Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Assessing the Effects of Design Complexity and Prompt Engineering on CAD Model Generation Using Microsoft Copilot

Rogelio Lara
Wayne State University
Hector Molina
Wayne State University
Tazdik Patwary Plateau
ptazdik@wayne.edu
Wayne State University

Research Article

Keywords: Artificial Intelligence, Capstone Design Courses, Computer-Aided Design, Design Automation, Generative Design

Posted Date: November 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11493537/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Assessing the Effects of Design Complexity and Prompt Engineering on CAD Model Generation Using Microsoft Copilot

Authors: Rogelio Lara [1, +], Hector Molina [1, +], Tazdik Patwary Plateau [1, *]

[1]Engineering Technology, James and Patricia College of Engineering, Wayne State University, 4855 Fourth St., Detroit, Michigan 48201

*Corresponding Author: Tazdik Patwary Plateau, ptazdik@wayne.edu

[+]Equal Contribution and Efforts

Keywords: Artificial Intelligence, Capstone Design Courses, Computer-Aided Design, Design Automation, Generative Design

Abstract

The use of generative artificial intelligence (AI) is rapidly increasing across various sectors, making tasks more accessible, efficient, and user-friendly. One emerging application is computer-aided design (CAD), where AI has the potential to enable users with little or no-prior CAD experience to create three-dimensional (3D) models through natural-language. Despite growing interest in AI-assisted design, the capability of publicly accessible generative AI systems to reliably generate CAD models across varying levels of design complexity remains insufficiently studied. This work benchmarks the performance of Microsoft Copilot for CAD model generation and investigates the influence of prompt engineering on model quality. A benchmark dataset comprising 60 design tasks was developed and categorized into easy, intermediate, and complex levels. The benchmark included primitive geometries, polygonal and polyhedral models, mechanical components, hollow structures, assemblies, lattice designs, and advanced engineering systems. For each prompt, the generated STL file was evaluated based on geometric accuracy, completeness of requested features, and overall structural validity using commercial software. The results revealed a strong dependence of generation success on design complexity, with easy-level tasks exhibiting high first-attempt success rates, while intermediate and complex tasks frequently required prompt refinement. Prompt length alone was found to have minimal influence on performance; instead, successful CAD generation depended primarily on clearly specified dimensions, geometric features, and functional relationships. Overall, the findings demonstrate that generative AI can lower the barrier to CAD modeling for non-expert users while improving productivity for experienced designers, thereby offering significant potential for engineering design, rapid prototyping, and additive manufacturing applications.

1 Introduction

Computer-aided design (CAD) serves as a foundational tool in engineering, manufacturing, product development, and rapid prototyping. Modern CAD software enables the creation of precise geometric models through dimensions, constraints, feature definitions, and assembly relationships [1,2]. However, developing proficiency in CAD requires a significant investment in training and practice. Beginners often struggle with sketch constraints, feature dependencies, and parametric modeling concepts, resulting in inaccurate or incomplete models. Even experienced CAD engineers must devote considerable time to creating and modifying components, particularly when generating repetitive features, parametric variants, and supporting parts for larger assemblies [3,4].

Recent advances in artificial intelligence (AI) have begun transforming digital design workflows [5–7]. The integration of AI with CAD and additive manufacturing has demonstrated the potential to reduce design time, accelerate iteration cycles, improve productivity, and enable the rapid exploration of design alternatives [8]. Unlike conventional CAD workflows that require extensive modeling expertise, generative AI systems allow users to describe objects using natural language, lowering the barrier to design creation and enabling broader participation in engineering and manufacturing activities.

Generative AI tools such as ChatGPT, Microsoft Copilot, Claude, and Grok have shown remarkable capabilities in interpreting natural-language instructions and generating text, images, computer code, and increasingly, three-dimensional (3D) images [9–12]. In the context of CAD generation, these systems provide an opportunity to transition from traditional geometry-based design toward language-driven design workflows. A user can describe a desired object using conversational language, and the AI system can generate a corresponding CAD model or STL file suitable for visualization and additive manufacturing. Such an approach has the potential to benefit both novice users with limited CAD experience and professional designers seeking to automate repetitive design tasks [9].

Despite these advances, the quality of AI-generated CAD models remains highly dependent on prompt formulation. Ambiguous, incomplete, or poorly structured prompts frequently result in missing features, dimensional inaccuracies, invalid geometries, or models that fail to capture the intended design. Consequently, prompt engineering has emerged as a critical technique for improving AI-generated outputs [13–17]. Prompt engineering involves the systematic refinement of user instructions to clearly communicate design intent, geometric constraints, feature requirements, and functional relationships. While substantial research has been devoted to prompt engineering for text generation and large language models, relatively little work has investigated how prompt design influences the generation of 3D CAD models and STL files [18].

Previous studies [8,19,20] have demonstrated that AI-assisted CAD systems perform reasonably well when generating simple geometries and moderately constrained parts. However, performance tends to deteriorate as design complexity increases, particularly for assemblies, mechanically constrained systems, lattice structures, and geometries containing multiple interacting components. Additional challenges have been reported in maintaining dimensional accuracy, preserving feature relationships, and generating manufacturable STL files without requiring extensive post-processing [21–23]. These limitations suggest that the effectiveness of AI-generated CAD models depends not only on the capabilities of the underlying AI model but also on the manner in which design requirements are communicated.

Although several specialized text-to-CAD systems [24–26] have recently been developed, many remain in the early stages of development, require significant setup effort, or are not easily accessible to the general public. Furthermore, these systems often assume some level of CAD knowledge, which may limit their adoption by casual users, students, and non-engineering stakeholders. In contrast, widely available general-purpose generative AI platforms are already integrated into many educational and professional environments, making them attractive candidates for AI-assisted CAD generation.

The objective of this study is to evaluate the capability of a publicly accessible, general-purpose AI platform, Microsoft Copilot, to generate 3D CAD models from natural-language prompts. Particular emphasis is placed on understanding how prompt engineering strategies influence model quality across varying levels of design complexity. Rather than focusing solely on the AI model itself, this work treats prompt structure as a controllable design variable and examines its effectiveness on geometric accuracy,

feature completeness, and overall STL file quality. The motivation behind this work stems from the growing interest in exploring how AI can simplify complex engineering tasks, particularly 3D design and CAD model generation. Many users require accurate STL files for visualization, prototyping, or manufacturing purposes but may lack the technical knowledge or experience of CAD software [3,27,28]. At the same time, CAD engineers often spend considerable time generating relatively simple or repetitive parts before assembling them into larger systems [29]. If AI can reliably create such parts from natural-language descriptions, it has the potential to reduce modeling time, improve design efficiency, and make CAD-based design more accessible to a broader group of users.

