [27] and *Llava-v1.6-mistral-7b* [28]). These models are tested against a ground truth CAD dataset labeled by domain experts. Their effectiveness is assessed using four key evaluation metrics: feature quantity accuracy, feature name matching accuracy, hallucination rate, and mean absolute error. These metrics provide a robust framework for evaluating the capabilities of VLMs to accurately identify manufacturing features across different levels of complexity.

The rest of the article is organized as follows: Sec. 2 reviews related literature and outlines the challenges in AFR. Section 3 explains the methodology, including the creation of the CAD dataset and the application of prompt engineering techniques. Section 4 presents the results and discusses the performance of the VLMs. Finally, Sec. 5 provides the conclusions, discusses the study's limitations, and suggests directions for future research.

2 Related Work

In this section, we review the literature on AFR and identify the challenges associated with its implementation. We also examine the application of VLMs in AFR and discuss the motivations driving this study.

2.1 Challenges in Automatic Feature Recognition. In CAM, features provide a higher-level abstraction of geometric shapes, which is essential for automating various computer-aided engineering activities [29]. Feature recognition involves identifying these features to seamlessly integrate design with downstream applications such as engineering analysis [30,31], reverse engineering [32], optimization [33], design validation [34], manufacturing planning [35,36], and design for excellence (DfX) [37]. However, CAD models often lack the high-level geometric and topological descriptions necessary for these tasks [38]. Consequently, feature recognition plays a key role in translating low-level geometric entities from CAD models into meaningful features with attributes that reflect both design intent and manufacturing functions [29]. Despite these efforts, effectively recognizing complex manufacturing features remains a significant challenge.

Over the past four decades, researchers have developed a wide range of AFR techniques, broadly classified into rule-based and learning-based approaches [6–8,16,39]. Early efforts focused on rule-based approaches, which rely on expert-defined rules to identify machining features based on their geometric and topological properties [16,40]. Rule-based techniques include logic rules and expert systems, volume decomposition, hint-based methods, graph-based methods, and hybrid systems [11,29]. Logic rules utilize expert-defined rules to recognize features based on specific geometric patterns, offering straightforward implementation but lacking in scalability and flexibility [16,41,42]. Graph-based methods model CAD using nodes and arcs to represent faces and edges, enhancing feature recognition performance but facing challenges with topology variations and high computational demands [43–47]. Hint-based approaches use retained data from machining features to infer types yet struggle with the flexibility needed for complex features [48,49]. Volume decomposition, which divides a CAD model into smaller volumes for feature assignment, faces similar issues with computational intensity [50–52]. Despite the integration of these methods into hybrid approaches to address complex feature recognition, they continue to confront significant challenges in managing intersecting features and computational efficiency, highlighting the need for more adaptable solutions in AFR [53].

In recent years, learning-based methods reduce reliance on extensive design rules through machine learning and deep learning techniques [7,9]. These approaches are categorized into artificial neural network (ANN)-based and convolutional neural network (CNN)-based methods, which convert machining features into low-level representations using extensive training datasets [11]. ANNs utilize face adjacency matrices or vectors that represent geometry and topology information as inputs [54,55]. However, these methods face challenges such as dependency on complex vector design and the loss of crucial part details, which reduce recognition performance and limit the range of identifiable features [8,29]. To

overcome these limitations, CNNs have been introduced with a two-step process: first, features are segmented from part models using algorithms like watershed or hint-based techniques; then, they are identified using 2D or 3D CNNs [56–58]. Recent advancements have simplified this to a one-step approach, using 2D images from various orientations to represent 3D models and predict cuboid bounding boxes that locate and identify features [59]. However, this method still struggles with challenges such as loss of geometry and topology information and inaccuracies in feature segmentation, impacting crucial downstream process planning tasks [11].

The limitations of traditional AFR methods highlight the need for more adaptable solutions. VLMs offer a promising alternative by combining computer vision with natural language processing (NLP) to interpret complex visual and textual information [18]. Unlike rule-based AFR methods that rely on predefined geometric rules, VLMs are flexible enough to recognize a variety of intricate manufacturing features. Furthermore, VLMs differ from learning-based methods, which typically require extensive training datasets, as they can effectively identify features through prompt engineering techniques [17,22]. This strategy enables precise VLM guidance without large-scale data requirements. While traditional AFR methods often focus on specific stock geometries like cuboids [7,10,11] or cylinders [60,61], VLMs handle a broader range of stock types and features in input images. Their capability to process both visual and textual data allows them to recognize diverse features in CAD, overcoming the scalability challenges faced by existing AFR techniques. By improving the adaptability of feature recognition, VLMs enhance critical downstream tasks such as tool selection, process planning, and cost estimation.

2.2. Large Language Models and Vision-Language Models for Automatic Feature Recognition. LLMs are powerful computational models designed to understand and generate human language [62–65]. LLMs like *GPT-4* and *LLaMA* [66], built on Transformer architectures, have revolutionized natural language processing by enabling efficient sequential data handling and capturing long-range text dependencies [67–69]. These models excel in in-context learning, allowing them to acquire skills and adapt quickly to new tasks without retraining, making them invaluable for a variety of applications [70]. Similarly, conversational interfaces such as Google's *Bard* [71] and OpenAI's *ChatGPT* [24] have simplified access to complex information, enabling users to address tasks from medical diagnostics to manufacturing design without the need for complex technical jargon [19,72]. However, tasks requiring spatial understanding highlight the limitations of text-only interfaces. In such cases, visual representations are essential, as seen in manufacturing where engineers use CAD drawings to accurately communicate intricate details of design and assembly [73,74].

To address these challenges, VLMs have been developed to process both image and natural language text. Significant progress has been made in leveraging VLMs across various applications, including image manipulation (e.g., *StyleCLIP* [75], *DiffusionCLIP* [76]), text-based video retrieval (e.g., *X-CLIP* [77]), and manipulation (e.g., *Text2Live* [78]), as well as 3D shape and texture manipulation (e.g., *AvatarCLIP* [79], *CLIPFace* [80], *Text2Mesh* [81]). More recently, advanced multimodal language models such as *GPT-4o* and *Claude-3.5-Sonnet* have shown significant potential since their launch. These models are capable of processing both images and text inputs, producing text outputs. Given the visual nature of manufacturing features in CAD, leveraging the capabilities of such models offers an effective approach to overcome AFR challenges.

Traditional AFR methods heavily depend on expert-defined rules or extensive training data and face difficulties with 3D geometric details. In contrast, VLMs may offer a more adaptable solution for recognizing complex features through prompt engineering, which can reduce the dependence on large datasets and minimize the need for human intervention.

The pioneering study by Picard et al. [19] demonstrated the potential of VLMs, particularly OpenAI's *GPT-4V*, in AFR. Their research utilized a dataset comprising 20 CAD with common machining features, marking a significant initial exploration into VLM applications for AFR. However, the dataset's limited scope did not fully capture the complexity of diverse manufacturing processes. Moreover, the evaluation metrics employed were not comprehensive enough to thoroughly assess VLMs' capabilities in recognizing a wide range of manufacturing features. A more in-depth study is needed to incorporate a broader range of CAD and develop more rigorous evaluation methods to better understand and enhance the capabilities of VLMs in AFR.

To this end, this study addresses AFR challenges and aims to enhance accuracy by analyzing five SOTA VLMs using prompt engineering strategies. The research employs a diverse CAD dataset, each CAD model varying in complexity based on feature complexity, feature quantity, and expert judgment. Four key evaluation metrics are defined to systematically analyze the VLMs' performance. This systematic approach is intended to advance the capabilities of VLMs in AFR, making them more applicable to the manufacturing domain.

3 Methodology

This section describes the proposed methodology, detailing the creation of a custom CAD dataset, the application of VLMs, the use of prompt engineering strategies, and the definitions of the proposed evaluation metrics.

Figure 1 outlines the proposed methodology for evaluating VLMs in AFR from CAD. The process starts with converting a CAD file into three different image views using the *CAD2Image* converter, based on the PYTHON Open Cascade (*PyOCC*) library [82]. These images, combined with proposed prompts outlining the analysis criteria, serve as inputs to five SOTA VLMs. A series of six distinct experiments based on the proposed prompt engineering techniques is conducted, instructing the VLMs to adhere strictly to the provided prompts and output format. The VLMs generate answers in JSON format, listing identified features from a predefined list of manufacturing features, including details about each feature's existence and quantity. The VLMs' performance is then evaluated using four key metrics, comparing the identified features to ground truth features to ensure both accuracy and adherence to the evaluation criteria.

3.1 Dataset Creation. To explore the capabilities of VLMs in AFR, the study starts with the creation of a customized dataset consisting of 100 unique CAD, each annotated with a detailed list of manufacturing features. The dataset includes a broad range of features pertinent to various manufacturing processes and is organized into primary categories and subcategories to facilitate in-depth analysis. The main categories include *machining features*, *extrusion features*, *freeform features*, *molding and casting features*, and *sheet metal features*. Machining features are further divided into subcategories such as holes, slots, steps, pockets, edges, and contours (e.g., chamfers, fillets), threads and spirals, and additional features like necks. Extrusion features include pipes, tubes, and various boss shapes. Freeform features consist of depressions and protrusions, while molding and casting features comprise ribs, gussets, and drafts. Sheet metal features encompass various features related to sheet metal fabrication, such as bending, punching, and stamping.

The dataset is categorized into three levels of complexity: easy, medium, and hard, determined by feature quantity, complexity, and human expert judgment. The easy-level dataset includes 33 CAD parts, each featuring fewer than six simpler features and excluding complex features such as gussets, ribs, drafts, and sheet metal features. The medium-level dataset contains 33 CAD parts

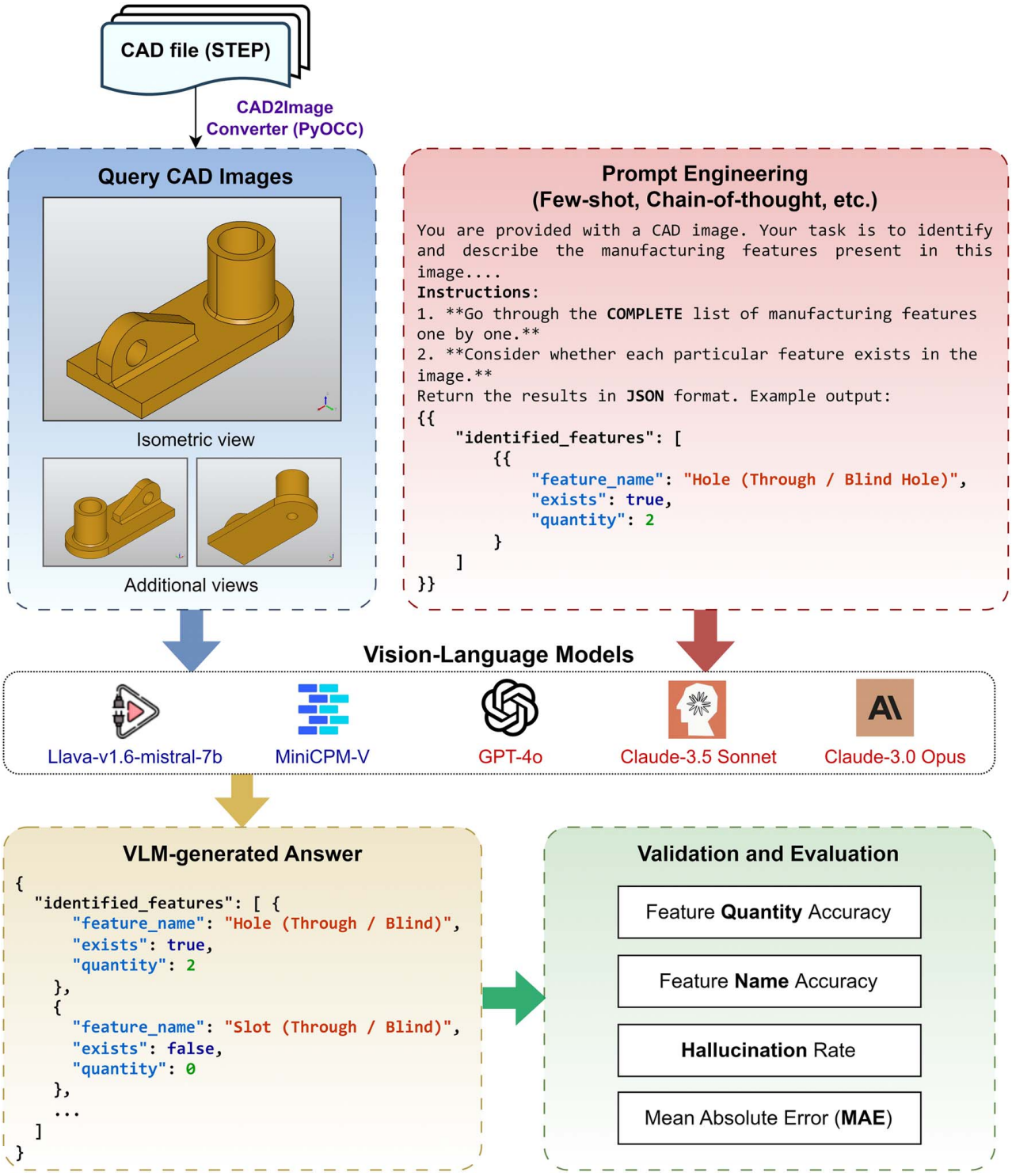


Fig. 1 Methodology overview

with designs of moderate complexity and includes nontraditional cylindrical stocks with a limited presence of complex features like sheet metal bends and threaded components. The difficult-level dataset consists of 34 CAD models, characterized by highly complex designs with numerous features, including intricate geometries such as casting and freeform features. Table 1 provides a detailed overview of the criteria used to define each complexity level.

