

FROM CONCEPT TO MANUFACTURING: EVALUATING VISION-LANGUAGE MODELS FOR ENGINEERING DESIGN

A PREPRINT

✉ Cyril Picard^{*1}, ✉ Kristen M. Edwards^{*1}, ✉ Anna C. Doris¹, ✉ Brandon Man¹,
✉ Giorgio Giannone^{1,2}, ✉ Md Ferdous Alam¹, and ✉ Faez Ahmed¹

¹Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139

²Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, DK 2800

ABSTRACT

Engineering Design is undergoing a transformative shift with the advent of AI, marking a new era in how we approach product, system, and service planning. Large language models have demonstrated impressive capabilities in enabling this shift. Yet, with text as their only input modality, they cannot leverage the large body of visual artifacts that engineers have used for centuries and are accustomed to. This gap is addressed with the release of multimodal vision language models, such as GPT-4V, enabling AI to impact many more types of tasks. In light of these advancements, this paper presents a comprehensive evaluation of GPT-4V, a vision language model, across a wide spectrum of engineering design tasks, categorized into four main areas: Conceptual Design, System-Level and Detailed Design, Manufacturing and Inspection, and Engineering Education Tasks. Our study assesses GPT-4V’s capabilities in design tasks such as sketch similarity analysis, concept selection using Pugh Charts, material selection, engineering drawing analysis, CAD generation, topology optimization, design for additive and subtractive manufacturing, spatial reasoning challenges, and textbook problems. Through this structured evaluation, we not only explore GPT-4V’s proficiency in handling complex design and manufacturing challenges but also identify its limitations in complex engineering design applications. Our research establishes a foundation for future assessments of vision language models, emphasizing their immense potential for innovating and enhancing the engineering design and manufacturing landscape. It also contributes a set of benchmark testing datasets, with more than 1000 queries, for ongoing advancements and applications in this field.

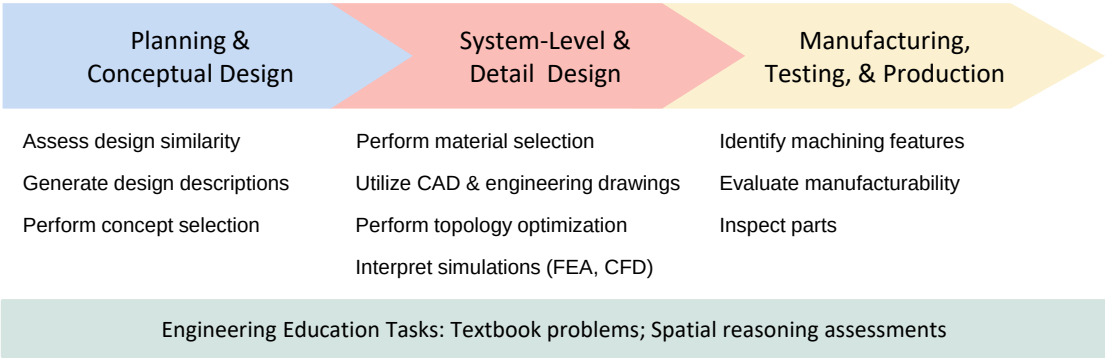


Figure 1: The product design and development process and the tasks we assess in each stage. The process is inspired by Ulrich et al. [2020].

^{*}Equal Contribution. Corresponding authors: cyrilp@mit.edu and kme@mit.edu

Contents

1	Introduction	3
2	Conceptual Design	6
2.1	Design Similarity	7
2.2	Design Descriptions	11
2.2.1	Match a Description to a Design	11
2.2.2	Generate a Design Description from a Sketch	14
2.3	Concept Selection	18
2.3.1	Generating Selection Criteria	18
2.3.2	Creating a Pugh Chart	19
3	System-Level And Detailed Design	22
3.1	Material Selection	23
3.1.1	Ashby Chart Look-Up	23
3.1.2	Ashby Chart Cross-Referencing	25
3.1.3	Material Selection for a Beam	25
3.2	Engineering Drawing Analysis and CAD Generation	28
3.2.1	Engineering Drawing Analysis	28
3.2.2	CAD Generation Using CADQuery, FeatureScript, and OpenSCAD	29
3.3	Topology Optimization	37
3.3.1	Basic Understanding	39
3.3.2	Invalid Design	48
3.3.3	3D Structure	51
3.4	Fluid Dynamics Simulation	54
3.4.1	Laminar Flow	55
3.4.2	Transition to Turbulent Flow	57
4	Manufacturing and Inspection	59
4.1	Design for Manufacturing	60
4.1.1	Design for Additive Manufacturing	60
4.1.2	Design for Subtractive Manufacturing	64
4.2	Post-Manufacturing Inspection	67
5	Engineering Education Tasks	71
5.1	Textbook Problems	72
5.2	Spatial Reasoning	79
6	Discussion	83
7	Conclusion and Future Work	85

1 Introduction

Large language models (LLMs) have shown promising performance in domains ranging from medicine [Arkoudas, 2023], to law [Katz et al., 2023], to mathematics and coding [Bubeck et al., 2023]. The chat-like interfaces offered by tools such as Google’s Bard [Manyika and Hsiao, 2023] or OpenAI’s ChatGPT [OpenAI, 2023] have enabled millions of users to query and leverage their immense implicit knowledge to assist in tasks ranging from diagnosing diseases to creating manufacturing drawing [Makatura et al., 2023], to supporting the conceptual design stage for a robotic arm [Stella et al., 2023]. Their use of natural language as the input modality offers an intuitive interface for humans to express a variety of problems, often in their mother tongue and without the need for technical jargon [Bubeck et al., 2023]. To better understand LLMs’ capabilities, researchers have investigated the performance of LLMs on standardized tests [Katz et al., 2023, Nori et al., 2023], and specific skills, such as reasoning or mathematics [Arkoudas, 2023, Bubeck et al., 2023], showing how well these models could perform. Siddharth et al. [2022] have identified many tasks within engineering design that could be automated by using natural language processing tools. Examples include automated design knowledge retrieval, discovery of user needs, and documentation generation.

However, expressing some tasks in text alone is prohibitively complex. In certain domains, especially those involving spatial information, a picture is truly worth one thousand words. Imagine if one had to write assembly instructions for IKEA furniture using text instructions alone. One would have to describe each part, define its orientation and instruct where to put screws repeatedly using text similar to “Insert screw A in the third hole starting from the top, where the top of the board is the shorted edge that has a slot through its length.” Unsurprisingly, engineers have relied on visual artifacts to communicate and support their work for centuries [Henderson, 1999]. From sketches that highlight the key working principles of mechanisms to manufacturing drawings that elicit all the information needed to manufacture, assemble, and inspect parts, visual representations are ubiquitous in engineering.

Recently, powerful multimodal language models have been proposed. In particular, text and vision models like GPT-4V [OpenAI, 2023], LLaVA 1.5 [Liu et al., 2023b], and Fuyu-8B [Bavishi et al., 2023] have shown immense promise since their public release. These vision-language models (VLMs) can take images and text as input and generate text as output. Specifically, GPT-4V builds upon the leading LLM, GPT-4, and allows for visual inputs. Researchers have begun exploring the capabilities of GPT-4V in several application domains. Examples include image understanding and reasoning [Yang et al., 2023b], image and language association [Liu et al., 2023a], and optical character recognition [Shi et al., 2023]. Given the visual nature of engineering design, can GPT-4V effectively perform tasks or assist engineers in this domain?

Engineering design encompasses a broad range of tasks within the product design process, as shown in Figure 1. These include: (1) generating and selecting concepts, (2) choosing between modular and integral structures, (3) sizing components and selecting materials, and (4) prototyping, manufacturing, and inspection. There are several key challenges that engineers face during this process. These include dynamically changing technologies and markets, trade-offs between desired qualities, the need to process, analyze, and synthesize lots of information, and time pressure during the process which necessitates quick decisions often without complete information [Ulrich et al., 2020]. Large multimodal language models like GPT-4V may be well-posed to automate tasks and support engineers with these challenges. For example, such models could sweep through thousands of sketches stored in knowledge databases and enable engineers to search through them using natural language. It could also analyze CAD models as they are constructed to include design-for-manufacturing rules early on, potentially reducing the manufacturing challenges in later stages of the process.

Objectives and Contributions This work is a preliminary exploration of GPT-4V’s capabilities as it pertains to engineering design tasks that require textual and visual inputs. We perform qualitative and quantitative analyses on a variety of engineering tasks for which GPT-4V may assist engineers, summarized in Figure 2. We developed and tested prompts that assess these tasks, and we discuss our findings and implications for using GPT-4V within engineering design. We aimed, wherever possible, for larger sample sizes and quantitative analyses, despite the lack of API at the start of our investigation. A detailed list of the tasks we explored, experiments we ran, and datasets we utilized and provide with this publication are detailed below:

- I. We developed and performed quantitative experiments to evaluate GPT-4V and create benchmarks for future VLMs.
 - (a) 410 queries in which GPT-4V assessed the design similarity of early-stage sketches and generated an idea map. Section 2.1.
 - (b) 90 queries to evaluate if GPT-4V can match early-stage sketches to their design descriptions. Section 2.2.1.
 - (c) 90 queries to evaluate if LLaVA 1.5 can match early-stage sketches to their design descriptions. Section 2.2.1.

- (d) 67 queries across 9 experiments in which we performed sequential prompting to characterize GPT-4V’s understanding of engineering drawings and to assess its ability to generate CAD scripts. We asked GPT-4V to work with three different scripting languages: CADQuery, FeatureScript, and OpenSCAD. Section 3.2.
 - (e) 60 queries from 20 CAD-generated parts that are tested for manufacturability via additive manufacturing in three trials. Section 4.1.1.
 - (f) 60 queries from 20 CAD-generated parts that are tested for identifying machining features in three trials. Section 4.1.2.
 - (g) 69 queries from 23 images testing if GPT-4V can identify defects in images of concrete from CODEBRIM (Concrete DEfect BRidge IMage dataset). Section 4.2.
 - (h) 132 queries from 44 textbook questions repeated three times. Section 5.1.
 - (i) 150 queries from three 10-question spatial reasoning tests repeated five times. Section 5.2.
- II. We developed and performed qualitative case studies of GPT-4V’s performance by:
- (a) Assessing GPT-4V’s ability to generate text descriptions from design sketches for five designs. Section 2.2.2.
 - (b) Evaluating GPT-4V’s generation of selection criteria and creation of a Pugh chart. Sections 2.3.1 and 2.3.2.
 - (c) Exploring GPT-4V’s ability to answer questions about material selection from Ashby charts. Section 3.1.
 - (d) Assessing GPT-4V’s understanding of topology optimization. Tasks include volume fraction estimation, technical captioning, and invalid design identification. Section 3.3.
 - (e) Assessing GPT-4V’s ability to interpret and draw conclusions from fluid dynamics simulation results via case studies. Specifically observing understanding for laminar flow (Section 3.4.1) and transient and turbulent flows (Section 3.4.2).
- III. We created and released datasets for future evaluations.
- (a) Datasets of the input images, input prompts, and answers for all eight quantitative experiments described above will be available on the project webpage <https://decode.mit.edu/projects/vlms4design/>
 - (b) Since the majority of experiments, images, and prompts were generated for this paper, we mitigate the chance that our data was part of GPT-4V’s training.

Structure of this paper We have structured the sections around the product design process as shown in Figure 1, inspired by Ulrich et al. [2020]. Section 2 delves into conceptual design, focusing on design sketches and text descriptions. Section 3 focuses on tasks related to the detailed design stage, where precision is important. Section 4 assesses the general knowledge of manufacturing and tests the performance on inspection tasks. Section 5 investigates the performance of GPT-4V on textbook problems and spatial reasoning tests, providing some insights into its over-arching engineering knowledge and skills.

Each section is structured as follows:

- (1) Overview and motivation
- (2) Evaluation, which is a concise summary of our findings
- (3) Subsections containing:
 - (a) Specific engineering tasks
 - (b) Methodology for quantitative experiments
 - (c) Prompt examples
 - (d) Task-specific results and discussions

Finally, Section 6 offers a broader perspective of the capabilities of GPT-4V for engineering design, discusses its limitations, and comparatively showcases the performance of an open-source vision-language model, LLaVA 1.5, on some tasks.

Experimental Setup We ran all our prompts within the ChatGPT user interface, specifically harnessing the capabilities of the *September 25th update*. It should be noted that the principal content of the paper and the primary experiments were conducted using this particular version², with a specific focus on the vision feature. We do not include internet access, plugins, or a code interpreter unless explicitly mentioned (for example see 3.3.1).

²The majority of the research and experimental work for the initial version of this paper was carried out before November 2nd, 2023, leveraging GPT-4V released on September 25th, 2023. Subsequent updates to the paper may reflect varying outcomes consequent to advancements or revisions in GPT-4V versions.