This study was conducted as part of a senior design project, where the student team used the university-authorized student version of Microsoft Copilot [12]. Copilot was selected because it is a widely used general-purpose generative AI platform that is accessible to students, educators, and non-expert users. Since this version represents the type of AI tool commonly available to general users, it provides a practical framework for evaluating real-world usability rather than the performance of a specialized or enterprise-level CAD system. The goal was to identify the types of 3D designs Copilot can generate effectively, the limitations it encounters, and the levels of design complexity that require prompt refinement or human intervention.

To achieve these objectives, a benchmark dataset consisting of 60 CAD-generation tasks was developed and categorized into easy, intermediate, and complex design levels. The generated 3D STL files were evaluated using geometric inspection and feature validation methods, while multiple prompt refinement strategies were investigated to determine their effectiveness in improving output quality. This research demonstrates the potential of generative AI to lower the barrier to entry for 3D modeling by enabling users with little or no CAD experience to generate functional and relatively accurate STL files through effective prompt engineering. In addition, the findings provide practical insights into the strengths and limitations of AI-assisted CAD generation and establish guidelines for improving the reliability of generative AI tools in engineering design, rapid prototyping, and additive manufacturing. Ultimately, such capabilities may make 3D design more accessible, faster, and more efficient across a wide range of engineering, educational, and industrial applications.

2 | Methodology

Our primary requirement for this project was to generate 3D CAD models using only AI generated prompts without manually creating models for the design phase. All the outputs that were produced were required to be exported as STL files that are suitable for 3D printing or visualization.

2.1 Selection of General-Purpose Generative AI tool. Microsoft Copilot was selected as the generative AI tool for this study because it is accessible to students, faculty, and staff at the university and is broadly used in academic environments. The version of Microsoft Copilot used in this work was not an enterprise-level or specialized CAD-integrated version; rather, it was the standard university-authorized student-accessible version. All CAD model generations were performed using the GPT-5 model available within Copilot. This version was intentionally selected because the objective of this study

was to evaluate the practical usability of a widely available generative AI tool for CAD model generation. Many students, educators, and general users may not have access to professional-level AI systems or specialized CAD automation tools. Therefore, using the student-accessible version of Microsoft Copilot allows this study to better represent the experience of typical users who may seek to generate STL files or simple CAD models through natural-language prompts without advanced CAD expertise.

2.2 Categorizing Prompt Questions. To evaluate the capability of Copilot in generating CAD models, a total of 60 prompt-questions were developed and categorized into three levels of complexity: easy, intermediate, and complex. These prompts were designed to assess the system's ability to generate CAD files for a wide range of geometric and engineering applications. The complete set of benchmark questions is provided in Supplemental Sections S1–S3. Representative examples from each complexity level, along with their corresponding design features, are presented in Table 1.

Table 1. Representative benchmark prompts and associated CAD generation capabilities from each complexity level.

Prompt Types	Features	Example
Easy	Consisted of basic geometry shapes with minimal features.	Create a sphere of 20 cm in diameter
Intermediate	Consisted of objects with multiple features, hollow sections or angled geometry.	Make a PVC pipe 90-degree elbow with 1 cm in thickness and 5cm in diameter
Complex	Consisted of multiple part assemblies or mechanically representative models.	Create 3 interlocking gears with the first one having 24 teeth, the second one having 18 teeth, the third having 12 teeth. Make sure the gear teeth fit perfectly in between the other gear teeth. Make the teeth trapezoid shaped

2.3 Selection Methodology of Questions. To understand the ability of Copilot GPT-5 to generate accurate 3D CAD models was evaluated through an iterative prompt-engineering workflow. Table 2 is showing a total of 60 benchmark tasks, categorized into easy, intermediate, and complex levels, were submitted to the chatbot. The complete list of benchmark questions for each complexity level is provided in the Supplemental Material, with easy-, intermediate-, and complex-level prompts presented in Tables S1, S2, and S3, respectively.

Table 2. Capability categories and representative benchmark questions used to assess Copilot's CAD generation capability. The benchmark dataset consisted of 60 prompts distributed across three complexity levels. E, I, and C represent Easy, Intermediate, and Complex benchmark questions, respectively.

Capability Category	Representative Shapes/Tasks	Question No
Primitive Shape Generation	Cube, Sphere, Cylinder, Hemisphere, Rectangular Prism	E3, E8, E11, E12, E15
Parametric Shape Generation	Cone, Torus, Pyramid, Hexagonal Prism, Funnel	E13, E17, E18, I16, I17

Polygonal and Polyhedral Geometry Generation	6-Sided Dice, 12-Sided Dice, Hexagon, Equilateral Triangle, Rhombus, Trapezoid	E2, E4, E5, E6, E9, E16
Organic and Freeform Shape Approximation	Football, Soccer Ball, Spheroid, Flower	E1, E7, E10, I8
Hollow Object Design	Hollow Cylinder, Hollow Cone, Hollow Box, Hollow Sphere	E14, I7, I11, I14
Boolean Feature Creation	Plate with Hole, Cube with Hole, Rounded Box with Through Hole, Mechanical Bracket	E19, E20, I1, I15
Mechanical Component Generation	Gear, Bolt, Bracket, PVC Elbow, Funnel	I5, I9, I15, I16, I19
Consumer Product Modeling	Coffee Mug, Ruler, Table, Storage Container	I6, I10, I12, I20
Architectural & Civil Structures	House, Dome Structure, Building, Bridge Truss	I2, I3, C3, C14
Assembly and Multi-Component Design	Table Assembly, Gear Assembly, Watch, Garbage Bin	I10, C1, C9, C10
Lattice and Cellular Structures	Honeycomb Wall, Lattice Cube, Honeycomb Panel	C5, C11, C17
Motion and Mechanism Design	Gear Assembly, Spiral Staircase, Cam-Follower, Scissor Lift	C1, C12, C16, C18
Curved and Revolved Geometry	Spheroid, Cylinder, Cone, Torus, Dome	E8, E10, E11, E13, E18, I3
Patterned and Repetitive Feature Generation	Honeycomb Wall, Perforated Hollow Block, Maze, Lattice Cube	C5, C7, C8, C11
Complex Assembly Design	Ferris Wheel, Watch, Garbage Bin, Parametric Gear System, Multi-Component Assembly	C4, C9, C10, C13, C20
Advanced Parametric Modeling	Interference-Fit Container, Parametric Gear System, Modular Honeycomb Panel, Constrained Assembly	I20, C13, C17, C20

2.4 Design Description and Design Alternatives. Initial prompt attempts frequently resulted in incomplete or inaccurate geometries, particularly for the intermediate and complex benchmark tasks. Common issues included missing holes, incorrect dimensions, incomplete features, distorted proportions, and misaligned components. These shortcomings were often attributed to insufficient design constraints or inadequate feature descriptions within the original prompt. To address these limitations, prompt engineering techniques were systematically applied [30,31]. The prompts were progressively refined by introducing additional dimensional constraints, detailed geometric descriptions, explicit feature requirements, and assembly relationships when applicable.