Figure 2 provides visual examples from each dataset category, illustrating the varying levels of complexity. The easy-level examples showcase parts with fewer and simpler features, while the medium and difficult levels display increasingly complex designs. The CAD files are primarily designed by the authors, while particularly challenging parts are sourced from *GrabCAD*.[2] Each part is converted into three different image views using the *PyOCC* library to serve as input for the VLMs.

[2]https://grabcad.com/

Table 1 Complexity criteria for dataset categorization

Complexity level	Feature count	Feature complexity	Example features excluded	Example features included
Easy	<6	Simple, nonoverlapping geometries	Gussets, ribs, drafts, sheet metal features	Holes, slots, simple extrusions
Medium	3–28	Moderate complexity with nontraditional cylindrical stock	Freeform, complex sheet metal features	Threads, bends, nontraditional stock shapes
Hard	5–125	Highly complex with intricate geometries and overlaps	None	Casting features, freeform, complex assemblies

Each CAD part is manually labeled by domain experts, specifying the presence and quantity of each feature. This ground truth data is crucial for evaluating the performance of the VLMs. Figure 3 illustrates the feature distribution across the dataset's different complexity levels, highlighting variations such as the high prevalence of the "*hole*" feature in more complex CAD. Additionally, the hard category contains a notably higher occurrence of intricate features like sheet metal and freeform features,

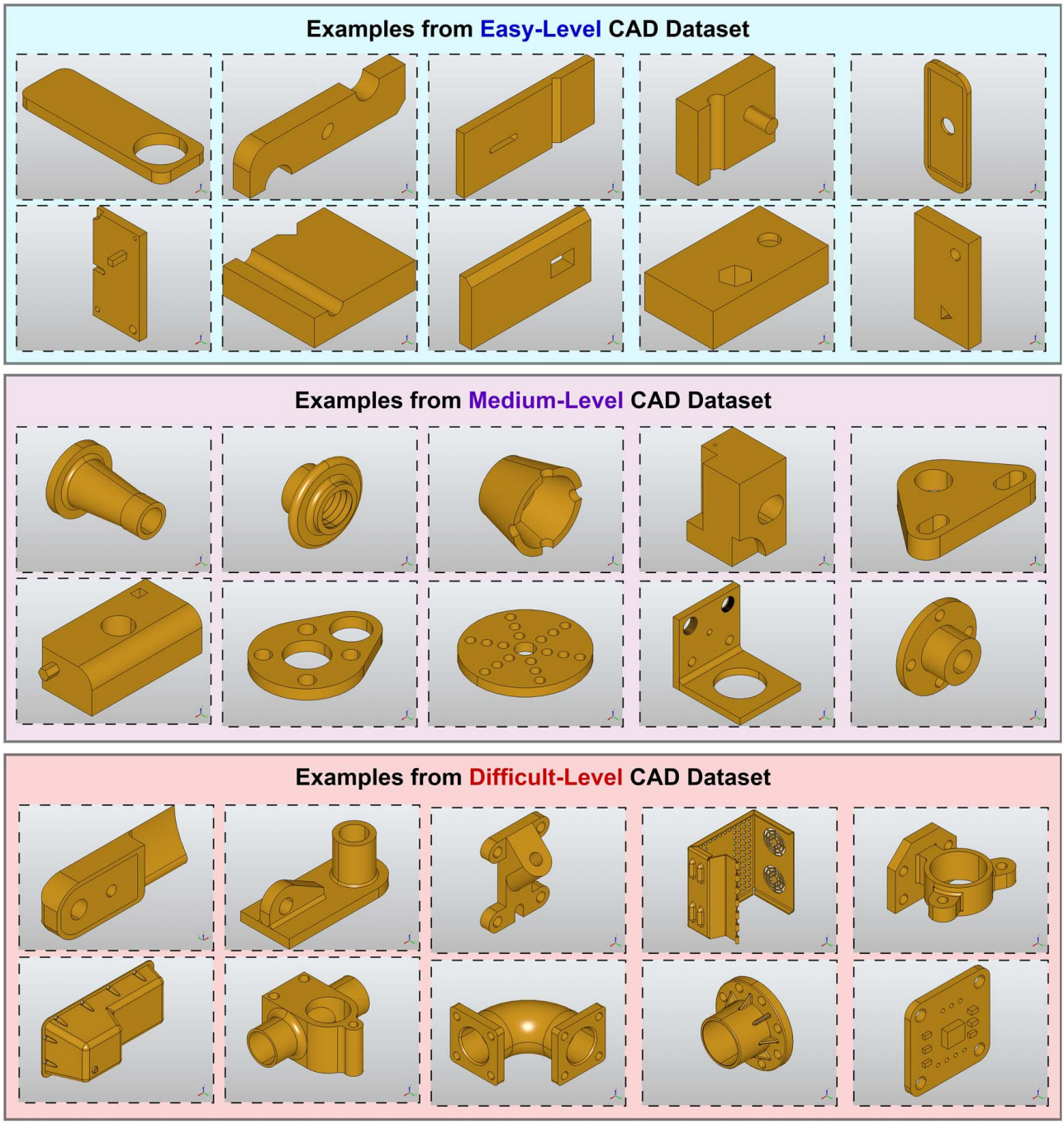


Fig. 2 Example CAD images from each category

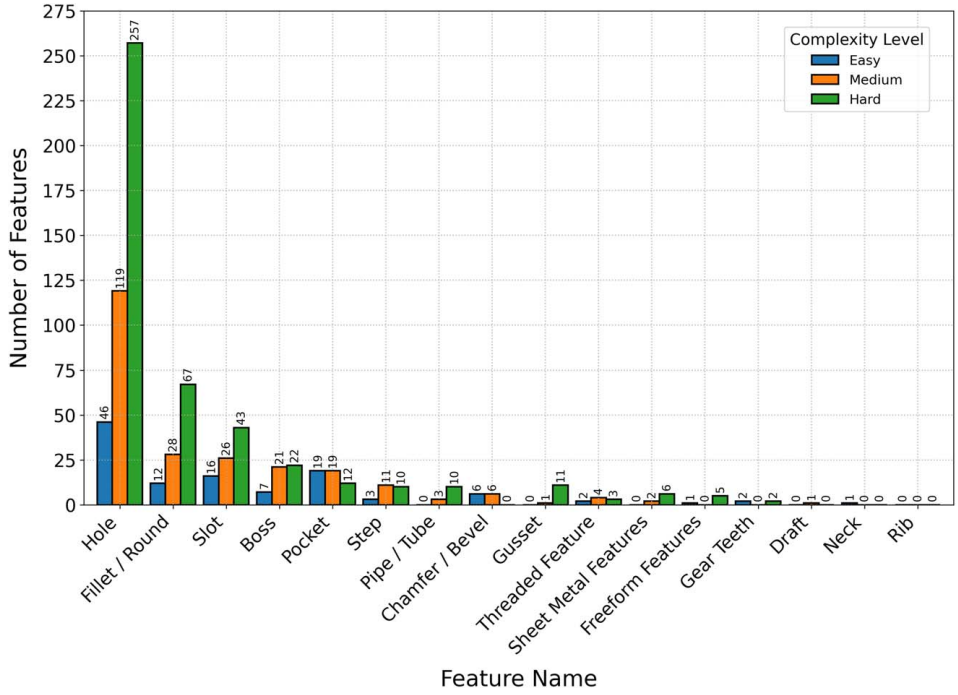


Fig. 3 Feature distribution across different dataset categories

which are less common or absent in the easy and medium categories.

3.2 Vision-Language Models. In this study, five SOTA VLMs, including three closed-source models and two open-source models, are implemented for manufacturing feature recognition:

- OpenAI's *GPT-4 omni* (gpt-4o-2024-08-06)[3] is selected, which is a closed-source VLM. GPT-4o is OpenAI's most advanced, cost-efficient, multilingual, and multimodal model that integrates visual inputs with text using a transformer-based architecture. It excels in traditional benchmarks for text, reasoning, and coding intelligence, while setting new records for multilingual, audio, and vision capabilities [83]. Its affordability and high accuracy in real-time vision tasks make it particularly well-suited for recognizing diverse manufacturing features in CAD models.
- Anthropic's *Claude-3.5-Sonnet* (claude-3-5-sonnet-20240620) [84] is a closed-source VLM that excels in vision benchmarking data, demonstrating exceptional performance in interpreting charts, graphs, and transcribing text from imperfect images [25].
- Anthropic's *Claude-3-Opus* (claude-3-opus-20240229) [84] is a closed-source VLM known for its advanced multimodal capabilities within the Claude 3 series. It demonstrates near-human comprehension and excels in complex vision tasks [26].
- *MiniCPM-Llama3-V2.5* (MiniCPM-Llama3-V-2_5) [27] is an open-source VLM that matches GPT-4V-level performance in a more compact form. It is optimized for efficient deployment on mobile and edge devices and surpasses similar VLMs in its class on benchmark datasets [85].
- *Llava-v1.6-mistral-7b* (llava-v1.6-mistral-7b-hf) [28] is an open-source VLM that offers enhanced image processing and logical reasoning capabilities. Built on the Mistral-7B language model, it achieves competitive performance in high-resolution multimodal data tasks [86].

[3]https://platform.openai.com

For all experiments described in the subsequent sections, the temperature for each VLM is set to zero.

3.3 Prompt Engineering. Prompt engineering involves creating natural language instructions, or prompts, to systematically extract knowledge from LLMs. Unlike traditional models, which often require extensive parameter retraining or fine-tuning for specific NLP tasks, the prompt engineering leverages the embedded knowledge within LLMs. This approach eliminates the need for extensive retraining, allowing researchers to interact with LLMs using natural language to achieve specific NLP tasks.

To address the second research question on enhancing the performance of VLMs, four commonly used prompt engineering techniques are implemented in this study: multiview query images, few-shot learning, sequential reasoning, and CoT reasoning. These prompt engineering techniques are discussed in detail in this section.

To assess the effectiveness of these strategies, we conduct six distinct experiments using various combinations of the prompt engineering techniques on each VLM:

- *Experiment 1: Zero-shot learning with a single CAD image view processed in parallel.*
- *Experiment 2: Zero-shot learning with a single CAD image view processed sequentially.*
- *Experiment 3: Zero-shot learning with multiple CAD image views.*
- *Experiment 4: Multiple image views with few-shot learning.*
- *Experiment 5: Multiple image views with zero-shot learning and CoT reasoning.*
- *Experiment 6: Multiple image views with few-shot learning and CoT reasoning.*

For all experiments, the temperature hyperparameter is set to zero to ensure consistent response behavior. Each experiment begins with a preamble to provide the VLMs with context, followed by the introduction of a manufacturing feature list and criteria under evaluation.

3.3.1 Manufacturing Feature List. In this study, a comprehensive dataset of CAD models is created, featuring a diverse range of

Manufacturing Feature List

```
{
    "Manufacturing Features": {
        "Machining Features": {
            "Hole (Through / Blind Hole)": [],
            "Slot (Through / Blind / T-Slot / Dovetail)": [],
            "Step (Through / Blind Step)": [],
            "Pocket (Blind / Through / Circular End Pocket)": [],
            "Edges & Contours": {
                "Chamfer / Bevel (Sharp Edge)": [],
                "Fillet / Round (Concave / Convex)": []
            },
            "Threads & Spirals": {
                "Threaded Feature": [],
                "Gear Teeth": []
            },
            "Additional Machining Features": {
                "Neck": []
            }
        },
        "Extrusion Features": {
            "Pipe / Tube": [],
            "Boss (Circular / Obround / Irregular / Rectangular, etc)": []
        },
        "Freeform Features (Depression, Protrusion)": [],
        "Molding & Casting Features": {
            "Rib": [],
            "Gusset": [],
            "Draft": []
        },
        "Sheet Metal Features": []
    }
}
```

Fig. 4 Manufacturing feature list

manufacturing features. These features are organized hierarchically to facilitate the recognition and categorization tasks performed by VLMs. Figure 4 illustrates this hierarchical organization in JSON format, where features are classified into five primary categories: machining features, extrusion features, freeform features, molding and casting features, and sheet metal features.

Each primary category is subdivided into relevant subcategories. For instance, within the *Edges & Contours* subcategory, features such as "chamfer" and "fillet" are included, while the *Threads & Spirals* subcategory lists features like "threads" and "gear teeth." These individual features, positioned at the lowest level of the hierarchy, are the specific targets for VLM recognition in CAD parts. The dataset contains 16 such features in total, and each CAD part is labeled with these ground truths.

Preamble (Single-view query image)

```
You are provided with a CAD image. Your task is to identify and describe the manufacturing features present in this image. Use the provided list of feature names, which is delimited by triple backticks. Use this list as a reference and strictly follow the naming conventions:

'''{manufacturing_features_names}'''
```

Preamble (Multi-vew query image)

```
You are provided with images containing multiple views of a 3D CAD model taken from different angles. Your task is to identify and describe the manufacturing features present in these images. Use the provided list of feature names, which is delimited by triple backticks. Use this list as a reference and strictly follow the naming conventions:

'''{manufacturing_features_names}'''
```

Fig. 5 Preamble prompt

This hierarchical structure categorizes features with similar geometric characteristics and manufacturing processes for efficiency. For example, various types of holes (such as blind, through, countersink, and tapered) are grouped under *Holes* due to their similar geometries and tooling requirements. This approach not only streamlines the VLMs' evaluation process by making features easily accessible but also ensures the input is concise to avoid overloading the VLMs.