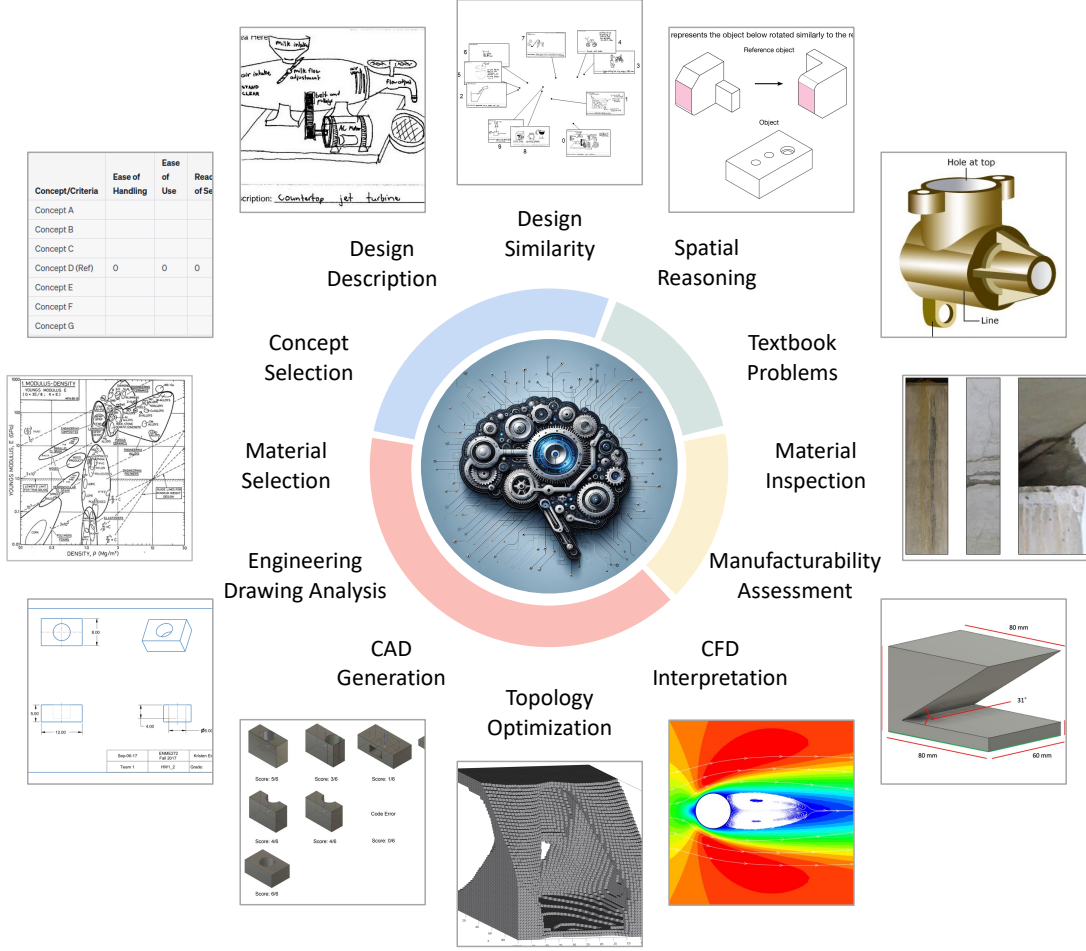


Figure 2: We explored GPT-4V’s ability to perform numerous engineering design tasks that utilize both visual and textual information.

Our methodology is outlined by the following key points:

- (1) **Emphasis on Images.** The primary function of our use of GPT-4V is centered around image-based tasks. This is where our efforts and resources are concentrated, aiming to explore the model’s understanding and analysis of visual data.
- (2) **Short Text Prompts.** We utilize short, straightforward prompts that prioritize vision tasks. This is to ensure a focus on visual analysis over the complexity of prompt engineering or creating lengthy custom instructions.
- (3) **Engineering Expertise through Prompting.** In instances where specialized knowledge is needed, we prompt GPT-4V to encourage engineering expertise, i.e. "You are an engineer with expertise in topology optimization". By doing so, the answers provided by the system tend to be more precise and detailed, leveraging specialized language and extracting the most accurate information from the visual content being analyzed.
- (4) **Transparent Prompts and Answers.** In each section, we provide examples of our exact prompts and the exact response from the model, unless shortened with ellipses.

By maintaining these standards, we hope to provide clear experiments that can be used as benchmarks for future LLM evaluations in engineering design.

2 Conceptual Design

Overview and Motivation Conceptual design is an early stage of the product development process, during which designers identify customer needs, establish target specifications, and then generate, select, and test many concepts [Ulrich et al., 2020]. Experts estimate that 70-80% of a product’s life-cycle costs are determined during the early design stages [Corbett and Crookall, 1986, Pahl et al., 2007], highlighting the importance of decisions made during this stage. There exists a rich body of research examining concept generation during this stage [Bryant et al., 2005, Toh and Miller, 2019, Das and Yang, 2022], creativity evaluation in conceptual design [Amabile, 1982, Shah et al., 2003, John and Sharon, 2009, Baer and Kaufman, 2019, Cseh and Jeffries, 2019], cost estimates and environmental assessments during conceptual design [Saravi et al., 2008, Fu et al., 2020], and, recently, multimodal machine learning in conceptual design [Yuan et al., 2021, Edwards et al., 2021, Song et al., 2023a, Su et al., 2023].

In conceptual design, two of the primary design modalities are hand-drawn sketches and textual descriptions. Often these modalities are combined, and early-stage designs are represented as a sketch with accompanying text. The inherent multimodality of this stage makes it an intriguing candidate for testing GPT-4V’s capabilities. In addition, a vision language model (VLM) such as GPT-4V has great potential to be used as a tool during conceptual design. Some of the main tasks in conceptual design include generating sketches and descriptions of design ideas, comparing and iterating upon those ideas, and ultimately selecting a design to move forward with. In the following experiments, we explore how GPT-4V can act as an aide for these tasks. Specifically, we see how the multimodal capabilities enable GPT-4V to perform engineering tasks when both design sketches and text prompts are included as input. For effective multimodal learning, it is important to have designs with both image and text modalities, and sufficient datasets of these designs. There exist a number of multimodal conceptual design datasets that we use in the following experiments. However, an overarching theme in machine learning within the engineering domain is that most datasets are small, which poses a challenge for data-driven models. Large pretrained models like GPT-4V can help overcome this challenge because they have been trained on a plethora of information (although the exact training information of GPT-4V is not yet released), meaning that they have general knowledge about the world. Another challenge is faced during concept selection. Experts suggest generating a multitude of conceptual designs and then down-selecting through design evaluation [Ulrich et al., 2020]; however, the evaluation step is often performed by experts, which takes time and resources [Baer and Kaufman, 2019]. GPT-4V may be able to help engineers during the conceptual design stage by utilizing general knowledge and sketch understanding to interpret and compare designs, move between design representations (text and image), and perform concept selection tasks. We aim to evaluate these capabilities in the following experiments. A concise summary of our findings is provided below.

Evaluation Here are our assessment areas for GPT-4V’s conceptual design abilities, and our findings for each:

- (i) **Assessing Design Similarity 2.1** How does GPT-4V’s consistency in assessing design sketch similarity compare to human benchmarks?
 - We measure consistency using two measures from Ahmed et al. [2018]— self-consistency and transitive violations in assessing sketch triplet queries. The model is able to assess design similarity with higher self-consistency than human raters (94% compared to 62.8% average for human raters) and as few transitive violations as the top human raters.
- (ii) **Matching Design Representations 2.2.1** Can GPT-4V accurately match design sketches with their text descriptions under varying information conditions?
 - We ran 90 queries: three trials each comprised of ten multiple choice questions for each of three different cases. We found the following results. When provided the entire design sketch including a handwritten description, the model matched a design sketch to its appropriate text description 10/10 times for all three trials, however with the handwritten description removed, the score dropped to an average of 5.33/10, which we can compare to a score of 2.5/10 for randomly matching. Incorrect answers were often in the form of choosing “None of the above,” so when given the same task without the “None of the above” option, the score rose to an average of 7/10.
- (iii) **Generating Design Descriptions 2.2.2** Is GPT-4V capable of generating effective descriptions for early-stage design sketches?
 - Qualitatively, we find that GPT-4V is able to generate accurate and useful design descriptions given hand-drawn sketches.
- (iv) **Generating Selection Criteria 2.3.1** How effectively does GPT-4V generate concept selection criteria in engineering design?
 - In our case study, we find that when provided a design task GPT-4V generates useful selection criteria that match those generated by design professionals.

- (v) **Generating a Pugh Chart 2.3.2** What is the extent and limitation of GPT-4V’s ability to generate Pugh charts for conceptual design evaluation?
- GPT-4V understands what a Pugh chart is, and can provide examples of the formatting, but often will not fill in the Pugh chart, or simply provide a “looks-like” Pugh chart given just a design task and design sketches. The model cites that it cannot fill in the Pugh chart without additional context about the designs, suggesting it may be able to if provided with more information.

2.1 Design Similarity

Determining if two designs are similar is an important part of conceptual design. Assessing design similarity can act as a proxy for assessing design novelty, which is a common criterion in concept selection [Ahmed et al., 2018]. Novelty expresses that a concept is rare, ingenious, imaginative, or surprising, as well as radical or transformational [Verhaegen et al., 2012]. However, novelty evaluation is often subjective. It can be easier for humans to articulate why they might rate designs as similar than why they would rate one design as more novel than another [Ahmed et al., 2018]. For this reason, past work has studied how humans assess similarity of concepts [Ahmed et al., 2018] as a method to build idea maps and identify novelty. Similarity comparisons can also help explore the design space by identifying clusters of similar ideas, potentially helping with faster design space exploration. Recently, researchers have compared how human evaluations of similarity compare to computationally determined similarity, and found that they diverge based on the level of abstraction of a product [Nandy and Goucher-Lambert, 2022].

Three main challenges arise with human evaluations of similarity:

1. **Evaluation speed and cost** Human evaluations are very expensive. Both the time and the cost of these evaluations are exacerbated as the number of designs increases since the number of similarity queries scales with n^2 the number of designs.
2. **Self-consistency** Humans may make different similarity assessments when they repeat the same evaluation.
3. **Transitive violations** Given designs A, B, and C, one cannot say that A is more similar to B, B is more similar to C, but that C is also more similar to A. This would violate the transitive property of inequality, since if $AB < AC$ (where AB is a measure of similarity between design A and design B) and $BC < BA$, then these imply that $CB < CA$, so saying $CA < CB$ cannot be true. A violation of this sort can be tested when the same three designs, which we call a triplet, are assessed for similarity multiple times.

In the following experiments, we evaluate if GPT-4V can effectively assess the similarity of designs, i.e. with high self-consistency and a low number of transitive violations. Our methodology, an example of our prompt, and the table of results are provided below.

Methodology We tasked GPT-4V with performing the same experiment that eleven human raters performed in Ahmed et al. [2018]. We have 10 early-stage design sketches from Starkey et al. [2016], Toh and Miller [2016] as shown in Figure 3. We group them into groups of three, which we call triplets. As shown in Context 2.1.1, we provide a triplet of design sketches labeled A, B, and C, and ask GPT-4V which design is most similar to A. Since we have 10 designs, we can make 360 triplets such that each design is considered design A for all 36 combinations of the other 9 designs as designs B and C. When given 360 of these triplet examples, we assessed if GPT-4V commits transitive violations. We then repeated 50 of the examples in order to evaluate GPT-4V’s self-consistency.

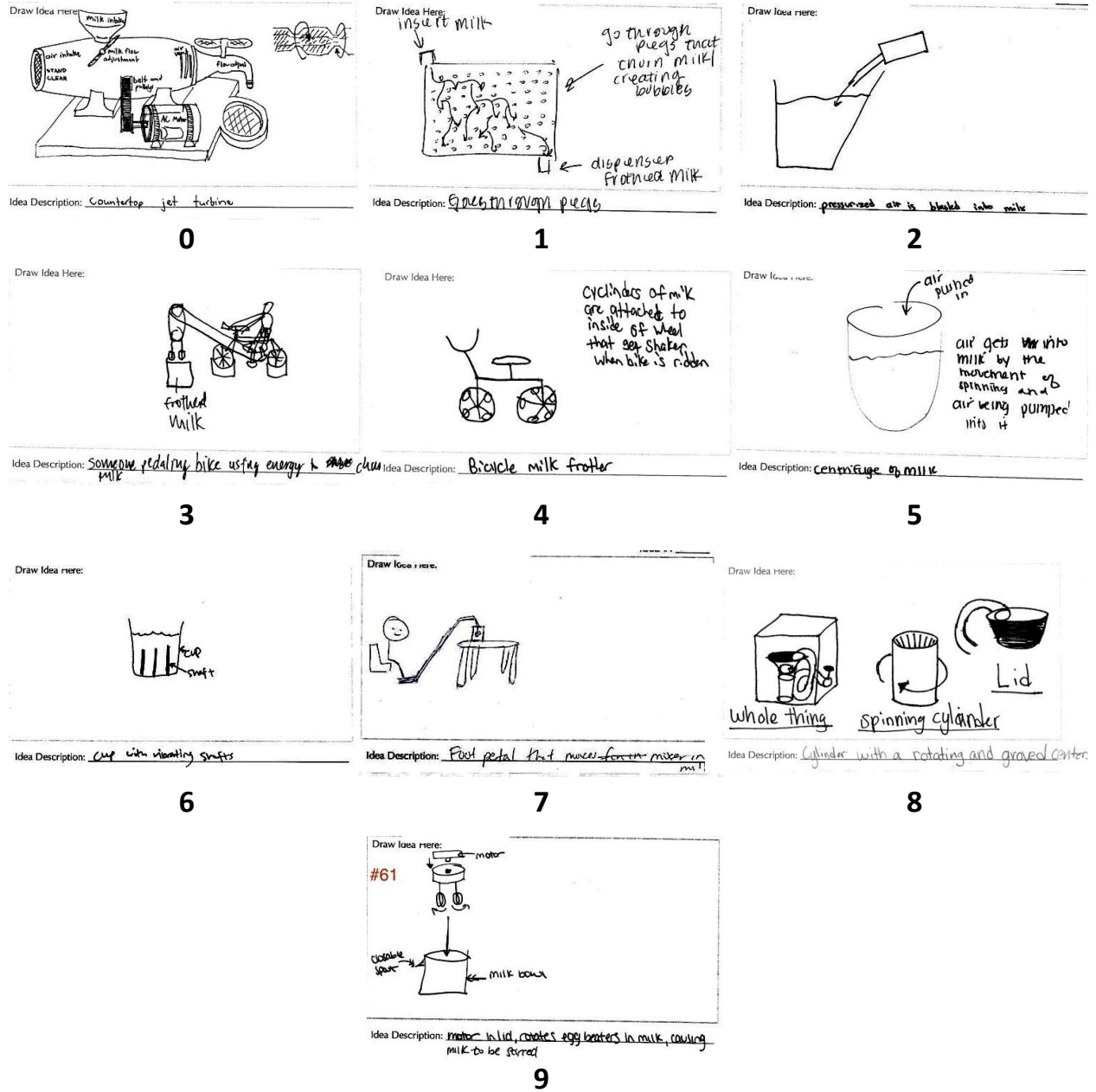
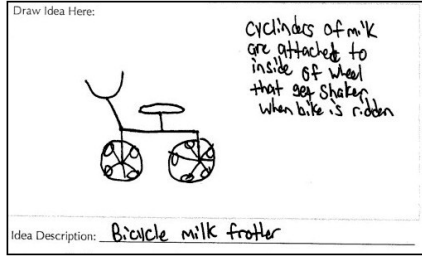


Figure 3: Ten conceptual designs of novel milk frothers. We task GPT-4V with assessing the similarity of these designs to one another.