2.5 Prompt Engineering. To investigate the impact of prompt formulation on CAD generation performance, multiple prompting strategies [30,31] were evaluated, including increasing prompt length and instructing Copilot to generate its own detailed prompt. Each benchmark question was allowed a maximum of six attempts. Following an unsuccessful iteration, the prompt was refined by introducing additional geometric constraints, dimensions, feature descriptions, and assembly relationships to improve design specificity and clarity. If the generated model remained unsatisfactory after five refinement cycles, a sixth and final attempt was performed in which Copilot was instructed to automatically generate a more detailed prompt before producing the 3D model. This

iterative workflow enabled the assessment of the effectiveness of prompt engineering in improving CAD model quality and completeness.

2.6 Evaluation of AI-generated CAD File. The resulting STL file was evaluated according to three criteria: (i) geometric accuracy, (ii) completeness of the requested design features, and (iii) overall structural validity. Geometric inspection, dimensional verification, and feature validation were conducted using FreeCAD and Autodesk software. The evaluation focused on determining whether the generated model satisfied the prompt specifications, including dimensions, topology, and functional design features. Models exhibiting missing features, significant dimensional deviations, or invalid geometries were classified as partially successful or unsuccessful depending on the severity of the discrepancy.

2.7 Project Context. This research was conducted as part of a senior design project for completion of BS degree requirement. This study was conducted from February 2026 to May 2026. The project aimed to evaluate the feasibility of using a publicly accessible, general-purpose generative AI platform for CAD model generation and STL file creation. Under faculty supervision, the student team developed benchmark prompts, performed iterative prompt-engineering experiments, evaluated the generated STL files, and analyzed model performance across multiple design complexity levels. This project-based framework provided an educational context for assessing how students and non-expert users may interact with AI-assisted CAD tools. It also enabled the evaluation of Microsoft Copilot as a practical design-support tool for rapid prototyping, engineering education, and early-stage product development.

3 Results

The results of this study reveal several important trends and insights regarding AI-assisted CAD generation. The analysis focuses on key factors such as success rate as a function of design complexity, the role of prompt engineering in improving model quality, the effect of prompt length, and the time required to generate STL files across different complexity levels. The study also examines the effectiveness of Copilot in generating various geometric features and CAD capability categories. To further quantify task difficulty, a difficulty index was calculated based on the number of prompt attempts required to achieve successful model generation. The complete set of generated CAD outputs is provided in the Supplemental Material, as shown in Figure S1. Finally, a V8 engine model was used as an extreme benchmarking case to evaluate the capability and limitations of Copilot in generating highly complex mechanical designs.

3.1 Success dependence on the Question Complexity. The success rate of AI-generated CAD models was strongly influenced by the complexity of the benchmark question. Figure 1 summarizes the percentage of designs successfully generated at each prompt attempt for the easy-, intermediate-, and complex-level benchmark datasets. For the easy-level questions (Figure 1a), Copilot demonstrated a high first-attempt success rate, with 85% of the designs generated successfully from the initial prompt. The remaining 15% required a second prompt iteration to achieve the desired geometry. No

easy-level question required more than two attempts, indicating that Copilot is highly capable of generating simple engineering features with minimal prompt refinement. The performance declined as question complexity increased. For the intermediate-level questions (Figure 1b), only 60% of the designs were successfully generated on the first attempt. An additional 20% were resolved after a second prompt iteration, while 10%, 5%, and 5% required a third, fourth, and fifth attempt, respectively. These results suggest that designs involving multiple features, assemblies, or geometric constraints benefit substantially from iterative prompt refinement. The most significant impact of prompt engineering was observed for the complex-level questions (Figure 1c). Only 20% of the benchmark designs were successfully generated during the first attempt. The highest success rate occurred during the second attempt (35%), followed by the third and fourth attempts (20% each). An additional 5% required five attempts before a satisfactory model was generated. The results indicate that for complex engineering designs involving assemblies, lattice structures, mechanisms, and highly constrained geometries, prompt refinement becomes essential to achieving accurate CAD generation. Overall, the results demonstrate a clear relationship between prompt complexity and the number of iterations required for successful CAD generation. While simple geometries can be generated reliably using a single prompt, increasingly complex designs require additional prompt engineering to provide sufficient geometric constraints, feature definitions, and assembly relationships. This highlights the importance of user-guided prompt refinement when employing generative AI tools for advanced CAD modeling applications.

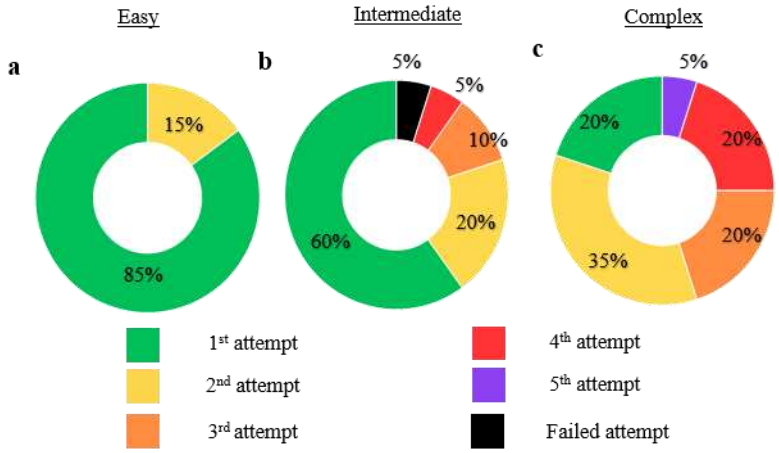


Figure 1. Percentage of benchmark questions successfully generated at each prompt attempt for (a) easy-level, (b) intermediate-level, and (c) complex-level CAD generation tasks. The results illustrate the increasing importance of prompt refinement as design complexity increases.