3.3.2 Single-View Versus Multiview Query Images. Building on the hierarchical feature organization discussed earlier, this section explores how single-view and multiview query images are used to potentially improve the performance of VLMs in AFR. As illustrated in Fig. 5, the preamble for each type of image provides specific prompts to the VLM, instructing it to identify and describe the manufacturing features from the images. These prompts are crafted using the feature names from the comprehensive list outlined in the previous section.

For single-view query images, a single CAD image is provided to the VLM. This setup restricts the VLM to analyzing the object from only one angle, limiting the perspective available. Conversely, multiview query images, as shown in Fig. 6, present three different views of a CAD model. This approach offers a broader visualization, potentially allowing the VLMs to access features that are not visible from a single isometric view. We anticipate that this method will result in higher accuracy, as incorporating multiple views significantly enhances the visual spatial reasoning capabilities of VLMs.

3.3.3 Zero-Shot Versus Few-Shot Prompt. Zero-shot and few-shot prompts are techniques used to interact with LLMs and VLMs across various domains. Zero-shot prompting requires language models to perform tasks based solely on their pretrained knowledge, without prior examples [70]. In this study, VLMs are directed to identify manufacturing features using an input query image along with textual prompts in a zero-shot learning approach, without the benefit of additional labeled examples.

Conversely, few-shot prompting provides the language models with one or more examples to impart task-specific insights. This approach generally outperforms zero-shot prompting in various benchmarks [21]. In this study, a *one-shot learning* method [22] is utilized, where the VLMs are supplied with the input query image and a manufacturing feature list, along with a single illustrative example. This example, shown in Fig. 7, includes a CAD and a structured JSON output detailing the ground truth features with their names and quantities. This comprehensive example includes all potential manufacturing features to aid the VLMs in learning and recognizing these features in unseen CAD images.

3.3.4 Parallel Prompt Versus Sequential Prompt. The structuring of input prompts, either parallel or sequential, serves as a useful technique to effectively query VLMs in identifying manufacturing features, as illustrated in Fig. 8.

Parallel prompting presents all relevant instructions to the VLMs simultaneously. This approach allows the VLMs to access all necessary data at once, enhancing processing efficiency and reducing response times. As demonstrated in Fig. 8, the VLM receives a complete list of features along with instructions to identify feature names and quantities in CAD. Although this method reduces the number of prompts needed and improves scalability for managing multiple feature categories, it may also increase the likelihood of model errors or hallucinations due to the complexity of processing multiple inputs at the same time [65].

Sequential prompting requires VLMs to process information in stages, building upon each previous step. This method is particularly effective for managing complex information. In this study, sequential prompting is applied in a structured manner: the VLM first reviews a comprehensive manufacturing features list and then identifies the presence and quantity of each feature within the

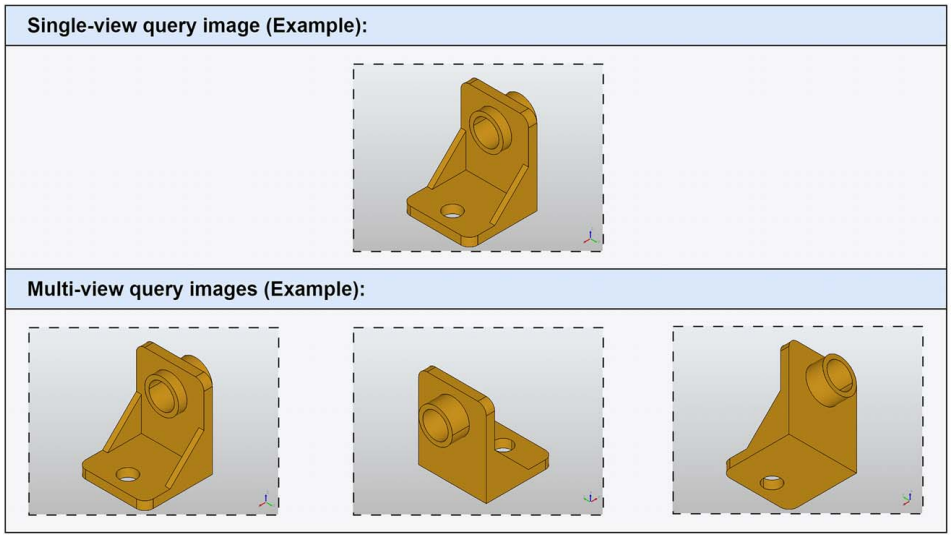


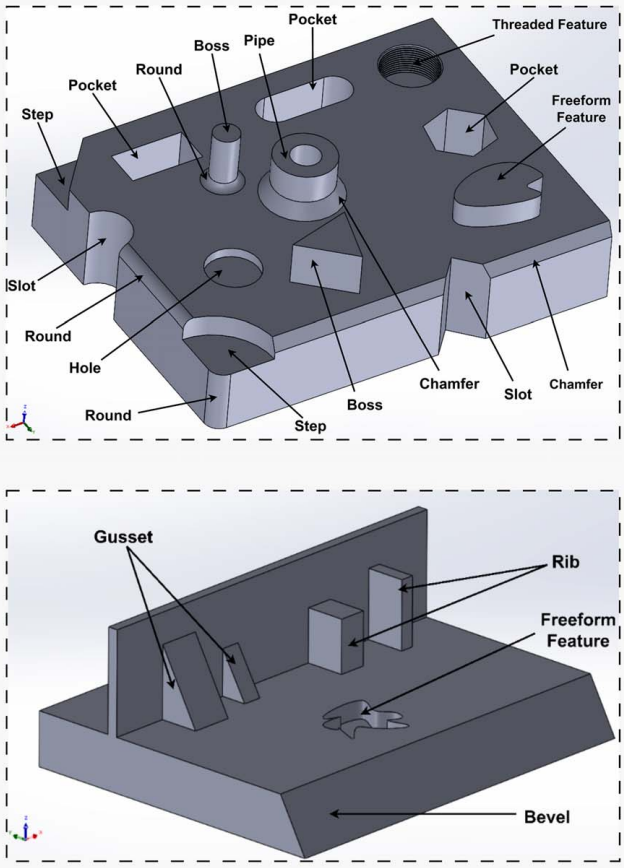
Fig. 6 Query CAD images: single view and multiview

Few-shot Learning Prompt

```
###Example Task:
Before proceeding with your task, here is an example to illustrate the expected format and
content of your response.

(Example image and answer will be provided below.)
```

Example image



Example answer


```
{
  "identified_features": [ {
      "feature_name": "Hole (Through / Blind Hole)",
      "exists": true,
      "quantity": 2
    },
    {
      "feature_name": "Slot (Through / Blind)",
      "exists": true,
      "quantity": 2
    },
    {
      "feature_name": "Step (Through / Blind Step)",
      "exists": true,
      "quantity": 2
    },
    {
      "feature_name": "Pocket (Blind / Through)",
      "exists": true,
      "quantity": 3
    },
    {
      "feature_name": "Chamfer / Bevel",
      "exists": true,
      "quantity": 2
    },
    {
      "feature_name": "Fillet / Round",
      "exists": true,
      "quantity": 3
    },
    {
      "feature_name": "Threaded Feature",
      "exists": true,
      "quantity": 1
    },
    ...
  ]
}
```


Fig. 7 Few-shot learning prompt

Parallel Prompt

```
Return the results in JSON format.
**Do not include any other content, including any punctuation around the text.**
Example output format:
{{
    "identified_features": [
        {{
            "feature_name": "Hole (Through / Blind Hole)",
            "exists": true,
            "quantity": 2
        }},
        {{
            "feature_name": "Fillet / Round (Concave / Convex)",
            "exists": false,
            "quantity": 0
        }}
    ]
}}
```

Sequential Prompt

```
Instructions:
1. **Review the COMPLETE list of manufacturing features one by one.**
2. **Determine whether each feature exists in the given images.**
3. **Consider the **hierarchy** of each feature name from the provided list to understand its
meaning in the manufacturing context.**
4. **Identify features using the provided list, adhering strictly to the naming conventions.**
5. **Examine each view of the part to identify all features. Different angles may reveal
features not visible in other views.**
6. **For each identified feature, provide the quantity:** The number of such features present
in the images (e.g., if there are two "Through Holes", then the quantity is 2). This should be
a numerical integer (0, 1, 2, ...).
```

Fig. 8 Parallel prompts versus sequential prompts

CAD images. This systematic assessment of each feature tends to enhance accuracy, though it may increase processing time.

3.3.5 Chain-of-Thoughts. Research has demonstrated that incorporating intermediate reasoning steps, known as a CoT, can significantly enhance the performance of LLMs across various reasoning tasks [87]. This approach has further evolved into the tree-of-thoughts [88], which introduces multiple reasoning pathways to further improve the decision-making capabilities of language models [89].

In this study, a three-step CoT reasoning framework is implemented to enhance feature recognition in CAD. First, the VLM evaluates the presence of a feature based on visual observations. Next, it analyzes the manufacturing context and feature hierarchy to assess their influence on feature recognition. Finally, the VLM determines the exact number of occurrences of the feature, providing a precise numerical output. This structured reasoning process, illustrated in Fig. 9, mirrors human cognitive approaches to problem-solving and ensures a detailed contextual analysis.

3.4 Evaluation Metrics. To address the third research question regarding the effective quantification and evaluation of VLMs in AFR, this study employs four key evaluation metrics to accurately assess the performance of the VLMs.

3.4.1 Feature Name Accuracy. Feature name accuracy (FNA) measures the correctness of the feature names identified by the VLM. It is defined as the ratio of correctly identified feature names to the total number of ground truth feature names in a CAD:

$$\text{FNA} = \frac{\text{correctly identified names}}{\text{total ground truth names}} \times 100\% \quad (1)$$

A high FNA value indicates that the identified features are correctly named. Accurate naming is crucial for ensuring that subsequent manufacturing processes are based on precise and reliable feature data.

3.4.2 Feature Quantity Accuracy. Feature quantity accuracy (FQA) evaluates the VLM's ability to recognize the correct number of features. It is defined as the ratio of the true positive quantity of features to the ground truth quantity:

$$\text{FQA} = \frac{\text{true positive quantity}}{\text{ground truth quantity}} \times 100\% \quad (2)$$

A high FQA value indicates that the model accurately identifies the correct number of each feature. This metric is crucial as it ensures accurate feature counts, optimizing manufacturing workflows and resource use, while reducing waste and enhancing production efficiency.

3.4.3 Hallucination Rate. Hallucination rate (HR) measures the frequency of incorrect or irrelevant features generated by the VLM. It is defined as the ratio of hallucinated features to the total

Chain-of-Thoughts (CoT) Prompt

```
Instructions:
1. **Go through the COMPLETE list of manufacturing features one by one.**
2. **For each feature, follow these steps to determine its existence and quantity:**
     **Step 1:** Identify the feature present in the CAD image. Describe what you see that
makes you think the feature exists or does not exist.
     **Step 2:** Consider the manufacturing context and hierarchy. Think about how this
context influences your decision.
     **Step 3:** Determine the quantity of the feature. Count the occurrences of this feature
in the image.
3. **Provide your answers in a structured format for each feature, including your reasoning
for existence and quantity.**
4. **Ensure that each feature from the provided list has been considered in your final
output.**
5. **Keep the reasoning concise and to the point, no more than one or two sentences.**
```

Fig. 9 Chain-of-thoughts prompt

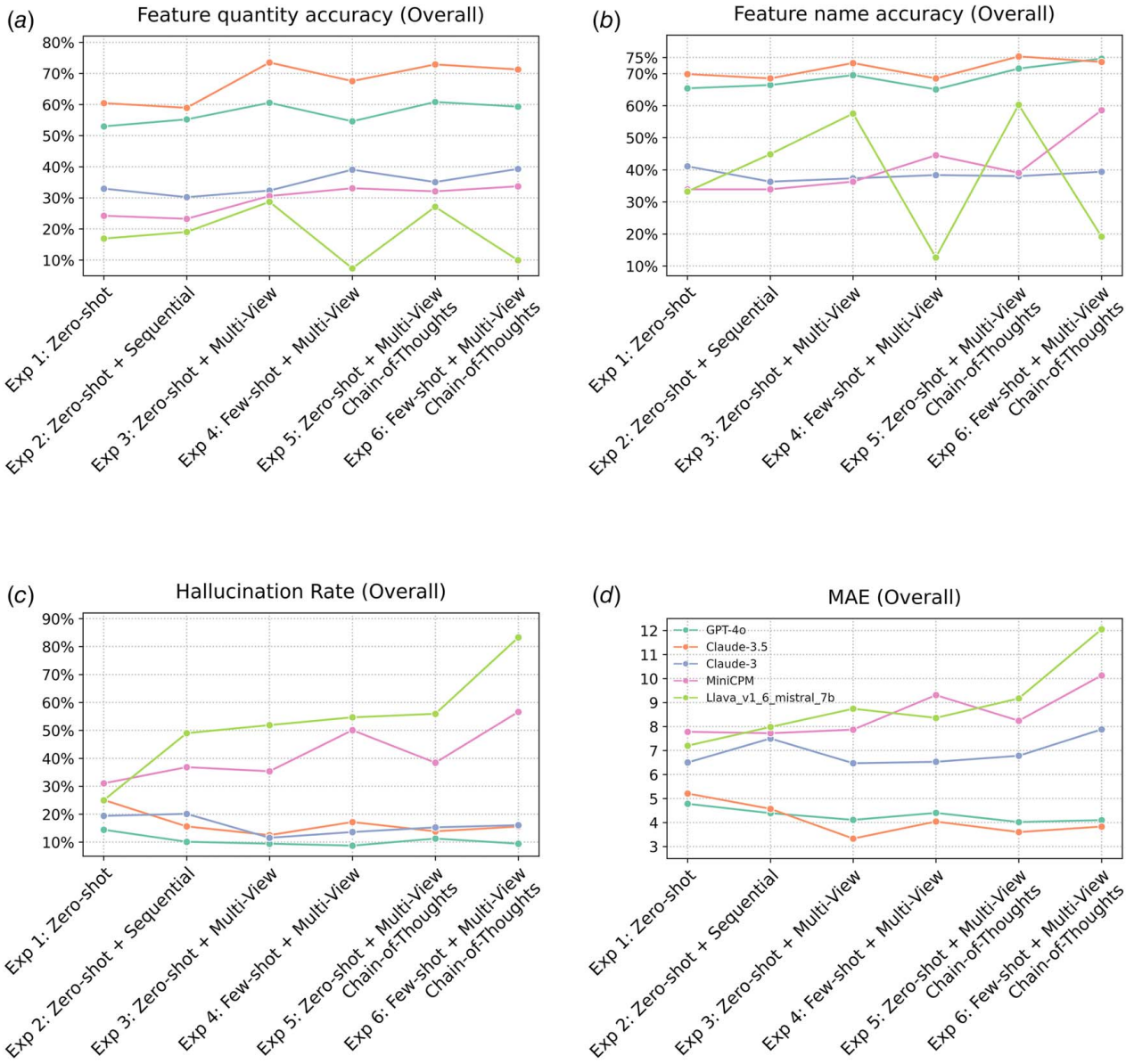


Fig. 10 Overall performance of five VLMs across six experiments for 100 CAD models. Claude-3.5 outperforms all other VLMs in feature quantity accuracy, feature name accuracy, and MAE. Conversely, GPT-4o excels in minimizing hallucination rates, while open-source VLMs demonstrate the lowest overall performance.

predicted features:

$$\text{HR} = \frac{\text{hallucinated quantity}}{\text{predicted quantity}} \times 100\% \tag{3}$$

A low HR value indicates fewer incorrect predictions, critical for accurate cost estimations and efficient process planning, avoiding costly manufacturing errors.