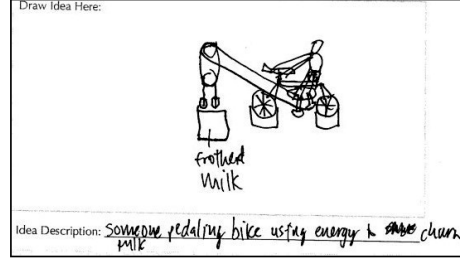
Context 2.1.1: Design Similarity

Prompt 1: (Design Similarity)

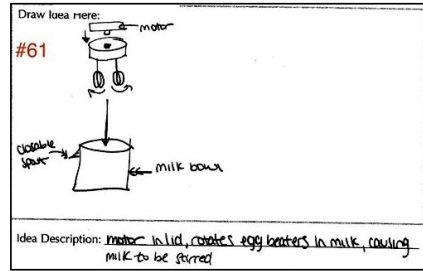
Which design is more similar to design A?



Design A



Design B



Design C

GPT-4V Answer:

Design B is more similar to Design A. Both involve bicycles and the concept of using pedaling or riding to froth milk. Design C is distinct as it involves a motor and an egg beater mechanism.

Table 1: A summary of the self-consistency and number of transitive violations of GPT-4V when evaluating 360 triplets for which designs are more similar. We compare the results with the corresponding values for eleven human raters reported in [Ahmed et al. \[2018\]](#). GPT-4V has higher self-consistency than the human raters and its number of transitive violations equals the lowest human rater value.

Rater	Self-consistency	Transitive Violations
1	91.6%	5
2	50.0%	5
3	83.3%	5
4	75.0%	10
5	58.3%	10
6	58.3%	20
7	41.6%	8
8	41.6%	20
9	58.3%	11
10	75.0%	12
11	58.3%	5
GPT-4V	94.0%	5

In addition, we utilize the generalized non-metric multidimensional scaling (GNMDS) technique introduced by [Agarwal et al. \[2007\]](#) to find 2-D embeddings of design sketches using triplet responses and generate a visualization of the ten designs where designs that are closer to each other are considered more similar. This technique is the same as used

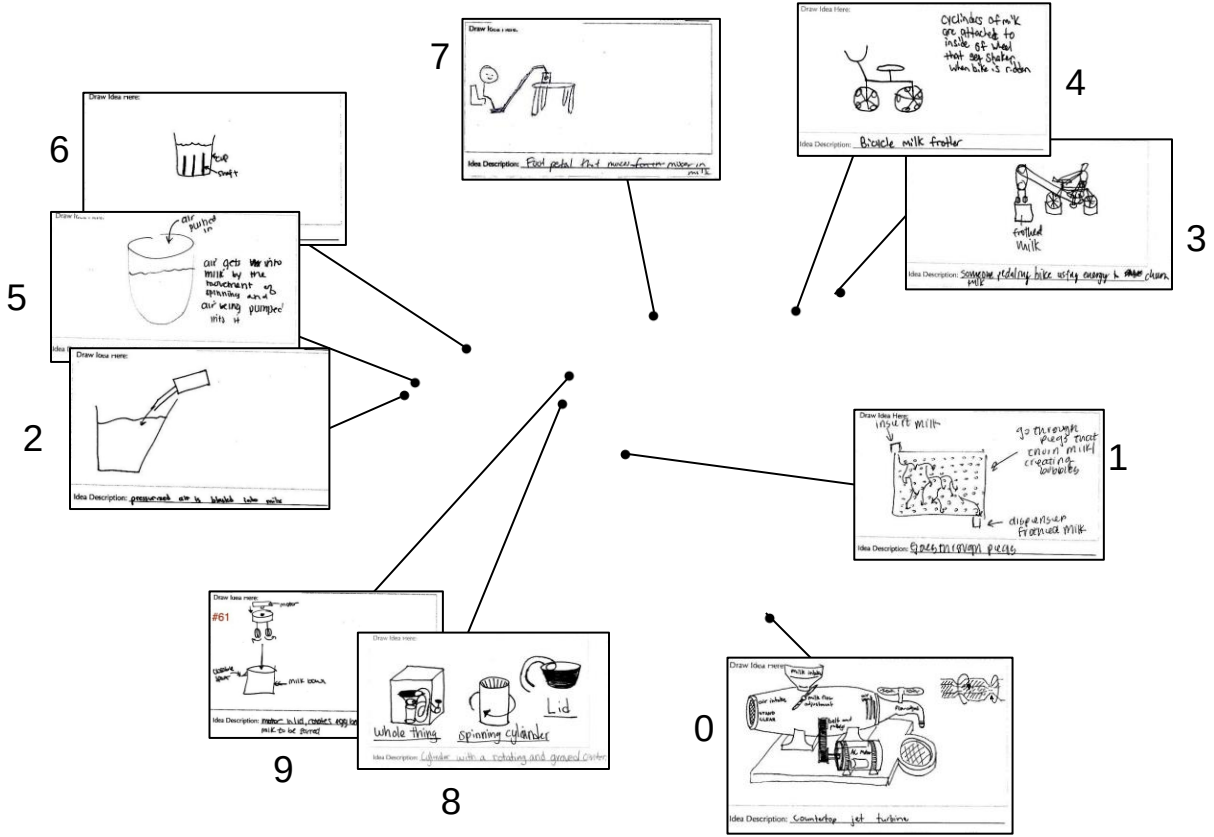


Figure 4: A map of the milk frother design sketches where sketches that are closer to each other are more similar. These are based on the responses by GPT-4V for 360 triplet similarity queries. We observe that the map clusters similar designs together and places unique designs further away from other designs.

in [Ahmed et al. \[2018\]](#) to generate a map of these designs from human ratings. The resultant map, referred to as the idea map of GPT-4V, is shown in Figure 4. The idea map shows a striking resemblance to human-generated idea maps reported in literature. We observe that the three designs that show cups filled with milk are grouped together (designs 2, 5, and 6) as well as the two bicycle-based designs (designs 3 and 4). This clustering of similar designs was also observed in the idea map of all human raters combined, shown in Figure 8 in [Ahmed et al. \[2018\]](#). GPT-4V’s idea map also places Sketch 0 further away from all other sketches, denoting that it was perceived as most novel. Coincidentally, the most novel sketch identified by the aggregated human ratings was also Sketch 0. Sketch 0, proposing a countertop jet turbine to froth milk was also the most novel sketch rated by the expert in their work. This serves as a validity check, demonstrating to us that GPT-4V is assessing the similarity of sketches in a meaningful way, which aligns with human ratings. We note that in past work, each human rater has a different map, and GPT-4V creates a unique map as well. The variability in individual human idea maps is likely influenced by diverse criteria for judging similarity. Consequently, establishing a definitive standard for sketch similarity is challenging. Therefore, we compared our results to the aggregated map by eleven human raters to gauge how GPT-4V’s assessments conformed with collective human wisdom.

These initial findings pave the way for future research avenues. First, GPT-4V’s capability to create idea maps is not only scalable, and ideal for evaluating a large number of design items, but also overcomes limitations faced by previous studies reliant on time-consuming human ratings. Second, the use of triplet queries for generating idea maps extends beyond sketches to other design forms like prototypes, 3D models, or multimedia, offering a novel approach to evaluating design similarity. These maps are valuable tools for designers, enabling them to better understand their design domain and leverage GPT-4V for more effective exploration of the design space.

Discussion: Design Similarity From the quantitative experiments using a total of 410 queries, with results summarized in Table 1, we find that GPT-4V is able to assess the similarity of designs with greater or equal performance as human

raters. In the 360 trials, GPT-4V made only five transitive violations, which matches the lowest number of transitive violations made by any of the eleven human raters. Additionally, in our trials, GPT-4V was self-consistent 94% of the time, which is greater than any of the human raters. A naive model could still be self-consistent without actually understanding much about a design sketch. So to gain insight into GPT-4V’s similarity assessment, we plot the designs using the GNMDS embedding technique, shown in Figure 4. We observe sensible clustering of three design sketches whose major features are cups with milk, and two design sketches whose major features are bicycles. While future work should explore how well VLMs perform at this task for other datasets, these results offer a promising suggestion that GPT-4V can effectively assess the similarity of conceptual design sketches. Future VLMs can be evaluated using these same methods to compare them against this version of GPT-4V. We have provided the dataset including all triplets for this purpose. These experiments were meant to test understanding, analysis, and evaluation. The results suggest that GPT-4V is able to understand and analyze design sketches in order to assess their similarity. Furthermore, assessing similarity among design triplets is a mundane and repetitive task. GPT-4V’s ability to perform this task may mean that human raters do not have to, which can save time and resources both in dataset creation and in the design process.

2.2 Design Descriptions

Through these experiments, we aim to evaluate how well GPT-4V understands the different representations of a design, in this case, textual and sketch. Understanding sketches is a first step in being able to evaluate and compare them, which is one of the end goals of the conceptual design phase [Ulrich et al., 2020]. We task GPT-4V with matching a design to its correct description given a number of options, and we also task GPT-4V with generating textual descriptions of a design given just a sketch. We specifically chose to perform description matching (section 2.2.1) in the form of multiple-choice questions because this allows for quantitative analysis. Additionally, we provide the exact questions and results so that future VLMs can be similarly evaluated. In fact, we evaluate both GPT-4V and LLaVA 1.5 in this manner; results are shown in Tables 2 and 3 respectively. While these description matching and generating tasks do not directly translate to common engineering tasks during the design process, we believe that GPT-4V’s performance on them sheds light on its ability to understand information from one modality (sketch) and then synthesize information in another modality (generating text). Furthermore, if a tool can automatically generate accurate and useful textual descriptions of conceptual design sketches, this could allow engineers to (1) create an easily-searchable catalog of early-stage designs, and (2) more easily generate multimodal datasets of paired sketches and text descriptions, which are necessary for multimodal machine learning in the engineering domain [Song et al., 2023b]. Automatically generating relevant textual descriptions for hand-drawn sketches can also help communicate design ideas to design team members and potential stakeholders, which is a primary role of sketching [Das and Yang 2022]. Lastly, it can help human raters judge design ideas for creativity, novelty, quality, and other common design metrics [Shah et al. 2003].

2.2.1 Match a Description to a Design

Given an image of an early-stage design sketch, and four different design description options, we test if GPT-4V can identify the correct description. We analyze GPT-4V’s performance on these simple tasks to gain a basic understanding of whether more challenging description generation tasks are possible. Our methodology, four examples of our prompts, and a table of our results are provided below.

Methodology We assessed if GPT-4V can match a design sketch to its correct text description for three different cases:

1. We provide the whole image including the handwritten text description, as well as four description options including “None of the above”. Context 2.2.1.
2. We provide the image with the handwritten text description removed, as well as four description options including “None of the above”. Context 2.2.2.
3. We provide the image with the handwritten text description removed, and only three text description options, removing the “None of the above” option. Context 2.2.3 and Context 2.2.4.

We ran 90 total queries: three trials each comprised of ten multiple choice questions for each of three different cases³. We included one multiple-choice question for each of the designs in Figure 3. Examples of these questions are shown in Context 2.2.1 - 2.2.4. As the answer options, we provided three text descriptions from among the 10 design sketches. Table 2 displays the results for each of the three cases, as well as the final score out of 10.

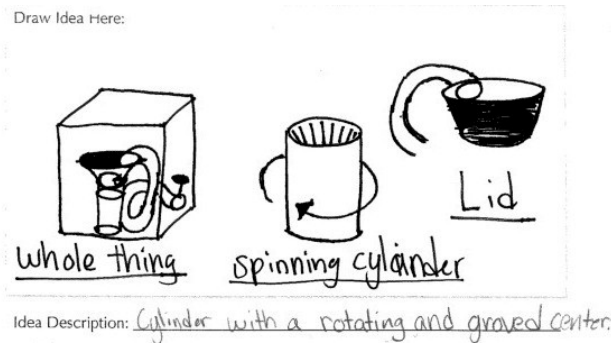
³Trial 1 used the *September 25th update* of GPT-4V; Trials 2 and 3 used the *November 6th update* of GPT-4V.

Context 2.2.1: Design Descriptions

Prompt 1: (Matching a Design to its Description: Providing the full design including handwritten text.)

Which text description matches the given design?

- A. Motor in lid, rotates egg beaters in milk, causing milk to be stirred
- B. Cylinder with a rotating and grooved center
- C. Pressurized air is blasted into milk
- D. None of the above



GPT-4V Answer:

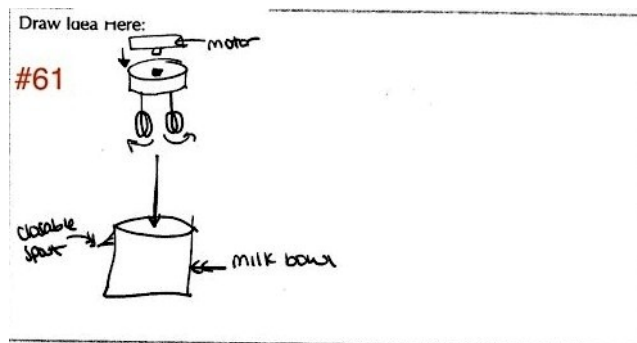
B. Cylinder with a rotating and grooved center.

Context 2.2.2: Design Descriptions

Prompt 1: (Matching a Design to its Description: Providing the design, but removing the handwritten text description.)

Which text description matches the given design?