3.2 Prompt Length Impact. Alternative prompting strategies were also investigated, including increasing prompt length and instructing Copilot to generate its own detailed prompts. The results indicated that prompt length alone did not guarantee improved performance. Instead, successful CAD generation was primarily dependent on the inclusion of clearly defined design features, dimensional constraints, and geometric relationships. These findings highlight the importance of effective prompt engineering in improving the reliability and completeness of AI-generated CAD models.

For example, an initial prompt requesting a football, "*Make an STL file of a football with a 20 cm radius at the midsection*," generated only a simplified oval-shaped geometry (Figure 2a). In the second attempt, the prompt was refined to include key identifying features: "*Make an STL file of an American football with a 20 cm diameter at the midsection and include the laces and seam lines*." This minor addition resulted in a significantly more realistic and complete football model (Figure 2b). The improvement demonstrates that explicitly specifying distinctive geometric features is more important than merely increasing prompt length.

Similarly, the prompt "*Make an STL file of a 12-sided dice with the sides of 5 cm*" failed to produce a satisfactory model during the first attempt (Figure 2c). The ambiguity arose because the geometric nature of the 12-sided die was not clearly defined. When the prompt was revised to "*Make an STL file of a 12-sided dice with pentagons as the faces, where each side of the pentagon is 5 cm*," Copilot successfully generated the desired geometry (Figure 2d).

An important consideration during prompt refinement was maintaining accessibility for non-expert users. Rather than introducing highly technical CAD terminology, the revised prompts were written using simple, descriptive language that could reasonably be provided by a typical user with limited CAD experience. The successful generation of football and 12-sided die models suggests that straightforward feature descriptions can substantially improve output quality without requiring specialized CAD knowledge. These observations indicate that prompt clarity and feature specificity are the primary factors influencing successful AI-assisted CAD generation.

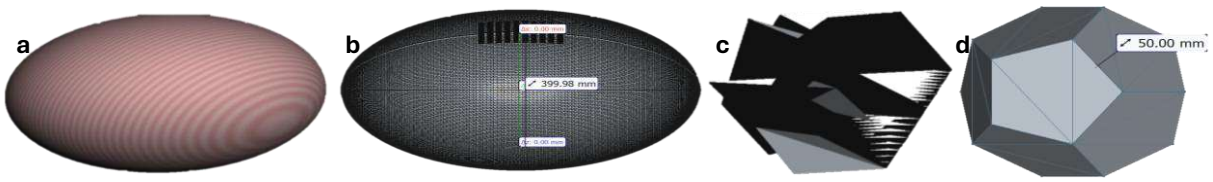


Figure 2. Examples illustrating the impact of prompt engineering on CAD generation performance. (a) Football model generated from the initial prompt, showing a simplified geometry; (b) football model generated after specifying laces and seam lines; (c) unsuccessful 12-sided dice generated from an ambiguous prompt; and (d) successful 12-sided dice generated after explicitly defining the pentagonal faces.

3.3 Prompt Engineering for CAD Designs. Prompt engineering played a critical role in improving the accuracy and completeness of AI-generated CAD models, particularly for complex assembly designs. To demonstrate this effect, a three-gear assembly benchmark problem was selected and iteratively refined through multiple prompt attempts. The generated STL files were obtained from Microsoft Copilot and subsequently visualized using Microsoft 3D Paint, FreeCAD and Autodesk software. Representative outputs from Attempts 1–3 are shown in Figure 3a–c.

The initial prompt specified the number of gears, tooth counts, diameters, and the requirement that the gears align and rotate properly. Although Copilot generated a

recognizable gear assembly, the resulting model achieved only 60% correctness because the tooth engagement and alignment were not fully resolved (Figure 3a). To improve the output, the second prompt explicitly stated that the gears should be arranged in a straight-line configuration and that the teeth should have a trapezoidal profile. This additional geometric information improved the generated model and increased the evaluation score to 80% (Figure 3b). In the third attempt, the prompt was further refined by explicitly specifying that the teeth of adjacent gears must properly mesh and align with one another. This additional constraint provided Copilot with a clearer description of the functional relationship between the gear components. As a result, the generated assembly exhibited correct tooth engagement, alignment, and overall geometry, achieving a 100% correctness evaluation (Figure 3c). These results demonstrate that successful AI-generated CAD design depends less on prompt length and more on the inclusion of precise geometric constraints, design intent, and functional relationships between components. Even relatively small modifications to the prompt, expressed in straightforward natural language, can substantially improve model quality. This finding is particularly important for novice users, as it suggests that effective CAD generation can be achieved without requiring advanced CAD terminology, provided that critical design features and relationships are clearly described.

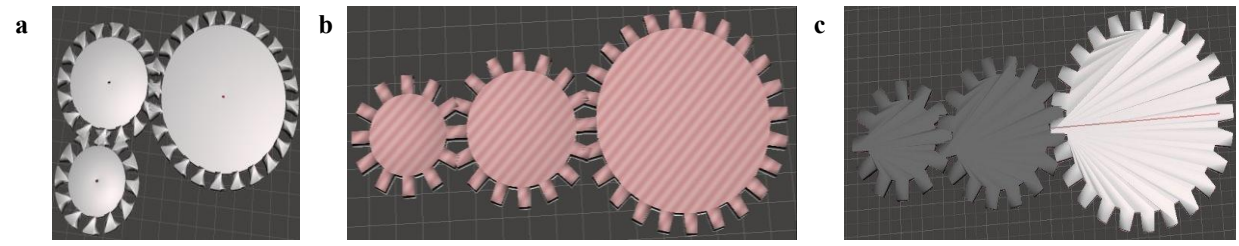


Figure 3. Effect of prompt engineering on the generation of a three-gear assembly: (a) initial prompt output (60% correctness), (b) refined prompt with gear arrangement and tooth-shape constraints (80% correctness), and (c) final prompt including explicit tooth meshing and alignment requirements (100% correctness).

3.4 Time Duration for Generating CAD file. The time required for Copilot to generate STL files increased with the complexity of the design task. For all benchmark evaluations, the same computing platform, Microsoft Copilot version, and AI model were used to ensure consistency. Generation time was measured manually using a stopwatch, starting from the moment the prompt was submitted until the STL file became available in the Copilot interface.