3.4.4 Mean Absolute Error. Mean absolute error (MAE) measures the difference between the predicted feature quantities and the actual quantities. It is calculated as the average of the absolute differences between the predicted and ground truth quantities for all features in a CAD:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |Q_i - \widehat{Q}_i| \tag{4}$$

where n is the total number of features being evaluated, Q_i is the ground truth quantity of the ith feature, $\widehat{Q}_i$ is the predicted quantity of the ith feature, and $|Q_i - \widehat{Q}_i|$ represents the absolute error of the ith prediction. This metric provides a clear measure of prediction accuracy. A small MAE value, closer to 0, indicates high accuracy in predicting feature quantities.

The four metrics serve as benchmarks to compare the VLMs' performance against human experts in feature labeling accuracy, quantity estimation, and error minimization.

4 Results and Discussion

This section presents an in-depth evaluation of five VLMs across six distinct experiments, designed to test their capability in recognizing manufacturing features in CAD. The analysis spans three levels of CAD complexity and employs a variety of prompt engineering techniques to assess VLMs' performance. Key evaluation metrics are used to compare the effectiveness of each VLM.

4.1 Overall Vision-Language Models' Performance Evaluation. The evaluation of five VLMs across six distinct experiments involving 100 CAD has yielded several key insights, as shown in Fig. 10. Claude-3.5 achieves the highest feature quantity

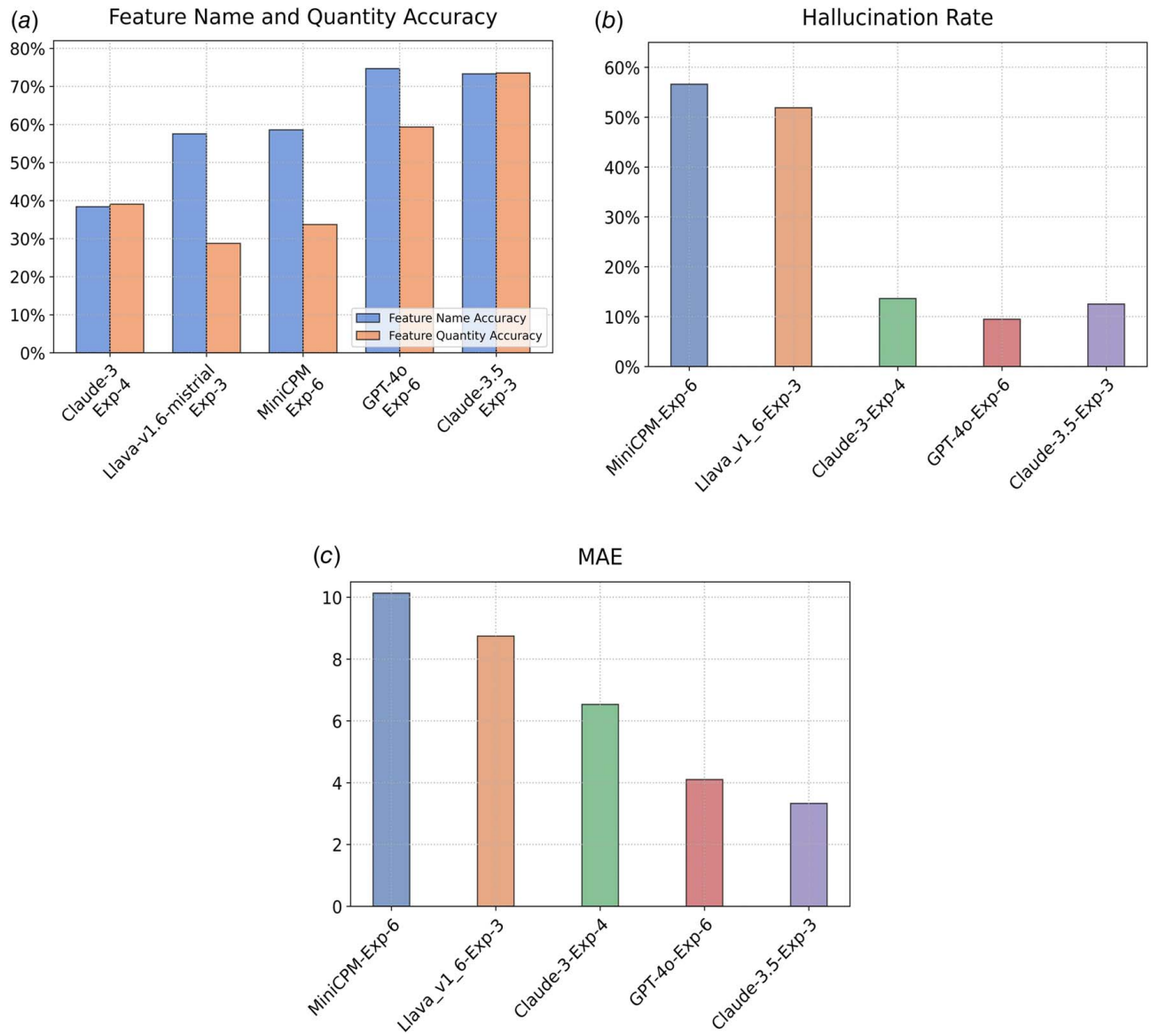


Fig. 11 Performance comparison between the top-performing experiments under each VLM. Claude-3.5 leads in feature quantity accuracy and MAE in Experiment 3, whereas GPT-4o excels in feature name accuracy and exhibits the least hallucinations.

accuracy of 74% using a zero-shot and multiview prompt strategy (Experiment 3). Claude-3.0, while outperforming open-source VLMs with an accuracy of 40%, does not match the performance of Claude-3.5. Open-source models consistently show lower accuracies (<40%). Claude-3.5 shows strong performance with zero-shot and multiview prompts, indicating that well-crafted zero-shot prompts can sometimes outperform few-shot prompts. This may be due to the fact that few-shot prompts can introduce ambiguity by being misinterpreted as part of a narrative rather than clear instructions.

Both GPT-4o and Claude-3.5 achieve a feature name accuracy of 75% in their optimal configurations: GPT-4o with a multiview and Few-shot-CoT setup (Experiment 6) and Claude-3.5 with a multiview and Zero-shot-CoT approach (Experiment 5). GPT-4o's success in few-shot learning underscores its ability to generalize effectively when provided with relevant labeled examples. This aligns with research indicating that few-shot prompts are most effective when the in-context examples closely align with the test cases.

In terms of hallucination rate, GPT-4o consistently leads across all experiments with the lowest rate of 8% in a few-shot with multiview configuration (Experiment 4). Claude-3.5 shows slightly higher hallucinations at 12%, while open-source models exhibit the highest rates, exceeding 30%. This trend suggests a performance decline in open-source models as prompt complexity increases, likely due to their smaller context windows compared to closed-source models.

Regarding MAE, Claude-3.5 records the lowest value at 3.2 in a zero-shot with multiview setup (Experiment 3), outperforming all other VLMs. Claude-3.0 attains a slightly better MAE value at 6.4 compared to open-source models but remains less effective than the other closed-source models. The open-source models show high MAE values as high as 12.0.

Figure 11 provides a comparative analysis of the top-performing experiments for each VLM. Overall, Claude-3.5 in the zero-shot with multiview setup (Experiment 3) emerges as the superior model based on accuracy and MAE scores, while GPT-4o is particularly effective in minimizing hallucinations. These findings highlight the distinct strengths of Claude-3.5 and GPT-4o across different evaluation metrics.

4.2 Performance Evaluation on Easy-Difficulty Computer-Aided Designs. The performance analysis of five distinct VLMs on 33 easy-difficulty CAD is summarized in Fig. 12. GPT-4o

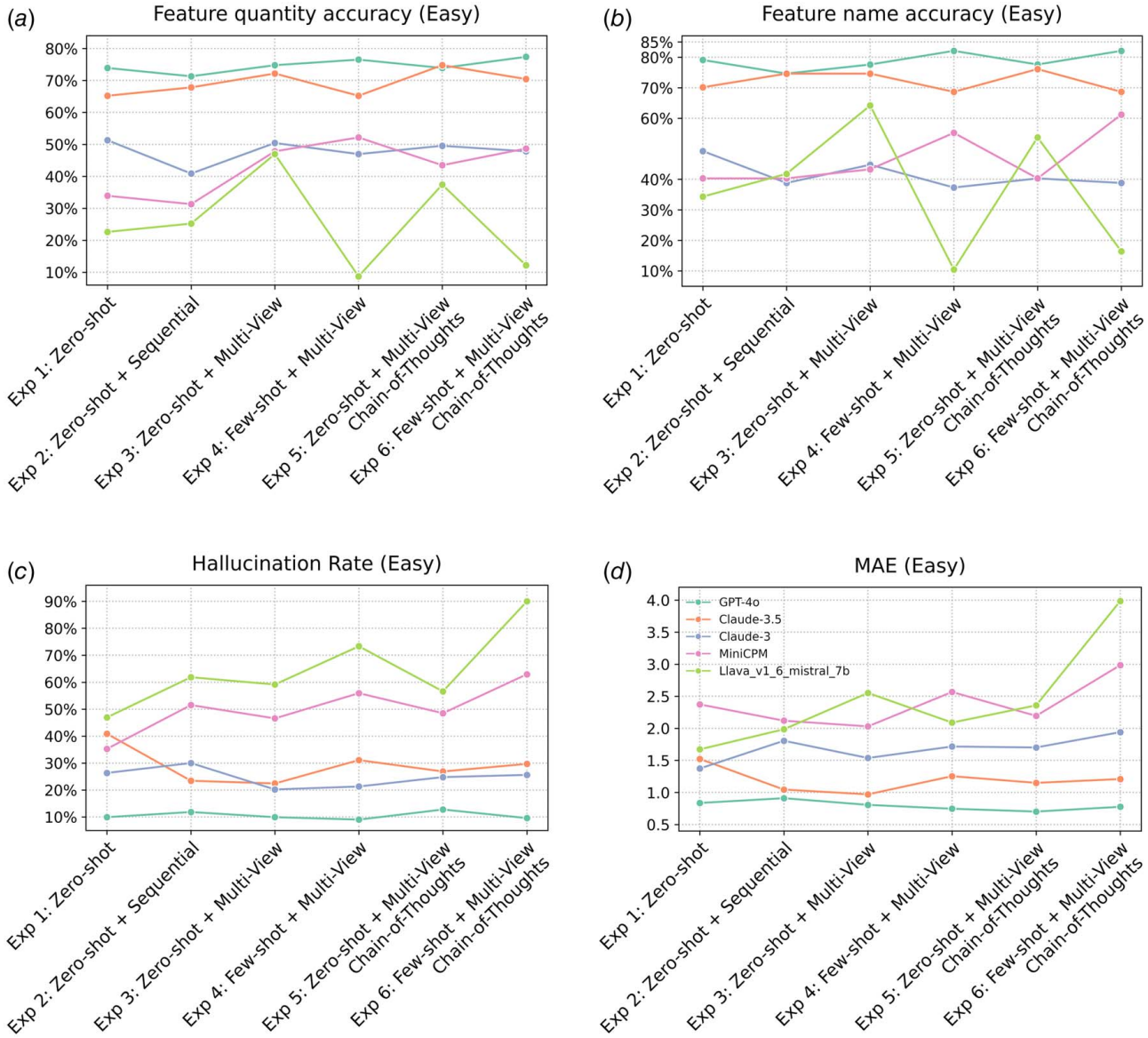
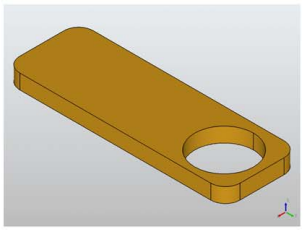
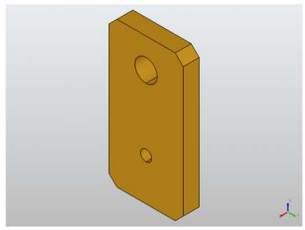
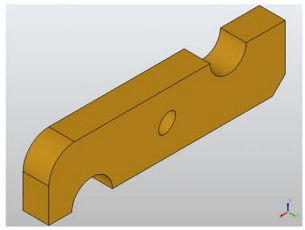
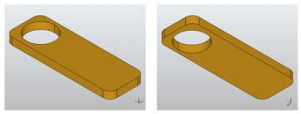
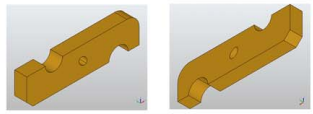


Fig. 12 Performance evaluation of five VLMs across six experiments for 33 easy CAD models. GPT-4o, particularly in Experiment 6, outperforms all other VLMs across all evaluation metrics.