- A. Bicycle milk frother
- B. Motor in lid, rotates egg beaters in milk, causing milk to be stirred
- C. Goes through pegs
- D. None of the above



GPT-4V Answer:

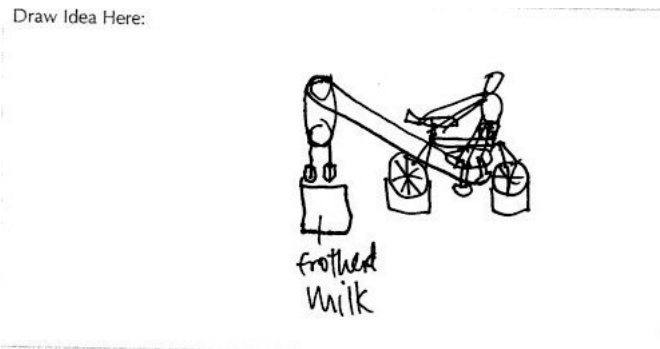
B. Motor in lid, rotates egg beaters in milk, causing milk to be stirred.

Context 2.2.3: Design Descriptions

Prompt 1: (Matching a Design to its Description: Providing the design, but removing the handwritten text description, and removing the "None of the above" option.)

Which text description matches the given design?

- A. Someone pedaling bike using energy to churn milk
- B. Cylinder with a rotating and grooved center
- C. Motor in lid, rotates egg beaters in milk, causing milk to be stirred



GPT-4V Answer:

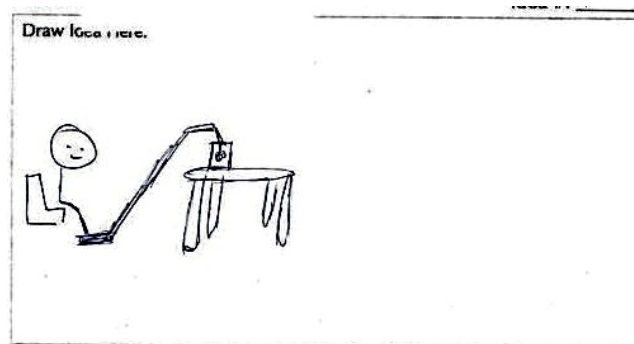
A. Someone pedaling bike using energy to churn milk

Context 2.2.4: Design Descriptions

Prompt 1: (Matching a Design to its Description: Providing the design, but removing the handwritten text description, and removing the "None of the above" option.)

Which text description matches the given design?

- A. Cup with vibrating shafts
- B. Countertop jet engine
- C. Foot pedal that moves mixer in milk



GPT-4V Answer:

D. A person sitting and looking at a computer on a table.

Note: None of the given options (A, B, C) match the provided design.

Table 2: Results for the three multiple-choice design description matching experiments. 90 queries were run across three trials, ten queries for each of the three cases. The full results for Trial 1 are displayed, as well as the scores for all three trials.

Design Number	Correct Answer	With Text Description	No text description	No text description, no “None of the above”
0	B	B	B	B
1	A	A	A	A
2	C	C	D	B
3	A	A	A	A
4	B	B	B	B
5	C	C	D	B
6	C	C	D	C
7	C	C	D	D
8	B	B	A	A
9	B	B	B	B
Trial 1 Score		10/10	5/10	6/10
Trial 2 Score		10/10	5/10	8/10
Trial 3 Score		10/10	6/10	7/10
Average Performance		100%	53.3%	70%

Table 3: Using LLaVA 1.5 rather than GPT-4V, here are the results for the three multiple-choice design description matching experiments.

Design Number	Correct Answer	With Text Description	No text description	No text description, no “None of the above”
0	B	B	A	A
1	A	A	A	A
2	C	C	C	C
3	A	A	C	A
4	B	B	B	B
5	C	C	C	B
6	C	C	B	B
7	C	C	D	C
8	B	B	A	A
9	B	B	B	B
Trial 1 Score		10/10	4/10	6/10
Trial 2 Score		10/10	4/10	5/10
Trial 3 Score		10/10	4/10	4/10
Average Performance		100%	40%	50%

In order to gain an understanding of GPT-4V’s capabilities in comparison to other VLMs, we perform this same experiment with LLaVA 1.5. The results are shown in Table 3. LLaVA 1.5 and GPT-4V alike perfectly match designs to their descriptions when the handwritten description is provided in the sketch. When the handwritten description is removed, GPT-4V outperforms LLaVA 1.5, as seen in Tables 2 and 3. In particular, GPT-4V’s errors are typically a result of selecting “None of the above,” and when that option is removed, and it is forced to select a description, its performance increases (from 5.33/10 to 7/10). LLaVA 1.5, on the other hand, makes errors that are difficult to interpret. For example, it consistently describes an image of a cup (design 5 in Figure 3) as “B. Foot pedal that moves mixer in milk.” for all six instances in which it is not provided the handwritten description.

This analysis reveals several key aspects of Vision Language Models (VLMs) like GPT-4V and LLaVA 1.5 in interpreting engineering designs. Firstly, the presence of textual context in sketches significantly enhances model accuracy, underscoring the importance of integrating text with visual data. The variability in model performance, particularly GPT-4V’s superiority in scenarios without handwritten descriptions, suggests that model choice should be tailored to specific task requirements. Additionally, GPT-4V’s tendency to select “None of the above” when uncertain highlights its cautious approach in ambiguous situations, reflecting a strategy to manage uncertainty. This behavior, along with the contrasting error patterns between GPT-4V and LLaVA 1.5, points to the need for deeper understanding and improvement in how different models process and interpret visual information, especially in the absence of textual cues. These findings are crucial for optimizing the use of VLMs in conceptual design contexts.

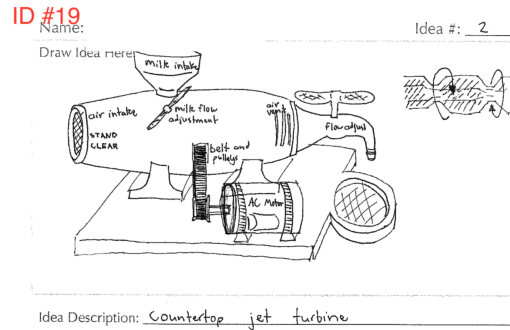
2.2.2 Generate a Design Description from a Sketch

Given an image of an early-stage design sketch, can GPT-4V generate a relevant and accurate design description?

Context 2.2.5: Design Descriptions

Prompt 1: (Generating a Design Description)

A student was asked to develop a new innovative product that froths milk in a short amount of time. Please describe this design.



GPT-4V Answer:

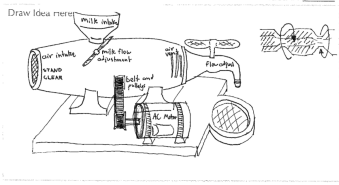
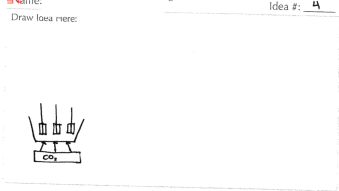
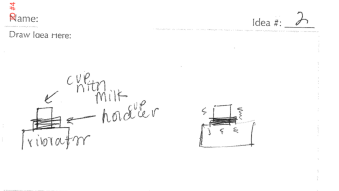
The design presented is titled "Countertop Jet Turbine." It appears to be a device intended to froth milk quickly using a turbine mechanism. Here's a breakdown of its features:

1. **Milk Intake:** This is where milk is poured into the device. It seems to be designed in a funnel shape to ensure easy pouring without spillage.
2. **Air Intake:** Positioned on the side of the main body, this feature likely serves the purpose of drawing in air to aid in the frothing process.
3. **Milk Flow Adjustment:** An adjustment knob or switch appears to be present, which probably allows the user to control the flow of milk or the intensity of frothing.
4. **Air Vent:** Located on the top side, the air vent might serve to release any excess air or to control the pressure within the device.
5. **Flow Adjust:** Positioned near the air vent, this seems to be another control, possibly to manage the flow of frothed milk out of the device or to adjust the consistency of the froth.
6. **Belt and Pulleys:** This part seems to be connected to an AC Motor. The belt and pulley system likely drives the turbine mechanism, converting the motor's rotational movement into the desired frothing action.
7. **AC Motor:** The power source of the device. This motor would drive the turbine at high speeds to froth the milk.
8. **Safety Warning - "STAND CLEAR":** This warning label suggests that the device might operate at high speeds or produce a forceful stream of frothed milk, and users should be cautious when it's running.

In summary, the "Countertop Jet Turbine" appears to be a high-powered milk frother that utilizes a jet turbine mechanism to aerate and froth milk in a short amount of time. The multiple adjustment controls suggest that users have a good degree of control over the frothing process.

We performed this experiment for five early-stage design sketches with varying drawing scores. The drawing scores are based on a Consensual Assessment Technique evaluation of students' milk frother designs [Starkey et al., 2016, Toh and Miller, 2016]. The scores can range from 1-7, but within the dataset of sketches, the scores range from 1-6. Table 4 shows the results for these design sketches. The selected sketches were chosen at random from among all sketches with a similar drawing score. We show how GPT-4V responds when simply asked to describe the design, versus when provided with a description of the original design task given to the students and then asked to describe the design. For brevity, in both cases, we prompt GPT-4V to respond in three sentences. The exact prompts are provided in the column headers of Table 4.

Table 4: GPT-4V generated descriptions from design sketches. We also include the expert-rated drawing score of each sketch. The sketches are ordered by descending drawing score.

Image	Drawing Score	Please describe this design in 3 sentences.	A student was asked to develop a new innovative product that froths milk in a short amount of time. Please describe this design in 3 sentences.
ID #19 Name: _____ Draw idea here:  Idea Description: Countertop jet turbine	5.5	The design depicts a countertop jet turbine. The apparatus features components such as an air intake, a milk intake, milk flow adjustment, an AC motor connected through belt and pulleys, and an air vent. Safety measures are indicated with a "STAND CLEAR" label near the air intake, and a flow adjustment knob is situated atop the device.	The design depicts a "Countertop Jet Turbine" that rapidly froths milk. It features an air intake for drawing in air, a milk intake for pouring in the milk, and adjustable settings for milk flow and air venting. The device operates using an AC motor with belts and pulleys to generate the required frothing action, and it comes with a warning to "STAND CLEAR" during operation.
Name: _____ Draw idea here:  Idea Description: pores inject gas below as mixers mix liquid	4.5	This design showcases a system with multiple vertical mixers placed above a platform labeled "CO2". The mixers appear to be designed to stir or agitate a substance. The idea description suggests that there are pores in the platform that inject gas (CO2) from below while the mixers simultaneously mix a liquid above.	The student's design showcases a device with multiple vertical mixers that stand on a base labeled "CO2." The mixers are intended to agitate and froth the liquid from above. Below these mixers, the base has pores that inject CO2 gas into the liquid, enhancing the frothing process and ensuring rapid milk frothing.
Name: _____ Draw idea here:  Idea Description: Vibrates cup vigorously	3.5	This design illustrates a concept for a cup with a built-in vibration mechanism. The cup features an attachment or compartment labeled "vibrator," suggesting it can shake or stir its contents. Additionally, there's an emphasis on the cup being a "milk holder," possibly indicating its primary use for beverages like milk or shakes.	The student's design presents a "Vibrates Cup Vigorouscup," which appears to be a cup with an integrated vibrator mechanism. This mechanism is designed to rapidly froth milk when it's placed inside the cup. The design also features a milk holder, ensuring that the milk is contained and frothed efficiently within the designated area.

Continued on next page

Table 4 – Continued from previous page

Image	Drawing Score	Please describe this design in 3 sentences.	A student was asked to develop a new innovative product that froths milk in a short amount of time. Please describe this design in 3 sentences.
	2	The design depicts a light bulb placed above a container filled with milk. The principle behind the design is that the longer the light bulb remains on, the hotter the milk beneath it gets. The overarching idea is to create a machine where the milk will eventually boil or bubble due to the heat from the light bulb.	The design features a light bulb-like apparatus that heats and froths milk. The longer the device is turned on, the hotter the milk becomes. The concept is to rapidly boil or create bubbles in the milk, ensuring it froths quickly.
	1.5	This is a sketch of an innovative design labeled “Idea #2.” The drawing depicts a blender-like device that has small holes in its instruments, possibly for fine blending or sieving. The accompanying description suggests it’s a blender with very small holes in the instruments, which might provide a unique blending mechanism or texture to the end product.	The design showcases a unique blender specifically tailored for frothing milk. It features a specialized blending apparatus with very small holes, ensuring thorough frothing within a short duration. The compact and ergonomic design of the blender ensures efficient milk frothing while minimizing spillage.

Discussion: Design Descriptions In this section, we aimed to assess GPT-4V’s ability to match different representations of an early-stage conceptual design and to generate one representation from another. The tasks we generated to explore these capabilities were: match a design sketch with its correct textual description, and generate a textual design description from a sketch. For each of these tasks, we gave various forms of the sketch in order to understand how the amount of handwritten text and the drawing skill in each sketch would affect these results. The quantitative results from the description matching experiment, shown in Table 2, provided a basic understanding of whether the later description generation tasks were possible at all. The results showed that given an entire design sketch including the handwritten text description, GPT-4V was able to match the sketch to the text description for 10/10 of the questions across all three trials. This result essentially just assured us that GPT-4V could comprehend the hand-written text in these drawings.

With this point verified, we next tested description matching if we removed the handwritten text description from the image. In this case, GPT-4V was able to match the sketch to the text description an average of 5.33/10 of the time. While this is still better than random chance, which would be an average of 2.5/10, this result demonstrated how important providing both modalities, text, and image, is in this design stage. We noticed that many of the incorrect answers were GPT-4V selecting the “None of the above” option, suggesting that none of the descriptions matched the design. In fact, 4/5 of the incorrect answers for Trial 1 and 2, and 3/4 incorrect answers for Trial 3 occurred this way. This is sensible to us, as the design sketches are often visually simple compared to their textual descriptions. An example of this is design 5 in Figure 3, which visually looks like a cup of milk, but whose text description is “Centrifuge of milk.” With these results in mind, we tested how well GPT-4V would match the sketch to the description if we removed the “None of the above” option. This led to an improved average of 7/10 correct. One interesting example is shown in Context 2.2.4, in which GPT-4V generated its own option because the model determined that the sketch did not match any of the provided options, despite one being correct.