Figure 4 compares the shortest, average, and longest generation times for the easy-, intermediate-, and complex-level benchmark questions. The easy-level prompts exhibited the fastest generation times, with a minimum of 17 s, an average of 51 s, and a maximum of 158 s. For intermediate-level prompts, the minimum, average, and maximum generation times increased to 20 s, 113 s, and 338 s, respectively. The complex-level prompts required the longest processing times, ranging from 36 s to 393 s, with an average generation time of 121 s. The results indicate a direct relationship between design complexity and CAD generation time. As the number of geometric features, constraints, assemblies, and design relationships increased, Copilot required additional computational time to interpret the prompt and generate a valid STL model. While the

minimum generation times remained relatively low across all complexity levels, the maximum generation times increased substantially for complex benchmark tasks, suggesting that highly detailed or multi-component designs impose a significantly greater computational burden on the AI-assisted CAD generation workflow. Overall, the results demonstrate that AI-generated CAD creation is feasible within a practical timeframe for a wide range of engineering tasks, however, users should expect longer generation times when creating complex structures and geometries.

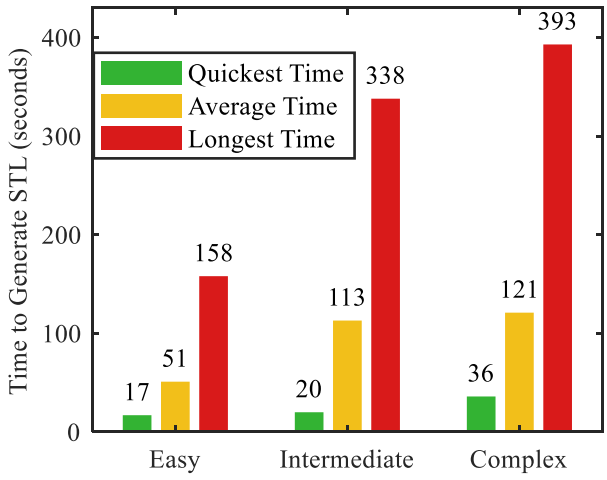


Figure 4. Comparison of the shortest, average, and longest STL generation times for easy-, intermediate-, and complex-level benchmark questions submitted to Microsoft Copilot. The results show an increasing time requirement with increasing design complexity.

3.5 Capability-Dependent Prompt Refinement and Difficulty Index Analysis. Figure 5 summarizes the relationship among CAD capability category, prompt-attempt success, and overall generation difficulty. In the left panel, each circular marker represents the percentage of successful STL generations for a specific CAD capability category at a given prompt attempt. The marker color corresponds to the success percentage according to the colorbar shown on the far left. Categories with high-percentage markers concentrated in the first attempt indicate that Copilot successfully generated those CAD models with minimal prompt refinement, while markers distributed across later attempts or the failed category indicate greater reliance on iterative prompt engineering.

The left panel shows that Copilot performed most effectively for well-defined and geometrically constrained categories. For example, parametric shape generation and curved-revolved geometry achieved 100% success in the first attempt, indicating strong capability in generating standard geometries such as cones, torus structures, pyramids, spheres, spheroids, cylinders, and related revolved shapes. Similarly, primitive shape generation showed 80% success in the first attempt and 20% in the second attempt, demonstrating that basic objects such as cubes, spheres, cylinders, hemispheres, and rectangular prisms were generally generated with little refinement. Categories such as hollow object design and Boolean feature creation also showed strong performance, with 75% success in the first attempt and 25% in the second attempt, suggesting that Copilot

can reasonably interpret common CAD operations such as hollowing and through-hole creation when the prompt includes clear dimensional information.

In contrast, categories requiring more complex spatial reasoning, feature relationships, repeated structures, or component interactions demonstrated broader distributions across multiple attempts. For example, assembly and multi-component design required multiple refinements, with successful generations distributed across the first, third, fourth, and fifth attempts. Motion and mechanism design also required repeated prompting, with successes appearing across the first, second, and third attempts. The most challenging behavior was observed for complex assembly design, where no successful output was achieved in the first three attempts; instead, most success occurred in the fourth attempt, with additional outputs requiring a fifth attempt or remaining unsuccessful. These results indicate that while Copilot is effective for single-part and geometrically explicit CAD tasks, its performance decreases for designs requiring mechanical relationships, alignment, functional interaction, or complex assembly-level understanding. To quantify the relative difficulty of each CAD capability category, a weighted difficulty index was calculated using Equation (1):

$$DI = \frac{(P_1 \times 1)+(P_2 \times 2)+(P_3 \times 3)+(P_4 \times 4)+(P_5 \times 5)+(P_F \times 6)}{100} \quad \text{(Equation 1)}$$

Here DI is the difficulty index, P_1 to P_5 represent the percentage of successful generations at the first through fifth prompt attempts, respectively, and P_F represents the percentage of failed or unsuccessful generations. Higher weights were assigned to later prompt attempts and failed outputs because they indicate greater difficulty in generating the requested CAD model. Therefore, a lower DI corresponds to a category that Copilot can generate more easily, while a higher DI indicates that multiple prompt refinements or failures were involved.

The right panel of Figure 5 presents the calculated DI for each CAD capability category. Green bars indicate categories with a DI below 1.5, representing low-difficulty tasks that were generally completed within the first one or two prompt attempts. Yellow bars represent moderate-difficulty categories with DI between 1.5 and 3, where additional prompt refinement was commonly required. Red bars represent high-difficulty categories with DI greater than 3, indicating that Copilot struggled to generate these models accurately.

The results show that complex assembly design exhibited the highest difficulty, followed by categories such as assembly and multi-component design, motion and mechanism design, and advanced parametric modeling. These categories require not only geometric generation but also an understanding of component relationships, alignment, mechanical functionality, and design intent. In contrast, categories such as parametric shape generation, curved and revolved geometry, and primitive shape generation showed low DI, confirming that Copilot performs well when prompts involve simple, single-body CAD models with clearly defined dimensions.

Overall, this analysis demonstrates that Copilot's CAD-generation capability is strongly dependent on the type of geometric and functional information required in the prompt. The AI system performs reliably for basic and moderately constrained shapes,

but its performance decreases when the task involves assemblies, mechanisms, repetitive structures, or complex design relationships. These findings highlight the importance of prompt refinement and suggest that general-purpose AI tools are most effective for early-stage CAD generation, concept development, and simple part creation, while complex engineering designs still require human oversight and additional CAD refinement.