Manufacturing Feature Recognition on Easy-Level CAD Images			
	Example 1	Example 2	Example 3
Isometric View			
Additional Views			
Ground Truth (by human expert)	• Hole (through/blind): 1 • Fillet/Round: 4	• Hole (through/blind): 2 • Chamfer/Bevel: 4	• Hole (through/blind): 1 • Fillet/Round: 1 • Chamfer/Bevel: 1 • Slot (through/blind): 2
GPT-4o Exp 6: Few-shot, Multi-view, CoT	✅ Hole (through/blind): 1 ✅ Fillet/Round: 4	✅ Hole (through/blind): 2 ❌ Chamfer/Bevel: 2	✅ Hole (through/blind): 1 ❌ Fillet/Round: 2 ❌ Chamfer/Bevel: 0 ❌ Slot (through/blind): 0
Claude-3.5 Exp 5: Zero-shot, Multi-view, CoT	✅ Hole (through/blind): 1 ❌ Fillet/Round: 8 ⚠️ Chamfer/Bevel: 8	✅ Hole (through/blind): 2 ❌ Chamfer/Bevel: 1 ⚠️ Step (through/blind): 1	✅ Hole (through/blind): 1 ❌ Fillet/Round: 2 ❌ Chamfer/Bevel: 0 ❌ Slot (through/blind): 2 ⚠️ Step (through/blind): 2
Claude-3 Exp 3: Zero-shot, Multi-view	❌ Hole (through/blind): 2 ✅ Fillet/Round: 4	✅ Hole (through/blind): 2 ❌ Chamfer/Bevel: 0	❌ Hole (through/blind): 2 ✅ Fillet/Round: 1 ❌ Chamfer/Bevel: 0 ❌ Slot (through/blind): 0
MiniCPM-Llama3 Exp 4: Few-shot, Multi-view	✅ Hole (through/blind): 2 ❌ Fillet/Round: 0 ⚠️ Chamfer/Bevel: 1	❌ Hole (through/blind): 3 ❌ Chamfer/Bevel: 1	❌ Hole (through/blind): 2 ❌ Fillet/Round: 0 ✅ Chamfer/Bevel: 1 ❌ Slot (through/blind): 3 ⚠️ Step (through/blind): 1 ⚠️ Pocket (through/blind): 1 ⚠️ Boss: 1 ⚠️ Freeform features: 1
Llava-1.6-mistral-7b Exp 3: Zero-shot, Multi-view	✅ Hole (through/blind): 2 ❌ Fillet/Round: 0 ⚠️ Chamfer/Bevel: 1	✅ Hole (through/blind): 2 ❌ Chamfer/Bevel: 1 ⚠️ Slot (through/blind): 1 ⚠️ Step (through/blind): 1 ⚠️ Pocket (through/blind): 1 ⚠️ Neck: 1	❌ Hole (through/blind): 2 ❌ Fillet/Round: 0 ✅ Chamfer/Bevel: 1 ❌ Slot (through/blind): 0

Note: ✅ *represents correct prediction;* ❌ *represents false prediction or misprediction;* ⚠️ *represents hallucination (extra feature not in ground truth)*

Fig. 13 Examples of ground truth and VLM-generated features for easy-level CAD. GPT-4o in Experiment 6 shows the highest accuracy and the lowest rate of hallucinated features compared to other setups.

achieves the highest feature quantity accuracy at 79% using the multiview with Few-shot-CoT prompt (Experiment 6), while the open-source models demonstrate the lowest accuracy in this category. GPT-4o also leads in feature name accuracy, achieving 83% in both Experiments 4 and 6, likely benefiting from the effective use of closely matching examples in few-shot learning.

In terms of hallucination rates, GPT-4o consistently records the lowest rate of 9% across multiple experiments. Although Claude-3.5 shows slightly higher hallucination rates than GPT-4o, both models significantly outperform the open-source models, which exhibit the highest rates. GPT-4o also records the lowest MAE of 0.65 in Experiment 5, maintaining strong performance

across other setups, while the open-source models display the highest MAE values.

Figure 13 showcases examples of VLM-generated results for easy CAD. GPT-4o, particularly in Experiment 6, stands out as the top-performing model on this test set, excelling across all evaluation metrics. Notably, all VLMs frequently struggle to differentiate between "chamfer" and "fillet" features, likely due to their subtle visual differences. Moreover, open-source models tend to generate more hallucinated features compared to their closed-source counterparts. Unlike GPT-4o, which tends to underestimate features, leading to conservative estimates, Claude-3.5 shows slightly higher hallucination rates, risking overestimations that could inflate costs in downstream tasks.

4.3 Performance Evaluation on Medium-Difficulty Computer-Aided Designs. The performance analysis of five VLMs on 33 medium-difficulty CAD is summarized in Fig. 14. Claude-3.5 achieves the highest feature quantity accuracy of 74.5% in Experiment 5, while both Claude-3.5 and GPT-4o attain the highest feature name accuracy, each reaching 74.9%. In contrast, Llava-v1.6-mistral-7b records the lowest accuracies, with feature quantity accuracy dropping to 10% and feature name accuracy to 15%.

In terms of hallucination rates, GPT-4o demonstrates the lowest rate of 8% across multiple experiments. The open-source models, however, exhibit significantly higher hallucination rates (>30%), reflecting their difficulty in handling the increased complexity of medium-level CAD. Furthermore, GPT-4o achieves the lowest MAE of 0.7 in Experiment 6, maintaining strong performance across various setups. In comparison, the open-source models show notably higher MAE values, exceeding 2.0.

Figure 15 provides visual examples of predictions on medium-level CAD, identifying GPT-4o in Experiment 6 as the most effective in minimizing errors and hallucinations. Notably, all VLMs struggle to differentiate between "pipe/tube" and "boss" features, likely due to their visual similarities. The open-source models particularly perform poorly, frequently misidentifying or hallucinating features across the CAD.

4.4 Performance Evaluation on Challenging Computer-Aided Designs. The evaluation of five VLMs on 34 challenging CAD is detailed in Fig. 16. Claude-3.5 stands out by

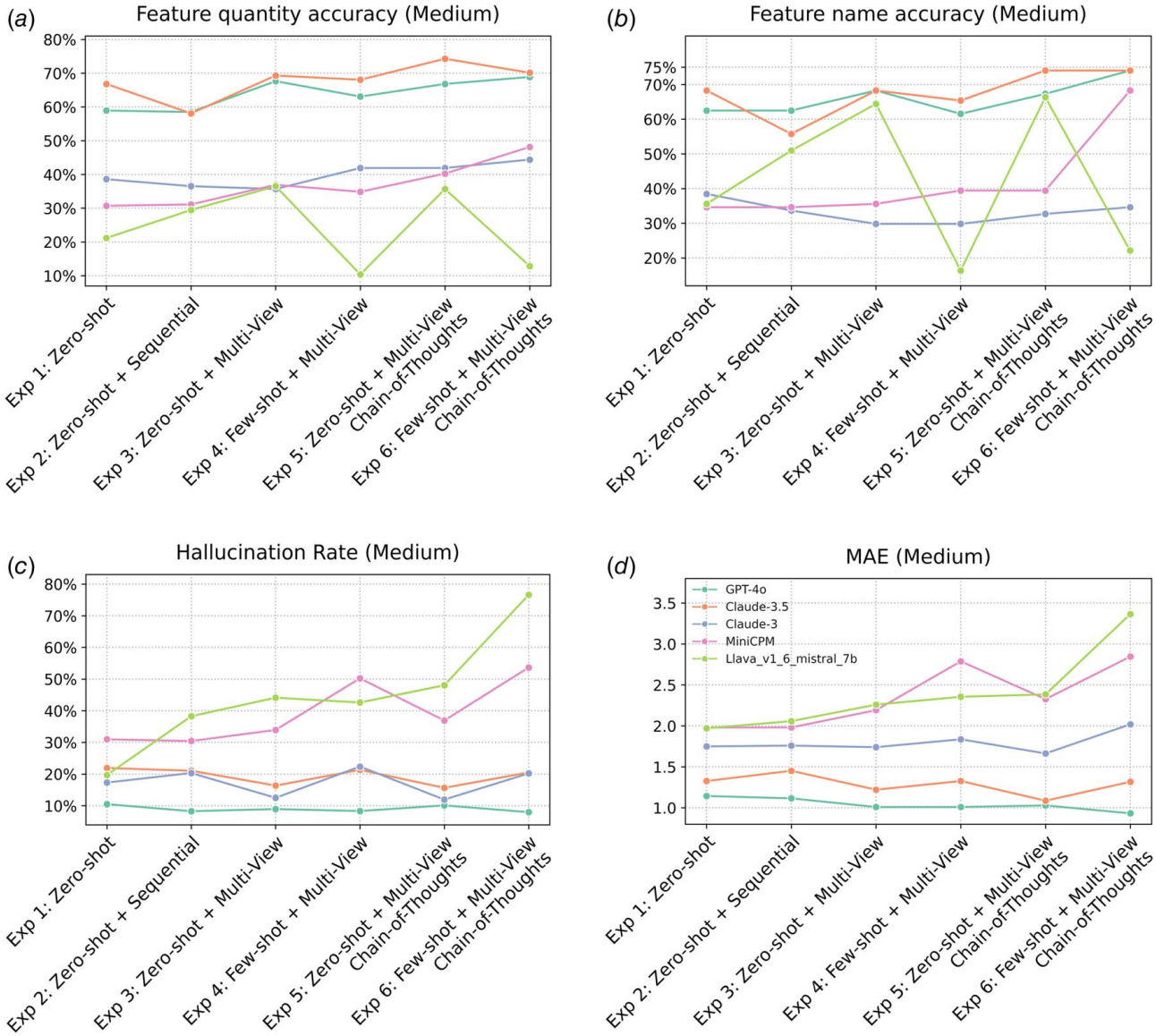
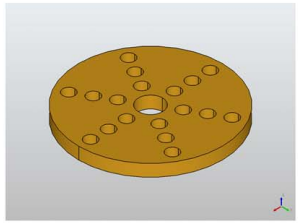
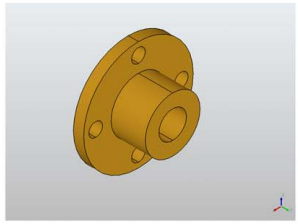
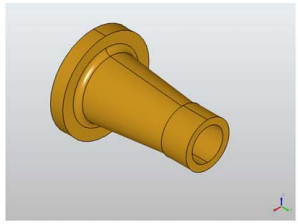
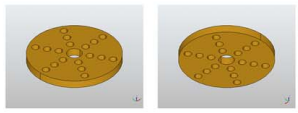
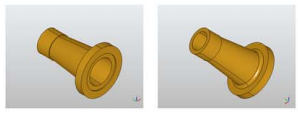


Fig. 14 Performance evaluation of five VLMs across six experiments for 33 medium CAD models. Claude-3.5 and GPT-4o show high accuracy in feature name and quantity, significantly outperforming other VLMs, and exhibit the lowest rates of hallucinated features.

Manufacturing Feature Recognition on Medium-Level CAD Images			
	Example 1	Example 2	Example 3
Isometric View			
Additional Views			
Ground Truth (by human expert)	• Hole (through/blind): 19	• Hole (through/blind): 5 • Pipe/Tube: 1	• Hole (through/blind):1 • Boss: 1 • Fillet/Round: 1 • Pipe/Tube: 1
GPT-4o Exp 6: Few-shot, Multi-view, CoT	✗ Hole (through/blind): 18	✓ Hole (through/blind): 5 ✗ Pipe/Tube: 0 ⚠ Boss: 1	✓ Hole (through/blind): 1 ✓ Boss: 1 ✓ Fillet/Round: 1 ✗ Pipe/Tube: 0
Claude-3.5 Exp 3: Zero-shot, Multi-view, CoT	✗ Hole (through/blind): 20	✓ Hole (through/blind): 5 ✗ Pipe/Tube: 0 ⚠ Boss: 1	✓ Hole (through/blind): 1 ✓ Boss: 1 ✗ Fillet/Round: 2 ✗ Pipe/Tube: 0
Claude-3 Exp 5: Zero-shot, Multi-view, CoT	✓ Hole (through/blind): 19 ⚠ Fillet/Round: 1	✗ Hole (through/blind): 6 ✗ Pipe/Tube: 0	✗ Hole (through/blind): 0 ✗ Boss: 0 ✗ Fillet/Round: 2 ✗ Pipe/Tube: 0
MiniCPM-Llama3 Exp 6: Few-shot, Multi-view, CoT	✗ Hole (through/blind): 10 ⚠ Pocket (through/blind): 1 ⚠ Chamfer/Bevel: 2	✗ Hole (through/blind): 4 ✗ Pipe/Tube: 0 ⚠ Pocket (through/blind): 1 ⚠ Chamfer/Bevel: 2 ⚠ Boss: 2	✓ Hole (through/blind): 1 ✓ Boss: 1 ✗ Fillet/Round: 0 ✗ Pipe/Tube: 0 ⚠ Pocket (through/blind): 1 ⚠ Chamfer/Bevel: 2 ⚠ Freeform Features: 1 ⚠ Step (through/blind): 2
Llava-1.6-mistral-7b Exp 5: Zero-shot, Multi-view, CoT	✗ Hole (through/blind): 2 ⚠ Pocket (through/blind): 1 ⚠ Fillet/Round: 1	✗ Hole (through/blind): 2 ✗ Pipe/Tube: 0 ⚠ Fillet/Round: 1	✗ Hole (through/blind): 2 ✗ Boss: 0 ✓ Fillet/Round: 1 ✗ Pipe/Tube: 0 ⚠ Chamfer/Bevel: 1 ⚠ Threaded Features: 1 ⚠ Neck: 1

Note: ✓ *represents correct prediction;* ✗ *represents false prediction or misprediction;* ⚠ *represents hallucination (extra feature not in ground truth)*

Fig. 15 Examples of ground truth and VLM-generated answers for medium-level CAD. GPT-4o and Claude-3.5 both show high accuracy and the lowest rate of hallucinated features compared to other VLMs.

achieving the highest feature quantity accuracy at 76% and feature name accuracy at 77% in Experiment 3. In contrast, Llava-v1.6-mistral-7b records the lowest accuracies, with only 6% in feature quantity and 10% in feature name accuracy.