We further explored GPT-4V’s capability to generate textual descriptions from design sketches. The results are shown in Table 4. We explore how well the model generates design descriptions for five designs with different “drawing

scores,” or level of perceived drawing ability. For each design, we task GPT-4V with generating a description using two different prompts, one with more information:

1. Please describe this design in 3 sentences.
2. A student was asked to develop a new innovative product that froths milk in a short amount of time. Please describe this design in 3 sentences.

Qualitatively, we assess that the model is able to generate useful and accurate text descriptions of designs even for sketches with very low drawing scores. We highlight in green the parts of the description that we believe showed understanding and relevance. GPT-4V extrapolates beyond just the given text to describe both the form and the function of the designs. This result once again highlights the importance of having both the text and image modalities in the design, as the generated descriptions pull heavily from the handwritten text to contextualize the sketch and explain how the design might function. Context 2.2.5 demonstrates GPT-4V’s ability to understand, infer, extrapolate, and generate information about a design from its sketch. For example, the sketch includes a labeled belt and pulley system, and GPT-4V includes this in the generated description: “**Belt and Pulleys:** This part seems to be connected to an AC Motor. The belt and pulley system likely drives the turbine mechanism, converting the motor’s rotational movement into the desired frothing action.” We assess that GPT-4V is able to effectively generate textual descriptions when provided with a detailed sketch including handwritten text. It is important to note that LLMs, like GPT-4V, may hallucinate. This means GPT-4V may perceive patterns or objects in the input that are imperceptible to humans, and therefore produce outputs that do not make sense. Engineers should perform checks on generated descriptions, and additional research should be done regarding generating text descriptions for sketches in many domains. However, from our results, GPT-4V is able to generate sensible text descriptions from early-stage sketched. This ability may help engineers (1) create an easily-searchable catalog of early-stage designs, and (2) more easily generate multimodal datasets of paired sketches and text descriptions.

2.3 Concept Selection

A core component of conceptual design is concept selection [Okudan and Tauhid, 2009, Miller et al., 2021]. There are various concept selection methods, ranging from those based on decision matrices, to uncertainty modeling, to heuristics [Okudan and Tauhid, 2009]. One of the most widely used concept selection methods for engineers is the Pugh Evaluation Matrix, sometimes called the Pugh chart [Pugh, 1991, 1995].

A Pugh Chart, also known as a Pugh Matrix, is a decision-making tool used in engineering and design. It involves comparing multiple options against a set of criteria, using a baseline for reference, to determine the most suitable choice. Each option is scored based on how well it meets each criterion compared to the baseline, facilitating an objective evaluation of alternatives. The first step in creating a Pugh chart is defining selection criteria, which will be used to evaluate and compare concepts. The method may vary, but a common practice is to then select a benchmark design, and score all other designs qualitatively based on how they compare to the benchmark for each of the selection criteria.

In the following experiments, we explore GPT-4V’s ability to both generate selection criteria given a design prompt, and use concept selection methods on design sketches. We utilize a case study presented in the Concept Selection chapter of *Product Design and Development* by [Ulrich et al., 2020]. This case study provides a design task:

“A medical supply company retained a product design firm to develop a reusable syringe with precise dosage control for outpatient use.”

Seven design sketches of reusable syringes are also included along with a Pugh chart and selection criteria. We use these as qualitative benchmarks with which we compare GPT-4V’s ability to generate selection criteria and a Pugh chart.

2.3.1 Generating Selection Criteria

Given a description of a design task, can GPT-4V generate selection criteria that would be used to evaluate designs?

Context 2.3.1: Concept Selection

Prompt 1: (Generating Selection Criteria)

A medical supply company retained a product design firm to develop a reusable syringe with precise dosage control for outpatient use. Please list the selection criteria that should be used to perform early-stage concept selection.

GPT-4V Answer:

Certainly! Early-stage concept selection for a reusable syringe with precise dosage control for outpatient use involves considering various factors to ensure safety, efficiency, and user-friendliness. Here's a list of selection criteria that should be considered:

Safety and Biocompatibility:

- Material used should be biocompatible and non-reactive with the medication.
- The design should prevent overfilling and accidental discharge.
- Features to prevent cross-contamination, especially since it's reusable.

Ease of Use:

- Intuitive design that can be used by patients with varying levels of dexterity.
- Clear markings and indicators for dosage.
- Ergonomic grip and handling.

Precision and Accuracy:

- The syringe should allow for precise measurement of medication.
- Minimal dead space to ensure full dosage delivery.
- Reliable mechanism that ensures consistent dosage every time.

Durability and Reusability:

- Ability to withstand repeated use without wear and tear.
- Resistance to breakage or malfunction.
- Long shelf-life.

Cleaning and Sterilization:

- Ease of disassembly and reassembly for cleaning.
- Material and design should be suitable for common sterilization methods.
- Features to ensure that all parts are adequately cleaned and sterilized.

Cost-effectiveness:

- Affordable production costs without compromising on quality.
- Long-lasting design that offers value over single-use syringes.

Compatibility:

- Compatibility with various types of medications.
- Compatibility with other medical equipment or accessories, like needles or vials.

Environmentally Friendly:

- Made of sustainable or recyclable materials.
- Design should minimize waste generation.

Storage and Portability:

- Compact design for easy storage and transport.
- Features to prevent leakage or contamination during storage.

Scalability:

- Design should be adaptable for mass production.
- Potential for future enhancements or upgrades.

...

2.3.2 Creating a Pugh Chart

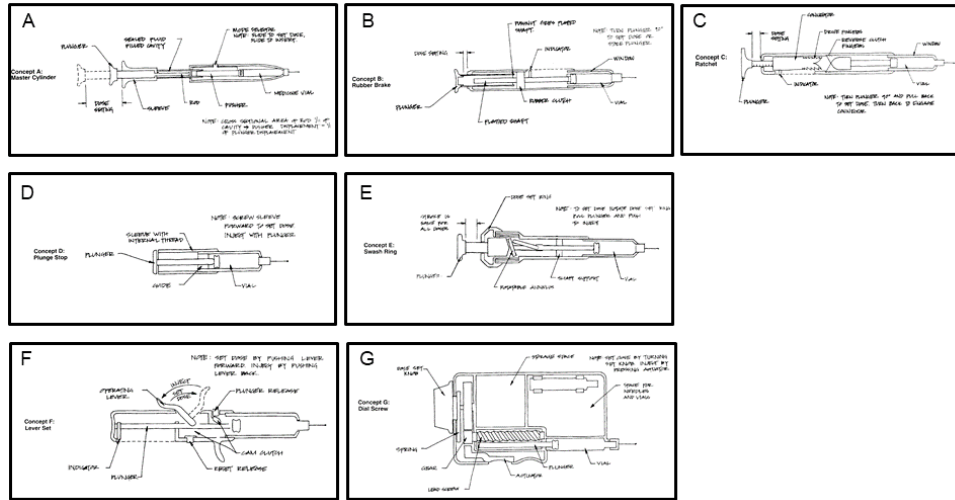
Given selection criteria, can GPT-4V analyze and evaluate designs and format this evaluation in a Pugh chart?

Context 2.3.2: Concept Selection

Prompt 1: Generating a Pugh Chart

Given concepts A-G and the following selection criteria, make a Pugh chart to select a concept. The reference concept is concept D.

- Selection criteria:
- Ease of handling
 - Ease of use
 - Readability of settings
 - Dose metering accuracy
 - Durability
 - Ease of manufacture
 - Portability



Images reproduced with permission from the authors.

GPT-4V Answer:

Creating a Pugh chart involves comparing various concepts against a reference concept across different selection criteria, and assigning scores to indicate whether the concept is better, worse, or the same as the reference. Here's an explanation on how to create a simplified Pugh chart, given the selection criteria you've provided:

- Set up the matrix:** List the concepts (A-G) in rows and the criteria (ease of handling, ease of use, etc.) in columns.
- Choose a reference concept:** In this case, Concept D is the reference and should be marked with a "0" (zero) score across all criteria since it's the standard for comparison.
- Score the concepts:** Evaluate each concept against the reference for each criterion. You can use "+1" for better than the reference, "0" for the same as the reference, and "-1" for worse than the reference.
- Total the scores:** Sum the scores for each concept.

Since I can't create a visual Pugh chart directly within this platform, I'll outline the matrix in text form:

Concept/Criteria	Ease of Handling	Ease of Use	Readability of Settings	Dose Metering Accuracy	Durability	Ease of Manufacture	Portability	Total
Concept A								
Concept B								
Concept C								
Concept D (Ref)	0	0	0	0	0	0	0	0
Concept E								
Concept F								
Concept G								

Table 5: Selection criteria provided by the *Product Design and Development* textbook and the GPT-4V equivalent.

Textbook Selection Criteria	GPT-4V Equivalent
Ease of handling	Ease of use: <i>Ergonomic grip and handling</i>
Ease of use	Ease of use
Readability of dose settings	Ease of use: <i>Clear markings and indicators for dosage</i>
Dose metering accuracy	Precision and accuracy
Durability	Durability and reusability
Ease of manufacture	Cleaning and Sterilization: <i>Ease of disassembly and reassembly for cleaning</i> Scalability: <i>Design should be adaptable for mass production</i>
Portability	Storage and Portability

Discussion: Concept Selection Through these experiments, we explored GPT-4V’s ability to perform two common concept selection tasks: generating selection criteria given a design task and making a Pugh chart given several design concepts. We found that GPT-4V was able to assess a design task and generate many relevant selection criteria, as shown in Context 2.3.1. For example, given this design task “A medical supply company retained a product design firm to develop a reusable syringe with precise dosage control for outpatient use,” GPT-4V generated selection criteria such as “Safety and Biocompatibility,” “Ease of Use,” and “Precision and Accuracy.” These criteria highlight that the design must be user-centered and safe in healthcare settings.

As a baseline to assess the generated criteria, we utilized the selection criteria and Pugh chart provided in [Ulrich et al., 2020]; these were made for this design task and used the same seven design concepts that we used. Table 5 shows the baseline selection criteria provided for the design task, as well as the equivalent selection criteria generated by GPT-4V, if applicable. We observe that for each of the seven criteria that the baseline Pugh chart used, GPT-4V outputs an equivalent criterion. It is important to note that some of GPT-4V’s equivalent criteria were subcategories, such as GPT-4V’s “Ease of Use: Clear markings and indicators for dosage,” which mapped to the textbook’s “Readability of dose settings.” These results demonstrate that GPT-4V is able to generate many relevant selection criteria, but that an engineer should still read the raw output and select relevant criteria as well as separate subcategories of certain criteria.

When tasked with creating a Pugh chart given selection criteria and several designs, we found that GPT-4V understood what a Pugh chart was and how to generate one, however, it was often reluctant to create one given the limited information, as shown in Context 2.3.2. GPT-4V was able to generate an empty Pugh chart with the correct matrix format (not shown in Context 2.3.2, which switched the typical rows and columns), and also understood that it would be filled with qualitative comparisons of concepts, with one reference concept. However, it would only fill in the Pugh chart with hypothetical values given the lack of information about each concept. Perhaps if an engineer provided more information about each concept, GPT-4V would have been able to generate an accurate Pugh chart, however, it failed to do so within our task format.

Overall, our findings suggest that GPT-4V can be potentially effective in assisting human designers in identifying key factors that should be considered in the design process. However, while GPT-4V can generate equivalent criteria to those used in traditional methods, its outputs may need refinement, such as categorizing subcriteria. In terms of creating a Pugh chart, GPT-4V understands the concept and can format the chart correctly, but its reluctance to fill in the chart without extensive information indicates a limitation. This suggests that while GPT-4V can be a useful tool for structuring and initiating the concept selection process, human input remains crucial for detailed analysis and decision-making. The implications for practitioners are clear: VLMs such as GPT-4V can be a valuable aid in the initial stages of design concept evaluation, but they may require careful oversight and additional information to realize their full potential in more complex decision-making tasks.

3 System-Level And Detailed Design

Overview and Motivation The generation of system-level and detailed designs is the phase of the product development process where full designs are realized, iterated upon, evaluated, and prepared for manufacturing. This phase of the design process envelops steps that are performed in parallel, such as outlining product architecture, prototyping, and developing robust designs [Ulrich et al., 2020]. The generation of system-level and detailed designs draws upon many skill sets and much knowledge, including computer modeling, spatial reasoning, and physics-based knowledge. While it is difficult to comprehensively assess GPT-4V’s ability to help with this phase of the design process, we consider the following case study when crafting our testing methodology.

Imagine Sarah, an experienced mechanical engineer, embarking on the design of a new lightweight bicycle frame. She begins with material selection, for which she must cross-reference technical charts to identify materials that balance strength, weight, and sustainability. Next, she delves into the creation of detailed CAD models and engineering drawings, requiring spatial reasoning and physics knowledge. Drawing upon her experience in design optimization, Sarah then explores design alternatives through topology optimization, seeking the most efficient material layout. Finally, she evaluates her design’s performance, simulating various conditions by utilizing special physics-based modeling software. Her project encapsulates the essence of system-level and detailed design, highlighting the need to examine AI’s role in four critical tasks: material selection, CAD and engineering drawings generation, topology optimization, and design performance evaluation. In her journey, Sarah encounters challenges where AI’s capabilities could be pivotal. We use this scenario to inspire our test bed for studying the effectiveness of VLMs in the system-level and detailed design phase.

Specifically, we focus on four main tasks in the system-level and detailed design phase that are frequently performed by engineers such as Sarah. These tasks are material selection (section 3.1), generation of CAD and engineering drawings (section 3.2), topology optimization for exploration of design alternatives (section 3.3), and design performance evaluation (section 3.4).