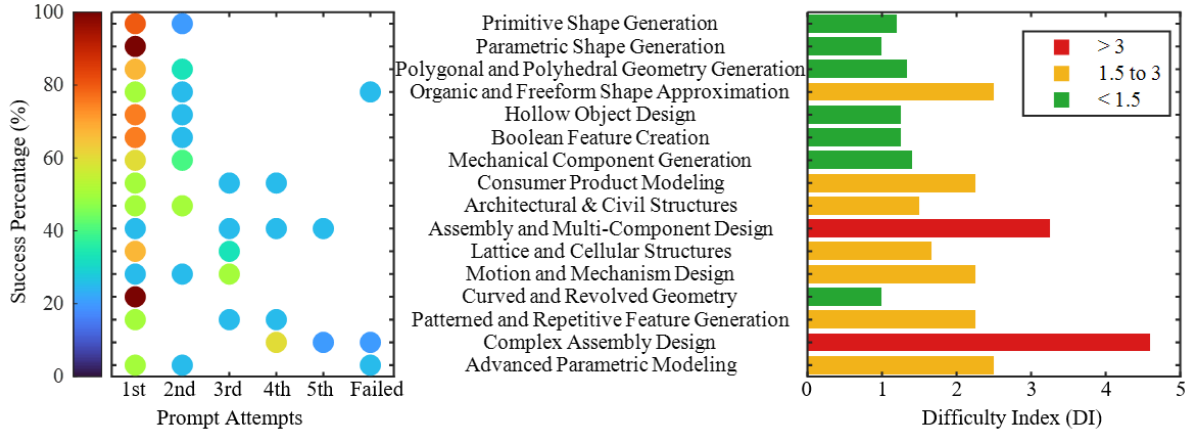


Figure 5. Capability-dependent performance of Microsoft Copilot for AI-assisted CAD generation. The left panel shows the distribution of successful STL generation across prompt attempts for different CAD capability categories, where the marker color represents the percentage of successful generations according to the color scale shown on the far left. The right panel presents the corresponding difficulty index calculated using Equation (1). Green bars indicate low-difficulty categories, yellow bars indicate moderate-difficulty categories, and red bars indicate high-difficulty categories. Overall, the results demonstrate that basic, parametric, and curved geometries can generally be generated with fewer prompt attempts, whereas assemblies, mechanisms, repetitive structures, and complex multi-component designs require greater prompt refinement.

3.6 Extreme CAD Challenge: V8 Engine Generation. To further evaluate the upper limits of Copilot's CAD generation capability, an extreme benchmark challenge involving the generation of a V8 engine was conducted. A V8 engine was selected because it contains numerous interconnected mechanical components, complex geometric relationships, and assembly-level features that are significantly more demanding than the geometries included in the easy, intermediate, and complex benchmark datasets [32]. Figure 6a–d presents the STL files generated from four successive prompt-engineering attempts, while Figure 6e shows a real V8 engine for visual comparison.

3.6.1 The first prompt simply requested a V8 engine STL model. Although Copilot successfully generated an STL file, the resulting geometry consisted primarily of a simplified engine block with limited detail and poor representation of the actual engine architecture (Figure 6a). The model generation required approximately 1 min 56 s. While recognizable as an engine-like object, many key components and mechanical relationships were absent.

3.6.2 The second prompt was expanded to explicitly request major engine components, including the pistons, crankshaft, camshaft, cylinder heads, intake manifold, and exhaust manifold. This refinement resulted in a noticeable increase in geometric detail and component representation (Figure 6b). The generation time increased to

approximately 2 min 53 s, indicating a greater computational effort associated with the more detailed design requirements.

3.6.3 The third prompt further emphasized realism by requesting a highly realistic CAD model with clean geometry, accurate proportions, well-defined surfaces, and suitability for rendering and 3D printing. Compared with the previous attempts, the resulting model exhibited an improved arrangement of engine components and a substantially more detailed overall structure (Figure 6c). The generation process required approximately 3 min.

3.6.4 The final prompt was refined further by explicitly requiring realistic mechanical relationships between components and prohibiting simplified or blurry geometries. The generated STL file showed the highest level of detail among all four attempts, including additional external housing, pulley-like structures, and more realistic engine proportions (Figure 6d). The model was generated in approximately 2 min 27 s.

Despite the progressive improvements achieved through prompt engineering, the generated models remained substantially simplified compared with a real V8 engine (Figure 6e). Internal components such as pistons, connecting rods, valve trains, timing mechanisms, bearings, and assembly constraints were not faithfully reproduced. Nevertheless, the results clearly demonstrate that prompt refinement can significantly improve the completeness and realism of AI-generated CAD models. More importantly, the study highlights the current limitations of general-purpose generative AI systems when tasked with producing highly detailed mechanical assemblies that require extensive engineering knowledge, part-to-part relationships, and manufacturing-level design fidelity.

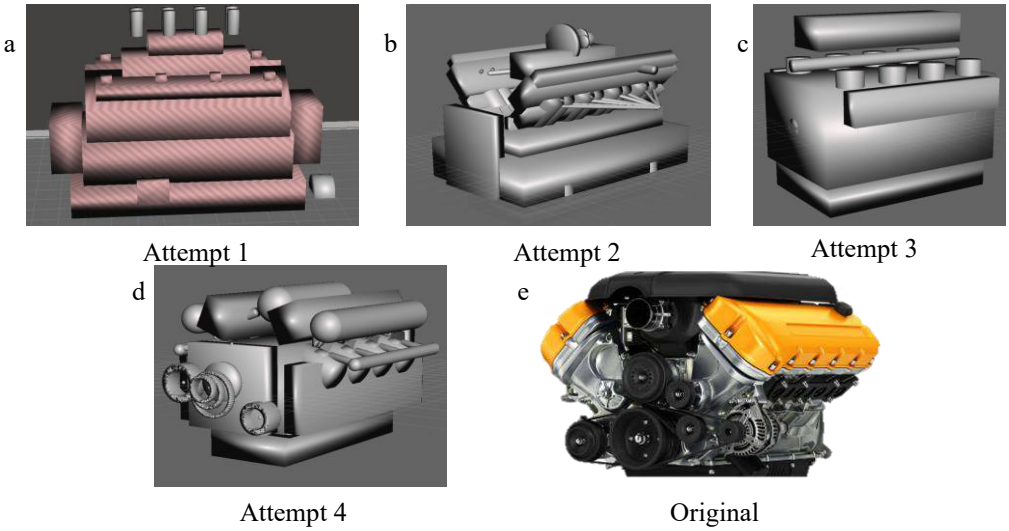


Figure 6. Progressive improvement in V8 engine CAD generation through prompt engineering. (a) First-attempt STL generated from a generic V8 engine prompt, (b) second-attempt STL generated after specifying major engine components, (c) third-attempt STL generated with additional realism and geometric quality requirements, (d) fourth-attempt STL generated using detailed mechanical and proportional constraints, and (e) reference image of a real V8 engine.