Claude-3 achieves the lowest hallucination rate of 9% in Experiment 4, closely followed by Claude-3.5. Additionally, Claude-3.5 shows the lowest MAE, scoring 1.4 in Experiment 3, whereas the open-source VLMs exhibit considerably higher hallucination rates (>28%) and the highest MAE values (>3.3). This highlights a significant performance gap between open-source and closed-source models in analyzing challenging CAD.

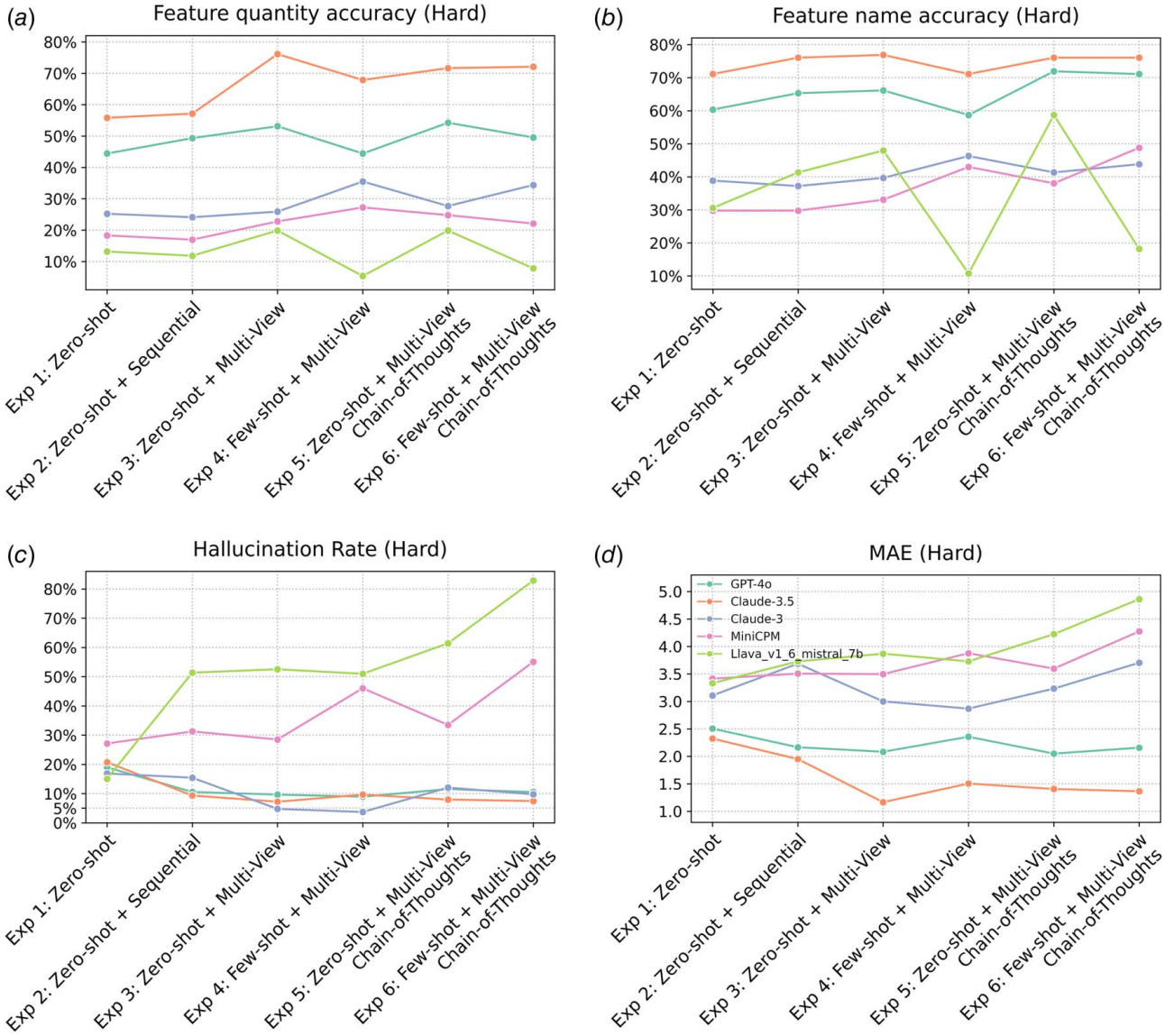


Fig. 16 Performance evaluation of five VLMs across six experiments for 34 challenging CAD models. Claude-3.5 in Experiment 3 demonstrates high accuracy in feature name and quantity and exhibits the lowest rates of hallucinated features.

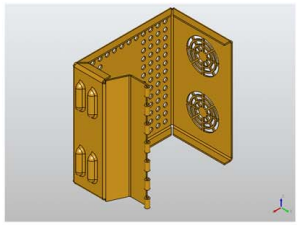
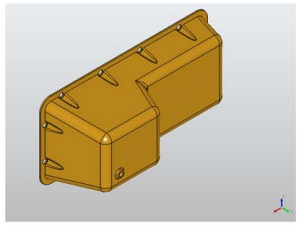
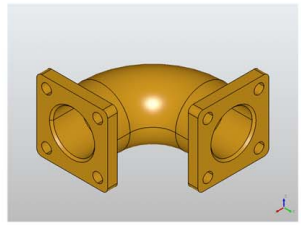
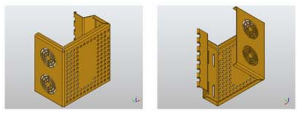
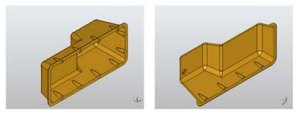
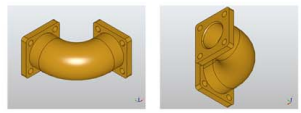
Figure 17 presents visual examples of predictions on these designs, demonstrating that Claude-3.5 in Experiment 3 excels across all metrics. This VLM's proficiency is especially apparent in its ability to closely match the ground truth in designs with high feature counts, such as predicting 114 out of 119 "blind holes" in one example. The overall high accuracy in recognizing challenging parts suggests that tasks difficult for human experts might be more manageable for VLMs.

4.5 Comparative Analysis With Prior Automatic Feature Recognition Methods. To better highlight the contributions of the proposed VLM-based AFR framework, a comparative analysis is presented in Table 2. This comparison contrasts the proposed approach with representative AFR methods developed over the past decade in terms of input modality, data requirements, recognition capabilities, and generalizability. Traditional AFR methods typically rely on 3D geometric representations such as voxel grids, boundary representations (B-Reps), or multiview CAD 2D slices, processed using rule-based algorithms or graph neural networks (GNNs). These approaches often require large training datasets, manually defined recognition rules, or task-specific schemas. Their recognition capabilities are usually limited to classification, segmentation, or rule-based identification of machining features. Although effective within narrow domains, such systems struggle to handle new manufacturing features without major retraining or customization.

In contrast, the proposed method adopts a prompt-based vision-language embedding strategy that works on single- or multiview CAD images and eliminates the need for task-specific training or explicit geometric modeling. This allows for broader feature recognition across multiple manufacturing domains (including machining, molding, casting, sheet metal, and additive manufacturing) with minimal supervision. Table 2 illustrates the shift from specialized, geometry-focused recognition methods to a more general, adaptable approach enabled by foundation models.

The proposed VLM-based AFR method provides a general-purpose, prompt-driven solution that supports both machining and nonmachining domains without the need for task-specific training or geometric decomposition. Compared to previous AFR methods, which tend to be narrowly focused, data-heavy, or designed for specific domains, the proposed approach simplifies the recognition pipeline, reduces manual effort, and offers strong performance with much broader applicability.

Manufacturing Feature Recognition on Difficult-Level CAD Images

	Example 1	Example 2	Example 3
Isometric View			
Additional Views			
Ground Truth (by human expert)	• Hole (through/blind): 119 • Slot (through/blind): 4 • Step (through/blind): 1 • Sheet metal: 1	• Hole (through/blind): 11 • Step (through/blind): 1 • Fillet/Round: 4 • Sheet metal: 1	• Hole (through/blind): 8 • Sheet metal: 1 • Pipe/Tube: 1 • Fillet/Round: 2
Claude-3.5 Exp 3: Zero-shot, Multi-view	✗ Hole (through/blind): 114 ✓ Slot (through/blind): 4 ✓ Step (through/blind): 1 ✓ Sheet metal: 1	✗ Hole (through/blind): 10 ✓ Step (through/blind): 1 ✓ Fillet/Round: 4 ✗ Sheet metal: 0	✓ Hole (through/blind): 8 ✗ Sheet metal: 0 ✓ Pipe/Tube: 1 ✓ Fillet/Round: 2 ⚠ Boss: 2
GPT-4o Exp 5: Zero-shot, Multi-view, CoT	✗ Hole (through/blind): 2 ✗ Slot (through/blind): 2 ✗ Step (through/blind): 0 ✓ Sheet metal: 1 ⚠ Freeform features: 4	✗ Hole (through/blind): 10 ✗ Step (through/blind): 0 ✗ Fillet/Round: 8 ✗ Sheet metal: 0 ⚠ Rib: 8	✓ Hole (through/blind): 8 ✗ Sheet metal: 0 ✓ Pipe/Tube: 1 ✓ Fillet/Round: 2
Claude-3 Exp 4: Few-shot, Multi-view	✗ Hole (through/blind): 4 ✗ Slot (through/blind): 0 ✗ Step (through/blind): 0 ✗ Sheet metal: 0	✗ Hole (through/blind): 4 ✗ Step (through/blind): 0 ✗ Fillet/Round: 8 ✗ Sheet metal: 0	✗ Hole (through/blind): 2 ✗ Sheet metal: 0 ✗ Pipe/Tube: 0 ✗ Fillet/Round: 4
MiniCPM-Llama3 Exp 5: Zero-shot, Multi-view, CoT	✗ Hole (through/blind): 4 ✗ Slot (through/blind): 0 ✗ Step (through/blind): 0 ✓ Sheet metal: 1 ⚠ Chamfer/Bevel: 1	✗ Hole (through/blind): 4 ✓ Step (through/blind): 1 ✗ Fillet/Round: 0 ✗ Sheet metal: 0 ⚠ Chamfer/Bevel: 2	✗ Hole (through/blind): 4 ✗ Sheet metal: 0 ✗ Pipe/Tube: 0 ✗ Fillet/Round: 1 ⚠ Step (through/blind): 1 ⚠ Chamfer/Bevel: 2 ⚠ Freeform Features: 1
Llava-1.6-mistral-7b Exp 5: Zero-shot, Multi-view, CoT	✗ Hole (through/blind): 2 ✗ Slot (through/blind): 1 ✓ Step (through/blind): 1 ✗ Sheet metal: 0 ⚠ Pocket (through/blind): 1 ⚠ Chamfer/Bevel: 1 ⚠ Fillet/Round: 1	✗ Hole (through/blind): 0 ✗ Step (through/blind): 0 ✗ Fillet/Round: 1 ✗ Sheet metal: 0 ⚠ Slot (through/blind): 1 ⚠ Pocket (through/blind): 1 ⚠ Chamfer/Bevel: 1	✗ Hole (through/blind): 1 ✗ Sheet metal: 0 ✗ Pipe/Tube: 0 ✗ Fillet/Round: 1 ⚠ Boss: 1

Note: ✓ *represents correct prediction;* ✗ *represents false prediction or misprediction;* ⚠ *represents hallucination (extra feature not in ground truth)*

Fig. 17 Examples of ground truth and VLM-generated answers for difficult-level CAD. Claude-3.5 shows exceptional accuracy, closely matching high ground truth counts in certain designs.