Evaluation We perform the following evaluations of GPT-4V’s capabilities:

- (i) **Material Selection 3.1** Can GPT-4V effectively assist in material selection based on property charts and design requirements?
 - We find that GPT-4V can be helpful at pointing out material families that meet general specifications, but that it struggles to identify materials that match specific numerical requirements.
- (ii) **Engineering Drawing Analysis 3.2.1** How accurately can GPT-4V extract and interpret information from engineering drawings?
 - Only 11% of the time was GPT-4V successful in describing a block with a blind hole part. Once told it was a block with a blind hole, GPT-4V was had an average performance score of 95% in extracting all dimensions from the engineering drawing and assigning the dimensions appropriate names.
- (iii) **CAD Generation 3.2.2** What is GPT-4V’s proficiency in generating and iteratively improving CAD scripts from engineering drawings?
 - Preliminary findings suggest that GPT-4V struggles with CAD generation of the block-with-blind-hole part, as it only succeeded once in nine attempts to generate correct CAD on the first try. Also, its iterative ability for CAD correction appears limited, as it doesn’t successfully correct incorrect CAD in subsequent iterations.
- (iv) **Topology Optimization 3.3** How well does GPT-4V comprehend and apply topology optimization principles in design?
 - This subsection evaluates GPT-4V’s comprehension of structural topology optimization, relying primarily on image-based information. The model’s expertise is tested across various dimensions: general and specific knowledge of the subject, capability in quantitative analysis, and proficiency in optimizing topologies, including those with potential flaws rendering them impractical. Our evaluation indicates that GPT-4V possesses a basic understanding of topology optimization. It demonstrates competency in articulating both general and nuanced aspects of optimized designs and constraint configurations, particularly when engaged as a field expert. Nevertheless, the model’s performance is lacking in quantitative reasoning tasks, such as calculating material usage.
- (v) **Fluid Dynamics Simulation 3.4** To what extent can GPT-4V analyze and interpret fluid dynamics simulation results?
 - This segment explores GPT-4V’s understanding of fluid dynamics simulation with minimal textual guidance. The assessment focuses on its ability to differentiate between various flow regimes such as

laminar, transient, and turbulent, and its skill in discussing key dimensionless quantities that characterize fluid states. The analysis reveals that GPT-4V possesses a basic understanding of fluid dynamics. It effectively interprets computational fluid dynamics (CFD) results and accurately identifies critical parameters like Reynolds and Mach numbers. The model demonstrates a nuanced understanding of fluid states, correctly interpreting and responding to questions about transient (not fully turbulent) simulation outputs.

3.1 Material Selection

Oftentimes, the selection of material comes early on in the detailed design phase, as material choice informs both the design and the utilized manufacturing method. Material selection requires balancing various constraints and requirements, such as material strength, stiffness, cost, density, embodied energy, electrical resistivity, and thermal conductivity [Ashby, 2016]. Choosing a material that meets an extensive list of requirements and constraints often requires cross-referencing multiple tables or charts, such as Ashby charts [Ashby, 2016]. Ashby charts enable engineers to visually represent the trade-offs between various material properties for different families of materials and many computational methods have been used to explore constraints imposed by them Jetton et al. [2023], Cranford [2023]. Provided with these charts for different material properties, LLMs have the potential to condense material information and to identify materials that meet certain criteria. Several groups have explored GPT’s ability to assist with material considerations. Saka et al. [2023] have used the GPT API to integrate ChatGPT into the building information modeling process to assist with material selection for components of a building. Makatura et al. [2023] looked at GPT-4’s ability to propose a manufacturing process for a part based on material selected. Buehler [2023] trained a model, MeLM (Mechanics Language Model), which was used for material-related tasks, like proposing microstructure designs that meet certain stress-strain responses. In this section, we conduct three independent experiments involving Ashby charts and material selection. To analyze the consistency of responses, each experiment is repeated three times.

3.1.1 Ashby Chart Look-Up

Methodology For this experiment, we provide GPT-4V with a density vs. Young’s modulus Ashby chart. We ask it to identify materials that meet certain density and Young’s modulus requirements. The purpose of this experiment is to assess whether GPT-4V can perform a simple “look-up” of feasible materials from the chart. We repeat this experiment three times. One of the repetitions can be seen in Context 3.1.1.

Context 3.1.1: Ashby Chart Look Up

Prompt 1: (Identifying materials that meet stiffness and density constraints.)

This is an Ashby chart used for the selection of materials in engineering design. I am looking for a material that has a density from 7-10 $\frac{\text{Mg}}{\text{m}^3}$ and a Young's Modulus greater than 100 GPa. Based on the chart, which materials meet these constraints?

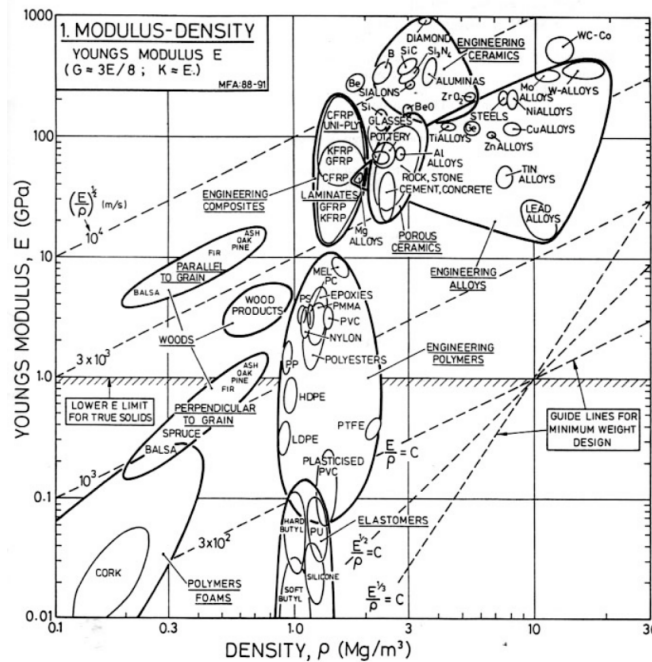


Image from Drela et al. [2005a].

GPT-4V Answer:

Based on the Ashby chart you've provided, the materials that meet the constraints of having a density between 7–10 Mg/m^3 and a Young's Modulus greater than 100 GPa are located in the top right quadrant of the graph. Here's how to interpret the chart for your specified ranges:

1. Identify the horizontal band corresponding to a Young's Modulus (E) greater than 100 GPa. This is above the horizontal line marked with 100 GPa on the left-hand side vertical axis.
2. Locate the vertical band corresponding to a density (ρ) between 7-10 Mg/m^3 . This band lies between the vertical lines marked 7 and 10 on the horizontal axis at the bottom.

The materials that fall within the overlap of these two bands are your candidates. From the chart, it looks like the following materials fall within this range:

- Steels (various types)
- Titanium alloys (Ti alloys)
- Nickel alloys (Ni alloys)
- Tungsten carbide-cobalt composites (WC-Co)

These materials are marked within ovals or circles and fall within the intersection of the specified Young's Modulus and density ranges. It is important to note that the exact positioning within the oval represents variations within the material category, so specific grades or compositions of steel, titanium alloys, nickel alloys, and WC-Co may need to be looked up to ensure they meet the exact requirements of your application.

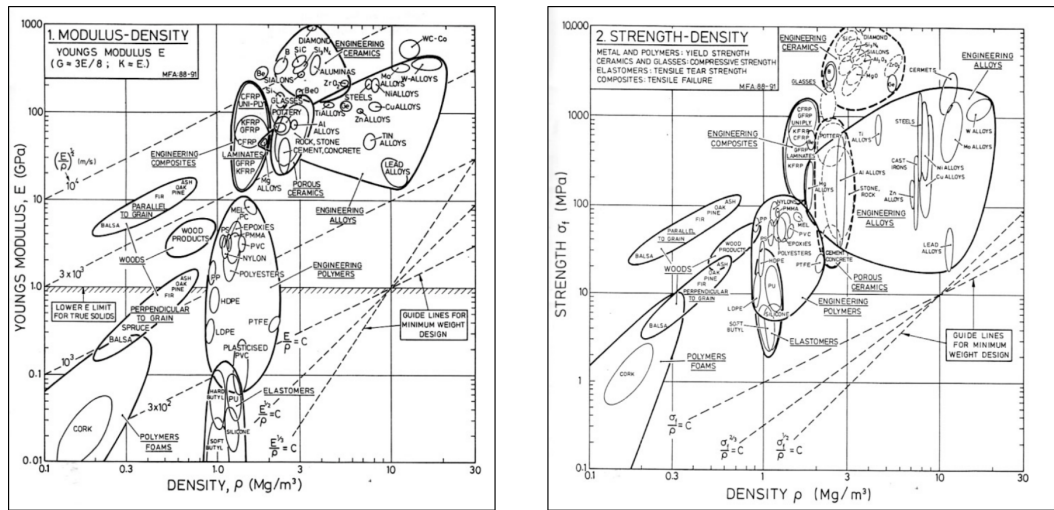
3.1.2 Ashby Chart Cross-Referencing

For this experiment, we give GPT-4V two Ashby charts, one showing density vs. Young's modulus and another showing density vs. strength. We then ask GPT-4V to cross-reference the two charts, identifying materials that meet certain specifications for all three material properties. The purpose of this experiment is to understand if GPT-4V can synthesize information from two material charts together. We repeat this experiment three times. One of the repetitions can be seen in Context 3.1.2.

Context 3.1.2: Ashby Chart Cross-Referencing

Prompt 1: (Identifying materials that meet stiffness, strength, and density constraints.)

These are two Ashby charts that are used for the selection of materials in engineering design. I am looking for a material that has a density from 1.0 to 3.0 Mg/m³, a Young's Modulus between 0.01 and 0.1, and a strength of 3 MPa. Based on the provided charts, which materials meet these constraints?



Images from Drela et al. [2005a] and Drela et al. [2005b].

GPT-4V Answer:

Based on the Ashby charts provided:

For the modulus-density chart on the left, materials that fall within a density of 1.0 to 3.0 Mg/m³ and a Young's Modulus between 0.01 and 0.1 GPa are located in the bottom center portion of the graph. From the chart, it seems that 'Polymers Foams' and 'Cork' could potentially meet these criteria.

For the strength-density chart on the right, you are looking for materials with a density from 1.0 to 3.0 Mg/m³ and a strength of 3 MPa. The materials that meet these criteria would be plotted around the 3 MPa line, which is not explicitly drawn on this graph. However, materials just above the 'Polymers Foams' region could potentially meet the strength requirement of 3 MPa. It's important to note that the exact overlap between the modulus and strength requirements cannot be determined precisely from these charts without more detailed scales, but 'Polymers Foams' appear in both regions of interest and might be a good starting point for further investigation. It's always recommended to check the detailed properties of these materials from a materials database or literature to ensure they meet the specific engineering requirements.

3.1.3 Material Selection for a Beam

In this experiment, we ask GPT-4V to help us select a material for a hypothetical beam, given general beam requirements. The purpose of this experiment is to understand if GPT-4V can translate the general requirements into material requirements and propose appropriate material families based on those requirements. We repeat this experiment three times. One of the repetitions can be seen in Context 3.1.3. A summary of all the Material Selection experiments can be found in Table 6.

Context 3.1.3: Material Selection for a Beam

Prompt 1: (Selecting materials based on design requirements.)

I am looking to select a material for a beam I am designing. I want the beam to be both stiff and light. Attached is an Ashby chart, which can be used for the selection of materials in engineering design. Based on this chart and the information provided, can you suggest materials that could be used for the beam?

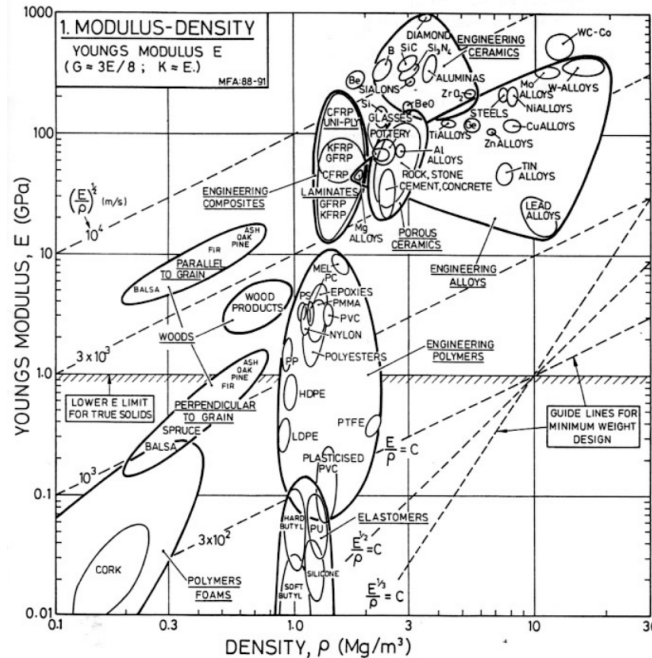


Image from Dreha et al. [2005a].

GPT-4V Answer:

The Ashby chart you've provided is used to select materials based on their stiffness (Young's modulus) and density. For a beam that is both stiff and light, you would want materials that have a high Young's modulus to density ratio. These materials would appear towards the top left of the chart, as this region represents materials with high stiffness and low density.

On the chart, materials are clustered into groups. The "Engineering Composites" group, which includes materials like CFRP (Carbon Fiber Reinforced Plastic) and GFRP (Glass Fiber Reinforced Plastic), stands out as particularly suitable for applications requiring high stiffness and low weight. Composites are known for their excellent strength-to-weight ratios and are commonly used in aerospace and automotive applications for this reason.

Another group that might be relevant is the "Engineering Alloys," especially those towards the left of the group indicating lower density. This could include certain aluminum alloys, which are known for their use in lightweight structures where stiffness is also a requirement.

To narrow down the selection further, you would consider additional factors such as cost, manufacturing processes, environmental resistance, and compatibility with other materials used in the design. It would also be important to ensure the selected material can withstand the specific loading conditions of your beam design.

Discussion: Material Selection Overall, we conclude that GPT-4V performs well at identifying broad material families that exhibit general properties (e.g. low density), but performs less well when given specific requirements or constraints (e.g., density between 1.0 and 3.0). This finding is illustrated by the results of the three experiments. The responses from all experiments and repetitions can be seen in Table 6. One repetition of each experiment is displayed in-full in Contexts 3.1.1, 3.1.2, and 3.1.3. For the Ashby Chart Look-Up experiment (see Context 3.1.1), we would expect the correct answer – materials that have a density between 7 and 10 $\frac{Mg}{m^3}$ and that have a Young's modulus greater than 100 GPa – to be steel, Ni-alloys, and Cu-alloys. (Zn-alloys and Mo-alloys lie on the border of the feasible region). For all

Table 6: Summarized results for the GPT-4V material selection experiments.