4 Conclusions

This study evaluated the capability of Microsoft Copilot, a publicly accessible generative AI platform, to generate CAD models and STL files from natural-language prompts. Through a benchmark dataset consisting of 60 design tasks of varying complexity, several important insights were obtained. First, both the number of prompt iterations and the CAD generation time increased as design complexity increased. Easy-level designs were often generated successfully on the first attempt, whereas intermediate and complex designs frequently required prompt refinement before acceptable STL files could be produced. Second, prompt length alone was not a reliable indicator of success. Instead, the quality of the generated CAD model was primarily determined by the clarity and specificity of the design requirements. Adding more text did not necessarily improve model quality unless the prompt explicitly described important geometric features, dimensions, and relationships between components. Third, the results demonstrated that generative AI does not inherently understand the defining characteristics of an object. For example, a prompt requesting a football often produced a simple spheroidal geometry whereas explicitly requesting features such as laces and seam lines resulted in a significantly more accurate football model. Similarly, even small changes in wording could substantially alter the final output which highlighs the importance of precise communication when interacting with commercially available general-purpose AI tools.

Overall, the findings indicate that generative AI can serve as a viable CAD-generation tool for users with limited CAD experience while also providing productivity benefits to professional CAD designers. The technology shows particular promise for rapidly creating simple components, conceptual models, and prototype-ready STL files, thereby reducing the time required for routine modeling tasks. However, limitations remain for highly complex assemblies, mechanically constrained systems, and intricate organic geometries. In these cases, AI-generated models should be viewed as a starting point for further refinement rather than a final engineering-ready design. The outcomes of this work help establish realistic expectations regarding the capabilities and limitations of widely available generative AI tools for CAD generation. Furthermore, the results demonstrate the potential of AI to lower barriers to entry in 3D modeling, making design creation more accessible to non-experts, CAD engineers, educators, rapid-prototyping users, and small-scale manufacturers who may not possess extensive CAD expertise. As generative AI technology continues to evolve, its integration into engineering design workflows may significantly improve accessibility, efficiency, and productivity across multiple industries.

Acknowledgment

The authors sincerely thank Dr. Ece Yaprak, Chair of Engineering Technology and course instructor for the Senior Design course, for continuous inspiration, guidance, and support throughout this project.

Funding Data

This research received no external funding.

Conflict of Interest

The authors declare that they have no conflict of interest.

Consent to Participate declaration

Not applicable.

Consent to Publish

Not applicable.

Clinical Trial Number

This study did not involve a clinical trial.

Ethics Approval

Not applicable. This study did not involve human participants, human subjects, animals, patient data, or clinical investigations.

Data Availability Statement

All data generated or analyzed during this study are included in this published article and its supplementary information files. The supplementary materials contain the complete benchmark dataset, including easy-, intermediate-, and complex-level benchmark prompts (Tables S1-S3), supplementary Figure S1 showed the evaluation results supporting the findings of this study.

Reference

[1] Koziar, M., Hubal, H., Burchak, I., Botviniev, M., and Saveliev, D., 2025, “The Impact of CAD Software on the Teaching of Engineering Graphics: A Systematic Review,” Periodicals of Engineering and Natural Sciences, **13**(1), pp. 17–39. https://doi.org/10.21533/pen.v13.i1.278.

[2] Kyratsis, P., Tzotzis, A., and Davim, J. P., 2025, “A Review on CAD-Based Programming for Design and Manufacturing,” *CAD-Based Programming for Design and Manufacturing*, P. Kyratsis, A. Tzotzis, and J.P. Davim, eds., Springer Nature Switzerland, Cham, pp. 1–26. https://doi.org/10.1007/978-3-031-78747-8_1.

[3] Hamade, R. F., Artail, H. A., and Jaber, M. Y., 2007, “Evaluating the Learning Process of Mechanical CAD Students,” Computers & Education, **49**(3), pp. 640–661. https://doi.org/10.1016/j.compedu.2005.11.009.
[4] Jinhe, L., Yongmin, Z., and Chengfan, G., 2026, “Enhancing Work-Readiness Through Scaffolding and Cognitive Transfer in CAD Education: A Twelve-Year Reflective Case Study,” Education Sciences, **16**(7), pp. 992–1013. https://doi.org/10.3390/educsci16070992.
[5] Chandrasekera, T., Hosseini, Z., and Perera, U., 2025, “Can Artificial Intelligence Support Creativity in Early Design Processes?,” International Journal of Architectural Computing, **23**(1), pp. 122–136. https://doi.org/10.1177/14780771241254637.
[6] Nashaat, B., and Elzeni, M. M., 2025, “Reviewing AI in Architectural Computational Design: Applications, Opportunities, and the AI-ACD Workflow for Improved Design Integration,” International Journal of Architectural Computing, p. 14780771251405443. https://doi.org/10.1177/14780771251405443.
[7] Tammisto, E., 2025, “Usage of Artificial Intelligence in Industrial Design Processes,” Tekoälyn hyödyntäminen osana teollisen muotoilun prosesseja. [Online]. Available: https://lutpub.lut.fi/handle/10024/169011. [Accessed: 11-Aug-2026].
[8] Daareyni, A., Martikkala, A., Mokhtarian, H., and Ituarte, I. F., 2025, “Generative AI Meets CAD: Enhancing Engineering Design to Manufacturing Processes with Large Language Models,” Int J Adv Manuf Technol. https://doi.org/10.1007/s00170-025-15830-2.
[9] Park, J., Murdivien, S. A., Choi, K., Mun, D., and Um, J., 2026, “Generative 3D Appearance Design: A Survey of Generation, Segmentation and Editing by Artificial Intelligence,” J Comput Des Eng, **13**(1), pp. 1–23. https://doi.org/10.1093/jcde/qwaf128.
[10] Rao, V. M., Hla, M., Moor, M., Adithan, S., Kwak, S., Topol, E. J., and Rajpurkar, P., 2025, “Multimodal Generative AI for Medical Image Interpretation,” Nature, **639**(8056), pp. 888–896. https://doi.org/10.1038/s41586-025-08675-y.
[11] Shoitan, R., Moussa, M. M., Tawfik, N., Cho, Y.-I., and Abdallah, M. S., 2026, “Exploring Generative Artificial Intelligence: A Comprehensive Guide,” PeerJ Comput. Sci., **12**, p. e3276. https://doi.org/10.7717/peerj-cs.3276.
[12] Stratton, J., 2024, “An Introduction to Microsoft Copilot,” *Copilot for Microsoft 365: Harness the Power of Generative AI in the Microsoft Apps You Use Every Day*, J. Stratton, ed., Apress, Berkeley, CA, pp. 19–35. https://doi.org/10.1007/979-8-8688-0447-2_2.
[13] Ronanki, K., Arvidsson, S., and Axell, J., 2026, “Prompt Engineering Guidelines for Using Large Language Models in Requirements Engineering,” *Software Engineering and Advanced Applications*, D. Taibi, and D. Smite, eds., Springer Nature Switzerland, Cham, pp. 245–262. https://doi.org/10.1007/978-3-032-04207-1_17.
[14] Qian, Y., 2025, “Prompt Engineering in Education: A Systematic Review of Approaches and Educational Applications,” Journal of Educational Computing Research, **63**(7–8), pp. 1782–1818. https://doi.org/10.1177/07356331251365189.
[15] Bansal, P., 2024, “Prompt Engineering Importance and Applicability with Generative AI,” JCC, **12**(10), pp. 14–23. https://doi.org/10.4236/jcc.2024.1210002.