5 Practical Implications, Limitations, and Future Work

The proposed framework demonstrates the effectiveness of VLMs in benchmarking AFR based on feature type and count. However, it currently lacks the ability to extract the spatial or geometric localization of features. This limitation arises from the image-text modality of VLMs, which restricts access to feature dimensions, positional coordinates, and orientations. The absence of such capabilities may hinder downstream tasks such as process planning, toolpath generation, and cost estimation—

Table 2 Comparison of prior AFR methods with the proposed VLM-based framework

AFR model (years)	Modality and representation	Data dependency	Recognition method	Generalizability
FeatureNet (2018) [9]	3D voxel grid and 3D CNN	24,000 CAD models (moderate)	Segmentation and classification	Moderate—limited to predefined 24 machining features only
MsvNet (2020) [8]	Multiview CAD 2D slices and CNN	24,000 STL files (moderate)	Segmentation and classification	Moderate—handles only 24 machining features via view aggregation
MBD and Process Semantics (2022) [40]	B-Rep with semantics and annotated B-Rep	None	Classification using process metadata	Moderate—limited to machining features only
Hierarchical CADNet (2022) [10]	Hierarchical B-Rep and dual-graph CNN	59,655 CAD models (high)	Classification and feature face label identification	Moderate—limited to 24 machining features; adaptable with retraining
DeepFeature (2022) [3]	B-Rep and attributed adjacency graph (AAG) GNN	3000 CAD models (moderate)	Decomposition and classification	Low—focused on only eight machining feature types
Hybrid graph and rule-based AFR (2023) [90]	B-Rep with feature traces and GNN	None	Rule-based with GNN	Low—specific to only 8 types of 3-axis milling features
AAGNet (2024) [7]	B-Rep graph AAG and geometric GNN	60,000 CAD models (high)	Semantic and instance segmentation	Moderate—limited to predefined 24 machining features only
CAFR (Surface GNN) (2024) [91]	B-Rep with surface graph and GNN	24,000 CAD models (moderate)	Feature classification	Moderate—designed for machining surface models only
Proposed method	Single/multiview image with text and vision-language embedding	No training dataset needed	Feature classification and quantification	High—prompt-based, generalizable across manufacturing domains

particularly when precise spatial context is required. Nevertheless, type and count information alone already offers significant actionable insights. For instance, identifying the number of drilled holes or milled slots directly informs tool selection and supports early-stage manufacturing planning. This level of semantic abstraction provides practical value across many decision-making scenarios. Consistent with this scope, most existing AFR approaches [3,5,7–11,16,40], including the present work, emphasize feature classification and quantification, with limited attention to spatial localization. To improve practical applicability, future research will explore the integration of spatial and dimensional attributes, leveraging annotations from corresponding 2D engineering drawings to enable geometry-aware reasoning in downstream applications.

Another limitation is that current VLMs are not designed to directly interpret native 3D CAD formats for AFR. While VLMs can process 2D images and text descriptions, they do not natively utilize the rich topological and geometric structure present in point clouds, meshes, or B-Reps. Consequently, spatial context is lost, especially in cases involving intersecting features or depth-sensitive geometries. Nevertheless, multiview 2D CAD projections convey substantial visual and spatial cues, aligning with how human engineers interpret geometric features from technical drawings. These limitations likely stem more from the absence of domain-specific fine-tuning than from inherent flaws in the representational format. For example, Claude-3.5-Sonnet performs comparably or better than nonexpert humans on hard difficulty samples. The proposed framework also supports a broader range of feature types across various manufacturing domains, beyond machining-specific primitives. Future work will investigate hybrid pipelines that combine VLM semantic reasoning with geometry-aware 3D models for improved robustness.

An additional limitation involves reasoning strategy. The current CoT implementation is restricted to single-turn inference. Multiturn reasoning is not pursued due to two key constraints: (1) token usage per high-resolution image can exceed 1000–1600 tokens [92], and with three views plus prompt text, the total approaches context limits for VLMs; and (2) runtime latency is significant, with even single-turn inference requiring several minutes for multiview input processing. Introducing iterative rounds would compound latency and system complexity. Reducing image resolution to save tokens is not viable, as feature-level detail is lost. Future work will investigate scalable multistep reasoning to balance resolution fidelity with computational efficiency.

6 Conclusions

This study demonstrated the potential of VLMs to automatically recognize a wide range of manufacturing features in CAD. By systematically applying prompt engineering strategies tailored for manufacturing, the proposed framework enhanced VLMs' ability to interpret complex visual and textual data across varying CAD complexities. A comprehensive evaluation revealed that Claude-3.5-Sonnet achieved the highest feature quantity accuracy (74%) and feature name matching accuracy (75%), while also maintaining the lowest MAE score (3.2). In contrast, GPT-4o exhibited the lowest hallucination rate (8%). However, the open-source models, Llava-v1.6-mistral-7b and MiniCPM-Llama3-V2.5, encountered significant challenges with higher hallucination rates (>30%) and lower accuracies (<40%).

The main contributions of this study included the development of a customized CAD dataset with varying feature complexities, the systematic implementation of prompt engineering strategies, and the introduction of tailored evaluation metrics to objectively assess model performance in AFR tasks. The findings illustrated the potential of these VLMs to handle complex geometrical data that often challenge human experts, thus offering a viable path toward automation in manufacturing design.

While the study effectively showcased the feasibility of VLMs in manufacturing feature recognition, several limitations highlighted potential directions for future research. The current CAD dataset, despite being categorized into three difficulty levels, did not fully capture the intricate diversity of real-world manufacturing scenarios. It primarily focused on isolated feature recognition and lacked complexities related to intersecting features and intricate assembly contexts. Future research will address this by expanding the dataset to better represent real-world conditions and uncover potential biases.

Another important limitation of the current approach was that the VLMs primarily recognized feature types and counts but lacked the ability to extract detailed spatial localization information, which is crucial for downstream manufacturing applications such as process planning, cost estimation, and quality control. To address these challenges, future research will aim to enrich the dataset with more realistic CAD scenarios and investigate methods for integrating spatial localization using annotations from corresponding 2D technical drawings.

An additional direction for future work involves enhancing reasoning strategies. The current implementation of prompt engineering, including CoT reasoning, was restricted to single-turn

inference. In this study, the CoT approach followed a three-step structure within a single prompt: visual identification, category reflection, and quantity estimation. Although this design improved intermediate reasoning, it did not support iterative refinement or contextual feedback. Future work will explore multiturn AFR workflows that decompose tasks across multiple inference rounds. These could incorporate dynamic prompting, memory mechanisms, and feedback loops to enable more adaptive reasoning. These strategies are particularly valuable for interpreting intersecting features, assembly-level designs, or ambiguous cases where context evolves across steps. By addressing these limitations, VLMs can be made more capable and reliable for complex automated manufacturing applications.

The code, CAD dataset, and additional prompt templates used in this study are available online.[4]

Acknowledgment

This work is supported by the Agency for Science, Technology and Research (A*STAR), Singapore, through the RIE2025 MTC IAF-PP grant (Grant No. M22K5a0045). It is also supported by the Singapore International Graduate Award (SINGA) (Awardee: Muhammad Tayyab Khan), funded by A*STAR and Nanyang Technological University, Singapore. Additional support was provided by an AcRF Tier 1 grant (RG70/23) from the Ministry of Education, Singapore, and Korea Planning & Evaluation Institute of Industrial Technology (KEIT) grant funded by the Korea government (MOTIE) (No. RS-2024-00442448).

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Elmesbahi, A., Buj-Corral, I., Mesbahi, J. E., and Bensaid, O., 2023, "New STEP-NC Compliant System to Automatic Process Planning for Turning Process," In Review.
[2] Basinger, K. L., Keough, C. B., Webster, C. E., Wysk, R. A., Martin, T. M., and Harrysson, O. L., 2018, "Development of a Modular Computer-Aided Process Planning (CAPP) System for Additive-Subtractive Hybrid Manufacturing of Pockets, Holes, and Flat Surfaces," Int. J. Adv. Manuf. Technol., **96**(5–8), pp. 2407–2420.
[3] Wang, P., Yang, W.-A., and You, Y., 2023, "A Hybrid Learning Framework for Manufacturing Feature Recognition Using Graph Neural Networks," J. Manuf. Processes, **85**, pp. 387–404.
[4] Heidari, N., and Iosifidis, A., 2024, "Geometric Deep Learning for Computer-Aided Design: A Survey," http://arxiv.org/abs/2402.17695, Accessed July 21, 2024
[5] Verma, A. K., and Rajotia, S., 2010, "A Review of Machining Feature Recognition Methodologies," Int. J. Comput. Integr. Manuf., **23**(4), pp. 353–368.
[6] Shi, Y., Zhang, Y., Xia, K., and Harik, R., 2020, "A Critical Review of Feature Recognition Techniques," Comput.-Aided Des. Appl., **17**(5), pp. 861–899.
[7] Wu, H., Lei, R., Peng, Y., and Gao, L., 2024, "AAGNet: A Graph Neural Network Towards Multi-Task Machining Feature Recognition," Rob. Comput. Integr. Manuf., **86**, p. 102661.
[8] Shi, P., Qi, Q., Qin, Y., Scott, P. J., and Jiang, X., 2020, "A Novel Learning-Based Feature Recognition Method Using Multiple Sectional View Representation," J. Intell. Manuf., **31**(5), pp. 1291–1309.
[9] Zhang, Z., Jaiswal, P., and Rai, R., 2018, "FeatureNet: Machining Feature Recognition Based on 3D Convolution Neural Network," Comput.-Aided Des., **101**, pp. 12–22.
[10] Colligan, A. R., Robinson, T. T., Nolan, D. C., Hua, Y., and Cao, W., 2022, "Hierarchical CADNet: Learning From B-Reps for Machining Feature Recognition," Comput.-Aided Des., **147**, p. 103226.
[11] Zhang, H., Zhang, S., Zhang, Y., Liang, J., and Wang, Z., 2022, "Machining Feature Recognition Based on a Novel Multi-Task Deep Learning Network," Rob. Comput. Integr. Manuf., **77**, p. 102369.
[12] Yeo, C., Kim, B. C., Cheon, S., Lee, J., and Mun, D., 2021, "Machining Feature Recognition Based on Deep Neural Networks to Support Tight Integration With 3D CAD Systems," Sci. Rep., **11**(1), p. 22147.
[13] Ning, F., Shi, Y., Cai, M., and Xu, W., 2023, "Part Machining Feature Recognition Based on a Deep Learning Method," J. Intell. Manuf., **34**(2), pp. 809–821.
[14] Sung, R. C. W., Corney, J. R., and Clark, D. E. R., 2001, "Automatic Assembly Feature Recognition and Disassembly Sequence Generation," ASME J. Comput. Inf. Sci. Eng., **1**(4), pp. 291–299.
[15] Khan, M. T., Feng, W., Chen, L., Ng, Y. H., Tan, N. Y. J., and Moon, S. K., 2024, "Automatic Feature Recognition and Dimensional Attributes Extraction From Cad Models for Hybrid Additive-Subtractive Manufacturing," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Washington, DC, Aug. 25–28.
[16] Babic, B., Nesic, N., and Miljkovic, Z., 2008, "A Review of Automated Feature Recognition With Rule-Based Pattern Recognition," Comput. Ind., **59**(4), pp. 321–337.
[17] Zhou, K., Yang, J., Loy, C. C., and Liu, Z., 2022, "Learning to Prompt for Vision-Language Models," Int. J. Comput Vision, **130**(9), pp. 2337–2348.
[18] Zhang, J., Huang, J., Jin, S., and Lu, S., 2024, "Vision-Language Models for Vision Tasks: A Survey," IEEE Trans. Pattern Anal. Mach. Intell., **46**(8), pp. 5625–5644.
[19] Picard, C., Edwards, K. M., Doris, A. C., Man, B., Giannone, G., Alam, M. F., and Ahmed, F., 2025, "From Concept to Manufacturing: Evaluating Vision-Language Models for Engineering Design," Artif. Intell. Rev., **58**(9), p. 288.
[20] Zhao, W., Wu, J., and Meng, Z., 2025, "AppPoet: Large Language Model Based Android Malware Detection via Multi-View Prompt Engineering," Expert Syst. Appl., **262**, p. 125546.
[21] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., et al., 2020, "Language Models Are Few-Shot Learners," Adv. Neural Inf. Process. Syst., **33**, pp. 1877–1901.
[22] Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., Li, Y., et al., 2025, "The Prompt Report: A Systematic Survey of Prompt Engineering Techniques,"
[23] Gu, J., Han, Z., Chen, S., Beirami, A., He, B., Zhang, G., Liao, R., Qin, Y., Tresp, V., and Torr, P., 2023, "A Systematic Survey of Prompt Engineering on Vision-Language Foundation Models,"
[24] "Hello GPT-4o," https://openai.com/index/hello-gpt-4o/, Accessed July 7, 2025.
[25] "Introducing Claude 3.5 Sonnet," https://www.anthropic.com/news/claude-3-5-sonnet, Accessed August 1, 2024.
[26] "Introducing the Next Generation of Claude," https://www.anthropic.com/news/claude-3-family, Accessed August 1, 2024.
[27] Hu, S., Tu, Y., Han, X., He, C., Cui, G., Long, X., Zheng, Z., et al., 2024, "MiniCPM: Unveiling the Potential of Small Language Models With Scalable Training Strategies,"
[28] "Liuhaotian/Llava-v1.6-Mistral-7b · Hugging Face," https://huggingface.co/liuhaotian/llava-v1.6-mistral-7b, Accessed August 1, 2024.
[29] Shi, Y., Zhang, Y., and Harik, R., 2020, "Manufacturing Feature Recognition With a 2D Convolutional Neural Network," CIRP J. Manuf. Sci. Technol., **30**, pp. 36–57.
[30] Regli, W. C., Gupta, S. K., and Nau, D. S., 1994, "Feature Recognition for Manufacturability Analysis," ASME 1994 International Computers in Engineering Conference and Exhibition, Minneapolis, MN, Sept. 11–14, American Society of Mechanical Engineers, pp. 93–104.
[31] Shi, Y., Zhang, Y., Baek, S., De Backer, W., and Harik, R., 2018, "Manufacturability Analysis for Additive Manufacturing Using a Novel Feature Recognition Technique," Comput.-Aided Des. Appl., **15**(6), pp. 941–952.
[32] Xú, S., Anwer, N., Mehdi-Souzani, C., Harik, R., and Qiao, L., 2016, "STEP-NC Based Reverse Engineering of in-Process Model of NC Simulation," Int. J. Adv. Manuf. Technol., **86**(9–12), pp. 3267–3288.
[33] Yildiz, A. R., Öztürk, N., Kaya, N., and Öztürk, F., 2003, "Integrated Optimal Topology Design and Shape Optimization Using Neural Networks," Struct. Multidiscipl. Optim., **25**(4), pp. 251–260.
[34] Chandrasegaran, S. K., Ramani, K., Sriram, R. D., Horváth, I., Bernard, A., Harik, R. F., and Gao, W., 2013, "The Evolution, Challenges, and Future of Knowledge Representation in Product Design Systems," Comput.-Aided Des., **45**(2), pp. 204–228.
[35] Harik, R., Capponi, V., Lombard, M., and Ris, G., 2006, "Enhanced Functions Supporting Process Planning for Aircraft Structural Parts," The Proceedings of the Multiconference on "Computational Engineering in Systems Applications.", Beijing, China, Oct. 4–6, IEEE, pp. 1259–1266.
[36] Harik, R. F., Derigent, W. J. E., and Ris, G., 2008, "Computer Aided Process Planning in Aircraft Manufacturing," Comput.-Aided Des. Appl., **5**(6), pp. 953–962.
[37] Harik, R. F., and Sahmrani, N., 2010, "DFMA+, A Quantitative DFMA Methodology," Comput.-Aided Des. Appl., **7**(5), pp. 701–709.
[38] Dimov, S. S., Brousseau, E. B., and Setchi, R., 2007, "A Hybrid Method for Feature Recognition in Computer-Aided Design Models," Proc. Inst. Mech. Eng. B, **221**(1), pp. 79–96.
[39] Yan, H., Yan, C., Yan, P., Hu, Y., and Liu, S., 2022, "Manufacturing Feature Recognition Method Based on Graph and Minimum Non-Intersection Feature Volume Suppression," In Review.