Repetition #	Ashby Chart Look-Up	Ashby Chart Cross-Referencing	Material Selection for a Part
Rep 1	Steels Ni-alloys Ti-alloys WC-Co	Polymer Foams	Top left of chart Engineering Composites Engineering Alloys
Rep 2	Steels Ti-Alloys	*Polymers Foams Woods w/ Lower Densities	Top left of chart Engineering Composites Engineering Ceramics Engineering Alloys Wood Products
Rep 3	*Steels Ni-Alloys WC-Co	*Polymers Woods Polymer Foams	Top left of chart Engineering Composites Engineering Ceramics Engineering Alloys Wood Products

*For these responses, GPT-4V expressed concern about making definitive material choices based on the resolution of the provided Ashby charts.

three repetitions, GPT-4V correctly answered that steels would be feasible materials. For two out of the three times, it also mentioned that Ni-alloys would meet the specified requirements. However, in all three iterations, it also included materials in its answer – either Ti-alloys or WC-Co – that do not meet our specifications; Ti-alloys have a density less than $7 \frac{Mg}{m^3}$ and WC-Co materials have a density greater than $10 \frac{Mg}{m^3}$. In none of the iterations did GPT-4V mention Cu-alloys in its answer, although this material group meets both the stiffness and density specifications.

GPT-4V performed poorly in our Ashby Chart Cross-Referencing Experiment (see Context 3.1.2 for the full response from one of the repetitions). The correct answers to the question were the soft butyl and elastomer materials, which have densities between 1.0 to $3.0 \frac{Mg}{m^3}$, Young’s moduli between 0.01 and 0.1 , and a strength of 3 MPa. Across the three repetitions, GPT-4V never identified either of these materials as meeting our requirements. Overall, it tends to conclude that polymer foams, foams, polymers, or woods would be suitable choices, but these materials do not meet our specifications. Polymer foams, for example, don’t meet the density requirement; many polymer foams have densities between 0.1 and $0.3 \frac{Mg}{m^3}$, suggesting that GPT-4V possibly confuses our $1.0 - 3.0 \frac{Mg}{m^3}$ density specification with this 0.1 to 0.3 range. It is important to note that for two of the three repetitions of this experiment, GPT-4V was hesitant to provide an answer due to the “resolution” of the provided images.

Our Ashby Chart Look-Up and Cross-Referencing experiment reveals possible areas of improvement needed in handling precise numerical data and complex information synthesis. The model’s struggle with accurately interpreting numerical constraints, as evidenced by these two experiments, highlights a shortfall in applying exact numerical ranges. Furthermore, its inability to effectively cross-reference and synthesize data from multiple sources underscores a challenge in processing multi-dimensional information. This issue is particularly pertinent in engineering where precision and multi-faceted data analysis are crucial.

GPT-4V performs much better when asked to propose potential material families for a beam that needs to be both lightweight and stiff (see Context 3.1.3). For all three experiment repetitions, GPT-4V correctly translates the stiffness specification into a high Young’s modulus requirement and the light-weight specification into a low-density requirement. For all trials, it correctly asserts that materials we would want to consider are towards the top left of the provided Ashby chart and proposes engineering composites and engineering alloys (and for two of the three repetitions, also proposes engineering ceramics and wood products). We conclude that while GPT-4V struggles to identify materials that meet specific numerical requirements, it is much better at proposing material families that meet general specifications.

Overall, the use of GPT-4V in material selection in engineering design showcases its potential as a supportive tool in the preliminary stages of decision-making and as an educational aid in materials science. Its ability to suggest material families based on general requirements can streamline the initial phases of design, allowing engineers to focus on finer details. This integration points towards a future where AI complements traditional engineering tools, enhancing the efficiency of design workflows. However, it also raises important ethical and practical considerations, such as over-reliance on VLM models without knowing their limitations and ensuring AI-generated recommendations of material align with safety standards and environmental concerns. Transitioning from the exploration of GPT-4V’s

capabilities in material selection, the research now shifts focus to another critical aspect of engineering design: VLM’s ability to interpret complex engineering drawings and contribute to the generation of Computer-Aided Design (CAD) models.

3.2 Engineering Drawing Analysis and CAD Generation

A critical step of the detailed design process is the generation of 3D models. Computer-aided design (CAD) software enables the creation of detailed solid models, allowing engineers to encode precise part dimensions and assembly relations between parts [Nyemba, 2022]. These CAD models pave the way for design for manufacturing, since detailed engineering drawings with manufacturing specifications are typically created from the 3D models [Nyemba, 2022]. CAD models are also useful for the different ways in which designs and parts can be visualized (e.g. cross-sections, wireframe views, etc.), enabling engineers to easily consider different aspects of their design [Nyemba, 2022]. We hypothesized that GPT-4 with vision would be better able to assist with CAD generation and engineering drawing analysis than GPT-4, since these two design forms – CAD models and engineering drawings – are inherently visual mediums.

We gain inspiration from the work of researchers who have explored the potential of GPT to assist with converting text into CAD [Makatura et al., 2023, Nelson et al., 2023]. For example, Makatura et al. devoted a large section of their work to the exploration of GPT-4’s ability to generate CAD designs from text. They looked at text to scripts that would generate 2D designs (DXFs and SVGs), demonstrating relative success in the design of 2D pieces of a cabinet. Makatura et al. [2023] then performed several case studies to illustrate GPT-4’s ability to convert text into scripts for 3D designs, using both CSG-based CAD languages and sketch-based CAD languages. These experiments showcased mixed success, oftentimes requiring prompts to be engineered with specific function signatures. The authors noted reasoning challenges, specifically when it came to spatial reasoning. They also cited iterative ability as both a capability and a drawback of GPT-4, as they found the model was sometimes able to correct errors through continued chat iterations but that more iterations also led to limited memory of previous information in a chat. However, a key limitation of past work is that it relied on text-only LLMs, while CAD is inherently a task that has significant visual aspects. In our study, we focus on evaluating the capabilities of VLMs.

To assess GPT-4V’s ability to analyze engineering drawings and generate CAD, we utilized a two-part experiment framework. An example of a full experiment can be seen in Context 3.2.1. The first part of each experiment assesses GPT-4V’s ability to analyze engineering drawings. In this part of the experiment, we test the model on two aspects: 1) its ability to describe a part based on an engineering drawing and 2) its ability to extract dimensions from an engineering drawing. In the second part of each experiment, we evaluate the model’s ability to generate a script that encodes the CAD of a part. We ask the model to do this based on the previously provided engineering drawing, the previously extracted part dimensions, and a CAD scripting language that we specify. In this part of the experiment, we score the model on the CAD that its script generates. If the CAD is not correct on the first attempt, we feed it back views of the generated CAD and ask it to iterate to fix any discrepancies it sees between the generated CAD and the original engineering drawing. We repeat this iterative process until five different CAD generation attempts have been made.

In total, we ran nine experiments, each conducted sequentially in a single chat context. Three groups of three experiments are identical (conducted for repetition), and the difference between the three groups of experiments is in the CAD scripting language specified. We now further elaborate on the experiment structure and the method for scoring each experiment:

3.2.1 Engineering Drawing Analysis

Methodology: The prompts for this part are identical across all nine experiments.

- **Part Description from an Engineering Drawing** - Prompt 1 (P1). GPT-4V is given an engineering drawing of a block with a blind hole, as seen in Context 3.2.1 P1. We chose to use this block-with-blind-hole part as the subject of our experiments since it represents one of the most basic yet functional parts that can be created using CAD, necessitating only two sketches and basic cut/extrude operations. The drawing follows typical engineering drawing conventions and was created in an undergraduate-level engineering course.⁴ As such, it is a drawing that we would expect undergraduate-level mechanical engineers to readily understand.
 - Scoring (1 point possible): We assign 1 point if GPT-4V correctly mentions that the part is a “block with a hole” or “block with a blind hole.” Any mention of a “through” hole receives no points as it shows an incorrect understanding of the underlying geometry.

⁴The engineering drawing of the block with blind hole shown in Context 3.2.1 was created as an assignment for the “ENME272: Introduction to Computer Aided Design” course at the University of Maryland.

- **Dimension Extraction from an Engineering Drawing** - Prompt 2 (P2). We next ask GPT-4V to extract the dimensions shown in the engineering drawing, assigning them appropriate names. We specifically ask GPT-4V to not extrapolate to create dimensions that are not explicitly called out in the drawing.
 - Scoring (10 points possible): 1 point is awarded for each of the five numbers shown on the drawing (8.00, 5.00, 12.00, 4.00, and $\varnothing 5.00$) that GPT-4V successfully extracts. Another point is awarded for each of the five dimensions to which it assigns an appropriate name. For the block dimensions — 8.00, 5.00, and 12.00 — we accept any assignment of [length, width, height] or [depth, width, height] to the three dimensions since the assignment of these labels depends on how the block is oriented. For the 4.00 dimension and the $\varnothing 5.00$ dimension, we expect labels of “hole depth” and “hole diameter” respectively, or equivalent names. For any dimensions GPT-4V lists beyond those shown in the drawing, we subtract 1 point for not following instructions.

The results from the Engineering Drawing Analysis evaluation can be seen in Table 7.

3.2.2 CAD Generation Using CADQuery, FeatureScript, and OpenSCAD

Methodology: The prompts for this part vary across the nine experiments as indicated below.

- **CAD Generation 1** — Prompt 3 (P3). Continuing in the same context window where P2 left off, we correct any dimensions that GPT-4V extracted from the drawing incorrectly, and we then ask GPT-4V to generate a CAD script of the block-with-hole part based on the engineering drawing provided in P1 and the dimensions it extracted in P2. For three of the experiments (experiments 1 – 3), we ask GPT-4V to do this using the CadQuery scripting language; for another three experiments (experiments 4 – 6), we ask GPT-4V to do this with a different scripting language, FeatureScript; and for the last three experiments (experiments 7 – 9), we ask GPT-4V to use the CAD scripting language OpenSCAD. Note that each language offers unique features and advantages:
 - CadQuery: An open-source CAD scripting module built in Python, CadQuery is easy-to-use for those already familiar with Python.
 - FeatureScript: The scripting language of the CAD software Onshape – a free cloud-based CAD software – FeatureScript is integrated into Onshape, enabling both traditional CAD modeling and custom, script-defined, parametric modeling.
 - OpenSCAD: Another open-source CAD scripting language built in C++, OpenSCAD is integrated into the CAD software FreeCAD and provides granular control over a model.

By utilizing these three scripting languages, our research aims to comprehensively assess GPT-4V’s ability to adapt to different CAD scripting environments and to evaluate its versatility in translating engineering drawings into functional CAD models.

- Scoring (6 points possible): We assign 1 point if the generated script has no errors when run. We award 1 point for each of the following features that the generated CAD possesses: the block has correct dimensions; the CAD has a hole on the largest block face; the hole is centered on a face; the hole has the correct depth; and the hole has the correct diameter. We subtract a point for each extra, incorrect feature that is present in the generated CAD (e.g. a second hole, a cutout in the block, etc.).
- **CAD Generation 2 - 5:** Prompts 4 - 7 (P4 - P7). If the CAD generated by the previous prompt has a syntax error when the code is run, we provide it to GPT-4V and ask it to fix the script. If the script runs but the generated CAD doesn’t have a perfect score, we ask GPT-4V to correct discrepancies between the generated CAD and the engineering drawing. We ask it to do this by providing it an image with four views of the CAD generated from the previously provided script (see 3.2.1 P4 for an example). These views show the CAD with hidden lines and coordinate systems visible for each view. If the CAD is still not receiving a perfect score by P7 (CAD Generation 5), we end the experiment.
 - Scoring (6 points possible for each prompt, P4 – P7): Scoring for these prompts is identical to the scoring for P3.

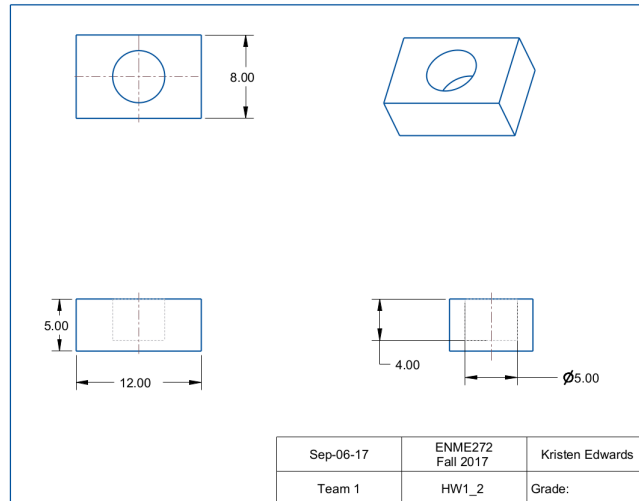
The results from the CAD Generation evaluation can be seen in Figure 5.

Please note that the scoring system used here primarily serves as an illustrative example to simplify the understanding of an aggregate score for readers. Practitioners may adjust the weights assigned to each question type based on specific requirements. We have made the raw data for each response available, facilitating benchmarking and allowing for flexibility in evaluating performance when varying the importance of different factors. The results from Part 1, Engineering Drawing Analysis, can be seen in Table 7 while the results from Part 2, CAD Generation, can be viewed in Figure 5.

Context 3.2.1: Experiment 1—Engineering Drawing Analysis and CAD Generation Experiment

Prompt 1: (P1 - Part Description from an Engineering Drawing)

This is an engineering drawing of a part, showing four views of the part. Can you tell me what the part is?