[16] Chen, B., Zhang, Z., Langrené, N., and Zhu, S., 2025, “Unleashing the Potential of Prompt Engineering for Large Language Models,” Patterns, **6**(6). https://doi.org/10.1016/j.patter.2025.101260.
[17] Velásquez-Henao, J. D., Franco-Cardona, C. J., Cadavid-Higuita, L., Velásquez-Henao, J. D., Franco-Cardona, C. J., and Cadavid-Higuita, L., 2023, “Prompt Engineering: A Methodology for Optimizing Interactions with AI-Language Models in the Field of Engineering,” DYNA, **90**(SPE230), pp. 9–17. https://doi.org/10.15446/dyna.v90n230.111700.
[18] Zhang, L., Le, B., Akhtar, N., Lam, S.-K., and Ngo, D., 2026, “Large Language Models for Computer-Aided Design: A Survey,” ACM Comput. Surv., **58**(9), p. 225:1-225:39. https://doi.org/10.1145/3787499.
[19] Li, K.-Y., Huang, C.-K., Chen, Q.-W., Zhang, H.-C., and Tang, T.-T., 2025, “Generative AI and CAD Automation for Diverse and Novel Mechanical Component Designs under Data Constraints,” Discov Appl Sci, **7**(4), p. 315. https://doi.org/10.1007/s42452-025-06833-5.
[20] Berger, E., Dammann, M. P., Mehlstäubl, J., Saske, B., Braun, F., and Paetzold-Byhain, K., 2025, “Challenges and Opportunities in the Integration of Generative AI with Computer-Aided Design,” Proceedings of the Design Society, **5**, pp. 881–890. https://doi.org/10.1017/pds.2025.10102.
[21] Sun, Y., Cheng, H., Zheng, S., Yu, H., and Zou, H., 2026, “Balancing Speed and Executability in Interactive Text-to-CAD Code Generation for Early-Stage Parametric CAD Ideation,” J. King Saud Univ. Comput. Inf. Sci., **38**(6), p. 323. https://doi.org/10.1007/s44443-026-00754-y.
[22] Zhou, Z., Han, J., Du, L., Fang, N., Qiu, L., and Zhang, S., 2026, “CAD-Judge: Toward Efficient Morphological Grading and Verification for Text-to-CAD Generation,” *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1021–1025. https://doi.org/10.1109/ICASSP55912.2026.11461978.
[23] Wendland, M., 2025, “Exploring the Potentials and Limitations of AI-Based CAD Design for Mechanical Engineering.” https://doi.org/10.36315/2025v2end018.
[24] Guan, Y., Wang, X., Xing, X., Zhang, J., Xu, D., and Yu, Q., 2025, “CAD-Coder: Text-to-CAD Generation with Chain-of-Thought and Geometric Reward,” *Advances in Neural Information Processing Systems*, Curran Associates, Inc., pp. 59765–59789. https://doi.org/10.52202/085713-1999.
[25] Wang, R., Sun, S., Ma, W., and Bian, J., 2026, “CAD-Tokenizer: Towards Text-Based CAD Prototyping via Modality-Specific Tokenization,” International Conference on Learning Representations, **2026**, pp. 147972–147992.
[26] Li, J., Ma, W., Li, X., Lou, Y., Zhou, G., and Zhou, X., 2025, “CAD-Llama: Leveraging Large Language Models for Computer-Aided Design Parametric 3D Model Generation,” *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18563–18573. https://doi.org/10.1109/CVPR52734.2025.01730.
[27] Chester, I., 2007, “Teaching for CAD Expertise,” Int J Technol Des Educ, **17**(1), pp. 23–35. https://doi.org/10.1007/s10798-006-9015-z.

[28] Berselli, G., Bilancia, P., and Luzi, L., 2020, “Project-Based Learning of Advanced CAD/CAE Tools in Engineering Education,” Int J Interact Des Manuf, **14**(3), pp. 1071–1083. https://doi.org/10.1007/s12008-020-00687-4.
[29] Frank, G., Entner, D., Prante, T., Khachatouri, V., and Schwarz, M., 2014, “Towards a Generic Framework of Engineering Design Automation for Creating Complex CAD Models,” International Journal of Advances in Systems and Measurements, **7**, pp. 179–192.
[30] Naser, M. Z., 2025, “A Review of Prompt Engineering Techniques for Large Language Models,” International Journal of Human–Computer Interaction, **0**(0), pp. 1–35. https://doi.org/10.1080/10447318.2025.2607553.
[31] Liu, Y.-Y., Zheng, Z., Zhang, F., Feng, J.-C., Fu, Y.-Y., Zhai, J.-D., He, B.-S., Zhang, X., and Du, X.-Y., 2025, “A Comprehensive Taxonomy of Prompt Engineering Techniques for Large Language Models,” Front. Comput. Sci., **20**(3), p. 2003601. https://doi.org/10.1007/s11704-025-50058-z.
[32] Osborne, G., Prater, G., Lesiv, R., Lamb, D., and Castanier, M., “Vehicle Concept Model Abstractions for Integrated Geometric, Inertial, Rigid Body, Powertrain, and FE Analyses.” https://doi.org/10.1115/IMECE2011-63590.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementalDocuments.docx