[4]https://github.com/Davidlequnchen/VLM-CADFeatureRecognition

[40] Xu, T., Li, J., and Chen, Z., 2022, "Automatic Machining Feature Recognition Based on MBD and Process Semantics," Comput. Ind., **142**, p. 103736.
[41] Wang, Q., and Yu, X., 2014, "Ontology Based Automatic Feature Recognition Framework," Comput. Ind., **65**(7), pp. 1041–1052.
[42] Zehtaban, L., and Roller, D., 2016, "Automated Rule-Based System for Opitz Feature Recognition and Code Generation From STEP," Comput.-Aided Des. Appl., **13**(3), pp. 309–319.
[43] Cai, N., Bendjebla, S., Lavernhe, S., Mehdi-Souzani, C., and Anwer, N., 2018, "Freeform Machining Feature Recognition With Manufacturability Analysis," Procedia CIRP, **72**, pp. 1475–1480.
[44] Campana, G., and Mele, M., 2020, "An Application to Stereolithography of a Feature Recognition Algorithm for Manufacturability Evaluation," J. Intell. Manuf., **31**(1), pp. 199–214.
[45] Ding, S., Feng, Q., Sun, Z., and Ma, F., 2021, "MBD Based 3D CAD Model Automatic Feature Recognition and Similarity Evaluation," IEEE Access, **9**, pp. 150403–150425.
[46] Colligan, A. R., 2022, "Deep Learning for Boundary Representation CAD Models," Doctoral thesis. Northern Ireland.
[47] Joshi, S., and Chang, T.-C., 1988, "Graph-Based Heuristics for Recognition of Machined Features From a 3D Solid Model," Comput.-Aided Des., **20**(2), pp. 58–66.
[48] Li, Z., Lin, J., Zhou, D., Zhang, B., and Li, H., 2018, "Local Symmetry Based Hint Extraction of B-Rep Model for Machining Feature Recognition," International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Quebec City, Quebec, Canada, Aug. 26–29.
[49] Li, H., Huang, Y., Sun, Y., and Chen, L., 2015, "Hint-Based Generic Shape Feature Recognition From Three-Dimensional B-Rep Models," Adv. Mech. Eng., **7**(4), p. 168781401558208.
[50] Woo, Y., and Sakurai, H., 2002, "Recognition of Maximal Features by Volume Decomposition," Comput.-Aided Des., **34**(3), pp. 195–207.
[51] Shi, Y., Deng, Z., and Zhong, J., 2018, "Recent Research and Prospect on Feature Recognition of Three-Dimensional Model," Third International Conference on Control, Automation and Artificial Intelligence (CAAI 2018), Beijing, China, Aug. 26–27.
[52] Sridharan, N., and Shah, J. J., 2005, "Recognition of Multi-Axis Milling Features: Part II—Algorithms & Implementation," ASME J. Comput. Inf. Sci. Eng., **5**(1), pp. 25–34.
[53] Verma, A. K., and Rajotia, S., 2009, "A Hybrid Machining Feature Recognition System," Int. J. Manuf. Res., **4**(3), p. 343.
[54] Prabhakar, S., and Henderson, M. R., 1992, "Automatic Form-Feature Recognition Using Neural-Network-Based Techniques on Boundary Representations of Solid Models," Comput.-Aided Des., **24**(7), pp. 381–393.
[55] Zhang, Y., Zhang, Y., He, K., Li, D., Xu, X., and Gong, Y., 2022, "Intelligent Feature Recognition for STEP-NC-Compliant Manufacturing Based on Artificial Bee Colony Algorithm and Back Propagation Neural Network," J. Manuf. Syst., **62**, pp. 792–799.
[56] Shi, P., Qi, Q., Qin, Y., Scott, P. J., and Jiang, X., 2022, "Highly Interacting Machining Feature Recognition via Small Sample Learning," Rob. Comput. Integr. Manuf., **73**, p. 102260.
[57] Felzenszwalb, P. F., and Huttenlocher, D. P., 2004, "Efficient Graph-Based Image Segmentation," Int. J. Comput. Vision, **59**(2), pp. 167–181.
[58] Shafarenko, L., Petrou, M., and Kittler, J., 1997, "Automatic Watershed Segmentation of Randomly Textured Color Images," IEEE Trans. Image Process., **6**(11), pp. 1530–1544.
[59] Shi, P., Qi, Q., Qin, Y., Scott, P. J., and Jiang, X., 2020, "Intersecting Machining Feature Localization and Recognition via Single Shot Multibox Detector," IEEE Trans. Ind. Inf., **17**(5), pp. 3292–3302.
[60] Ibrahim, A. D., Hussein, H. M. A., and Abdelwahab, S. A., 2021, "Automatic Feature Recognition of Cross Holes in Hollow Cylinders," J. Inst. Eng. (India): C, **102**(2), pp. 257–274.
[61] Ibrahim, A. D., Abdelwahab, S. A., Hussein, H. M. A., and Ahmed, I., 2019, "Review of Automatic Feature Recognition of Cylindrical Parts," Int. Res. J. Eng. Technol., **06**(12), pp. 1219–1233.
[62] Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., et al., 2023, "ChatGPT for Good? On Opportunities and Challenges of Large Language Models for Education," Learn. Individ. Differ., **103**, p. 102274.
[63] Gao, J., and Lin, C.-Y., 2004, "Introduction to the Special Issue on Statistical Language Modeling," ACM Trans. Asian Lang. Inf. Process., **3**(2), pp. 87–93.
[64] Meltzer, P., Lambourne, J. G., and Grandi, D., 2023, "What's in a Name? Evaluating Assembly-Part Semantic Knowledge in Language Models Through User-Provided Names in Computer Aided Design Files," ASME J. Comput. Inf. Sci. Eng., **24**(1), p. 011002.
[65] Grandi, D., Jain, Y. P., Groom, A., Cramer, B., and McComb, C., 2024, "Evaluating Large Language Models for Material Selection," ASME J. Comput. Inf. Sci. Eng., **25**(2), p. 021004.
[66] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., et al., 2023, "LLaMA: Open and Efficient Foundation Language Models,"
[67] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł, and Polosukhin, I., 2017, "Attention Is All You Need," Adv. Neural Inf. Process. Syst., **30**, pp. 5998–6008.
[68] Huang, Y., Xu, J., Lai, J., Jiang, Z., Chen, T., Li, Z., Yao, Y., et al., 2024, "Advancing Transformer Architecture in Long-Context Large Language Models: A Comprehensive Survey,"
[69] Naghavi Khanghah, K., Wang, Z., and Xu, H., 2024, "Reconstruction and Generation of Porous Metamaterial Units via Variational Graph Autoencoder and Large Language Model," ASME J. Comput. Inf. Sci. Eng., **25**(1), pp. 1–26.
[70] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I., 2019, "Language Models Are Unsupervised Multitask Learners," OpenAI blog, **1**(8).
[71] Manyika, J., and Hsiao, S., 2023, "An Overview of Bard: An Early Experiment With Generative AI," AI. Google Static Documents, 2.
[72] Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., et al., 2023, "Sparks of Artificial General Intelligence: Early Experiments With GPT-4," https://www.microsoft.com/en-us/research/publication/sparks-of-artificial-general-intelligence-early-experiments-with-gpt-4/, Accessed July 6, 2025
[73] McGee, D., 2000, "On Line and on Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering," https://muse.jhu.edu/pub/1/article/33532/summary?casa_token=zBAUAPRk7lgAAAAA:nudu8U3J7IhEHujFEeEVQSC8JJGMNBM8Sxq_ovvdNVJSe9CiBKsMy1XI_gmt_ctBWxPhQDimvVo, Accessed August 3, 2024
[74] Song, B., Zhou, R., and Ahmed, F., 2023, "Multi-Modal Machine Learning in Engineering Design: A Review and Future Directions," ASME J. Comput. Inf. Sci. Eng., **24**(1), p. 010801.
[75] Patashnik, O., Wu, Z., Shechtman, E., Cohen-Or, D., and Lischinski, D., 2021, "StyleCLIP: Text-Driven Manipulation of StyleGAN Imagery," IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, Oct. 10–17.
[76] Kim, G., Kwon, T., and Ye, J. C., 2022, "DiffusionCLIP: Text-Guided Diffusion Models for Robust Image Manipulation," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, June 18–24.
[77] Ma, Y., Xu, G., Sun, X., Yan, M., Zhang, J., and Ji, R., 2022, "X-CLIP: End-to-End Multi-Grained Contrastive Learning for Video-Text Retrieval," Proceedings of the 30th ACM International Conference on Multimedia, Association for Computing Machinery, New York, Oct. 10–14, pp. 638–647.
[78] Bar-Tal, O., Ofri-Amar, D., Fridman, R., Kasten, Y., and Dekel, T., 2022, "Text2LIVE: Text-Driven Layered Image and Video Editing," Computer Vision—ECCV *2022*, S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, eds., Springer Nature, Cham, Switzerland, pp. 707–723.
[79] Hong, F., Zhang, M., Pan, L., Cai, Z., Yang, L., and Liu, Z., 2022, "AvatarCLIP: Zero-Shot Text-Driven Generation and Animation of 3D Avatars,"
[80] Aneja, S., Thies, J., Dai, A., and Nießner, M., 2023, "ClipFace: Text-Guided Editing of Textured 3D Morphable Models," Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Proceedings, Los Angeles, CA, Aug. 6–10, pp. 1–11.
[81] Michel, O., Bar-On, R., Liu, R., Benaim, S., and Hanocka, R., 2022, "Text2Mesh: Text-Driven Neural Stylization for Meshes," IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, June 18–24.
[82] Paviot, T., 2022, "PythonOCC-Core: Python package for 3D geometry CAD/BIM/CAM ," Zenodo.
[83] "Hello GPT-4o | OpenAI," https://openai.com/index/hello-gpt-4o/, Accessed September 8, 2024.
[84] "Models—Anthropic," https://docs.anthropic.com/en/docs/about-claude/models, Accessed September 8, 2024.
[85] "Openbmb/MiniCPM-Llama3-V-2_5 · Hugging Face," https://huggingface.co/openbmb/MiniCPM-Llama3-V-2_5, Accessed August 1, 2024.
[86] Lee, H. L., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Jae, Y., 2024, "LLAVA-NeXT: Improved Reasoning, OCR, and World Knowledge," LLaVA. https://llava-vl.github.io/blog/2024-01-30-llava-next/, Accessed July 20, 2024.
[87] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. V., and Zhou, D., 2022, "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," Adv. Neural Inf. Process. Syst., **35**, pp. 24824–24837.
[88] Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T., Cao, Y., and Narasimhan, K., 2023, "Tree of Thoughts: Deliberate Problem Solving With Large Language Models," Adv. Neural Inf. Process. Syst., **36**, pp. 11809–11822.
[89] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S., 2023, "Reflexion: Language Agents With Verbal Reinforcement Learning," Adv. Neural Inf. Process. Syst., **36**, pp. 8634–8652.
[90] Wang, P., Liu, W., and You, Y., 2023, "A Hybrid Framework for Manufacturing Feature Recognition From CAD Models of 3-Axis Milling Parts," Adv. Eng. Inform., **57**, p. 102073.
[91] Böhm, S. A., Zagar, B. L., Riß, F., Kortüm, C., and Knoll, A. C., 2024, "Automated Feature Recognition in Surface Cad Models Based on Graph Neural Networks,"
[92] "Vision," Anthropic. https://docs.anthropic.com/en/docs/build-with-claude/vision, Accessed May 5, 2025.