**GPT-4V Answer:**

Certainly! Based on the provided engineering drawing, the part depicted is a rectangular block (or prism) with a cylindrical through-hole. Here's a breakdown:

1. The top-left view shows a top view of the part. We can see a circle, which indicates a cylindrical hole through the part. The dimensions indicate that the length of the rectangular block is 12.00 units and the width is 5.00 units.
2. The bottom-left view is a front view, where the height of the block is shown to be 8.00 units.
3. The bottom-right view is a right-side view, showing the height of the block as 4.00 units and the diameter of the cylindrical hole as 5.00 units.
4. The top-right view is an isometric or 3D view that gives a perspective of how the part looks in 3D.

So, to summarize, it's a rectangular block with dimensions 12.00 units x 5.00 units x 8.00 units with a cylindrical hole of diameter 5.00 units running through its shorter width (5.00 units).

Cont.—Experiment 1—Engineering Drawing Analysis and CAD Generation Experiment

Prompt 2: (P2 - Dimension Extraction from an Engineering Drawing)

The part depicted in the engineering drawing is a block with a blind hole. Based on the engineering drawing views, can you please extract all dimensions of the part that are shown on the drawing, assigning each dimension an appropriate name? Only list dimensions that appear in the drawing; do not extrapolate from them to create more dimensions.

GPT-4V Answer:

Of course! Based on the provided engineering drawing, here are the dimensions of the part:

1. Length of the Block: 12.00 units (Shown on the bottom-left view)
2. Width of the Block: 5.00 units (Both top-left and bottom-left views indicate this width)
3. Height of the Block (from top-left view): 8.00 units
4. Height of the Block (from bottom-right view): 4.00 units (This is likely the depth of the blind hole, as it doesn't span the entire height)
5. Diameter of the Blind Hole: Ø5.00 units (Shown on the bottom-right view)

From the information provided, the hole is a blind hole because it doesn't go all the way through the block, as indicated by the different heights in the top-left and bottom-right views.

Prompt 3: (P3 - CAD Generation 1)

You are correct that the height of the block is 8.00 units, but it is not correct to say that the height of the block is also 4.00 units. The 4.00 unit dimension represents the depth of the blind hole. Now that you have extracted the dimensions and given that you still have the engineering drawing, please use the CAD scripting language, CadQuery, to generate a script that encodes the CAD of the part.

GPT-4V Answer:

Of course, I apologize for the oversight. Let's proceed with creating a CadQuery script for the part based on the provided engineering drawing.

Here's a CadQuery script to generate the CAD model of the part:

```
\hl{import cadquery as cq}

# Parameters from the drawing
length = 12.00
width = 5.00
height = 8.00
hole_diameter = 5.00
hole_depth = 4.00

# Create a block with the specified dimensions
block = cq.Workplane("XY").box(length, width, height)

# Create a blind hole in the block
block_with_hole = block.faces(">Z").workplane().circle(hole_diameter / 2).cutBlind(-hole_depth)

# Export result or display, depending on your setup
# block_with_hole.exportStl("block_with_hole.stl")

show_object(block_with_hole)
```

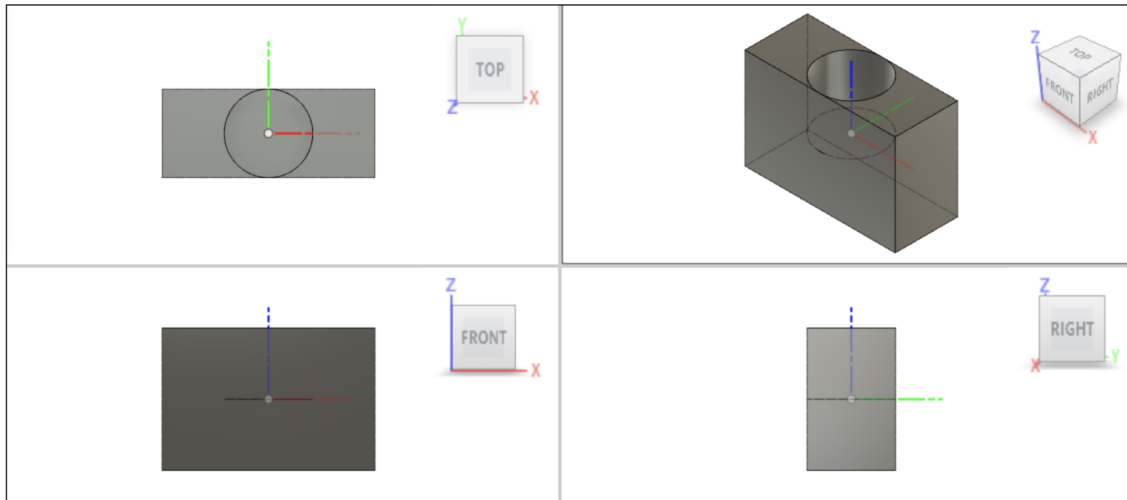
You would need to have CadQuery installed and integrated with your CAD system to run the script. The “show_object” function is just a placeholder; in a real environment, you'd use the appropriate function to visualize or export the result, such as the commented “exportStl” function.

When you run this script with CadQuery, it will generate a block with the specified dimensions and a blind hole according to the drawing.

Cont.—Experiment 1—Engineering Drawing Analysis and CAD Generation Experiment

Prompt 4: (P4 - CAD Generation 2)

That's not quite right – the CAD of the part created using the OpenSCAD script you just gave me does not match the part shown in the original engineering drawing I provided you. The attached photo shows four views of the part's CAD that was created using the CadQuery script you just provided. Each view shows a different orientation of the CADed part, as seen by the coordinate system shown in each view. The views show the CADed part in a shaded-with-hidden-edges style. Based on these CAD views, can you update the CadQuery script to fix any issues that you see, so that the part created using the script matches the part shown in the original drawing I provided?



GPT-4V Answer:

Certainly, I apologize for the oversight. Let's correct the CadQuery script.

From the CAD views you provided and comparing it with the original engineering drawing, I observe the following:

1. The hole is a through-hole and not a blind hole. This means the depth of the hole is equivalent to the height of the block.
2. The hole is off-centered in both X and Y directions.

Let's create a new CadQuery script with these corrections:

```
import cadquery as cq

# Parameters from the drawing
length = 12.00
width = 5.00
height = 8.00
hole_diameter = 5.00
hole_depth = height # making it a through-hole

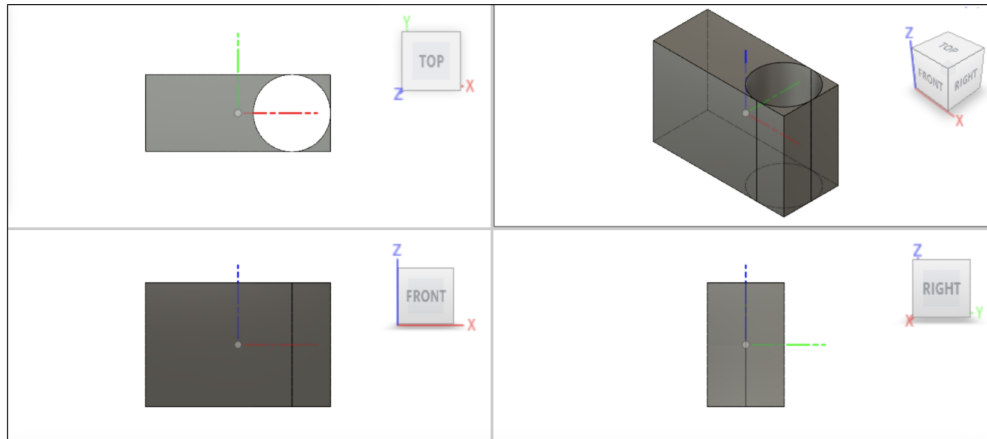
# Assuming half the width of the block minus half the hole diameter for the Y offset (based on drawing)
offset_y = (width - hole_diameter) / 2
# Assuming half the length of the block minus half the hole diameter for the X offset (based on drawing)
offset_x = (length - hole_diameter) / 2
# Create a block with the specified dimensions
block = cq.Workplane("XY").box(length, width, height)
# Create a through-hole in the block at the specified offsets
block_with_hole = block.faces(">Z").workplane().center(offset_x, offset_y).circle(hole_diameter / 2).cutThruAll()

# Export result or display, depending on your setup
# block_with_hole.exportStl("block_with_hole_corrected.stl")
show_object(block_with_hole)
```

The hole is now centered at "offset_x" and "offset_y" from the left-bottom corner. It's a through-hole, as observed from the CAD views. Run this corrected CadQuery script to get a more accurate representation of the part shown in the original engineering drawing.

Cont.—Experiment 1—Engineering Drawing Analysis and CAD Generation Experiment*Prompt 5: (P5 - CAD Generation 3)*

That's not quite right – the CAD of the part created using the OpenSCAD script you just gave me does not match the part shown in the original engineering drawing I provided you. The attached photo shows four views of the part's CAD that was created using the CadQuery script you just provided. Each view shows a different orientation of the CADed part, as seen by the coordinate system shown in each view. The views show the CADed part in a shaded-with-hidden-edges style. Based on these CAD views, can you update the CadQuery script to fix any issues that you see, so that the part created using the script matches the part shown in the original drawing I provided?

**GPT-4V Answer:**

Certainly! Based on the provided CAD views, let's identify and rectify the discrepancies between the CAD views and the original engineering drawing:

1. Top View: The CADed part has a cylindrical cutout at the left side, but it's not completely cutting through. The original engineering drawing indicated that this cylinder should be a through hole.
2. Front View: The rectangular cutout looks fine in relation to the original drawing. However, we can't confirm the depth of this cutout from this view alone.
3. Right View: The circular cutout seems to be centered vertically, which seems accurate as per the original drawing.

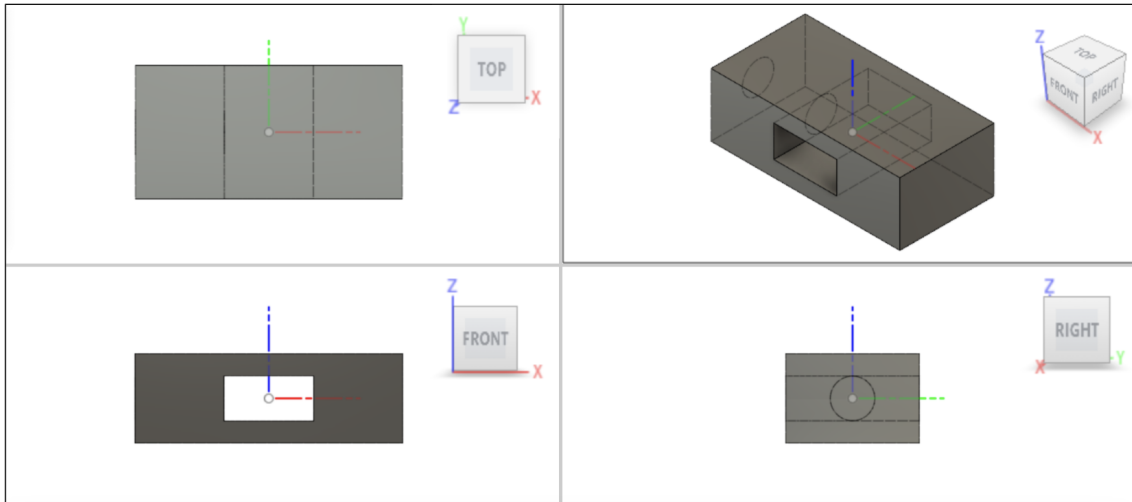
Given these observations, let's make corrections to the CadQuery script:

```
import cadquery as cq
# Parameters for the base box
length = 60
width = 30
height = 20
# Parameters for the cylindrical cutout
diameter = 10
depth = height # This should be a through hole
# Parameters for the rectangular cutout
cutout_length = 20
cutout_width = 10
cutout_depth = 10 # Assuming a depth based on the original drawing
# Create the base box
part = cq.Workplane("XY").box(length, width, height)
# Create the cylindrical cutout
part = part.faces("<X").workplane().hole(diameter, depth)
# Create the rectangular cutout
part = part.faces(">Y").workplane(centerOption="CenterOfBoundingBox").rect(cutout_length, cutout_width).cutThruAll()
show_object(part)
```

This updated CadQuery script corrects the cylindrical cutout to be a through hole. However, without having the exact measurements and depth of the rectangular cutout from the original drawing, I've made an assumption for its depth. You might need to adjust the depth value for the rectangular cutout ("cutout_depth") if it's different from the original drawing. Let me know if further adjustments are needed!

Cont.—Experiment 1—Engineering Drawing Analysis and CAD Generation Experiment*Prompt 6: (P6 - CAD Generation 4)*

That's not quite right – the CAD of the part created using the OpenSCAD script you just gave me does not match the part shown in the original engineering drawing I provided you. The attached photo shows four views of the part's CAD that was created using the CadQuery script you just provided. Each view shows a different orientation of the CADed part, as seen by the coordinate system shown in each view. The views show the CADed part in a shaded-with-hidden-edges style. Based on these CAD views, can you update the CadQuery script to fix any issues that you see, so that the part created using the script matches the part shown in the original drawing I provided?

**GPT-4V Answer:**

Absolutely, let's address this step-by-step. From the views provided, I can make the following observations and the corresponding adjustments to the CadQuery script:

1. Holes on Top Surface: The original engineering drawing shows two holes on the top surface, whereas the current CadQuery script only creates one. The placement of the hole created by the script also seems off from the provided views.
2. Cutout in Front View: The cutout shown in the "FRONT" view doesn't match the original engineering drawing. The cutout seems wider and doesn't seem to extend the full depth of the object as seen in the 3D view.

Let's make the necessary adjustments to the CadQuery script based on these observations:

```
import cadquery as cq

# Dimensions
L = 60 # Length
W = 40 # Width
H = 20 # Height
HOLE_D = 10 # Diameter of the holes
HOLE_P = 20 # Distance from edge to center of the holes
CUTOUT_D = 40 # Width of cutout
CUTOUT_H = 15 # Depth of cutout

# Main block
result = cq.Workplane("XY").box(L, W, H)

# Holes on top
result = result.faces(">Z").workplane().pushPoints([(HOLE_P, W/2), (L-HOLE_P, W/2)]).hole(HOLE_D)

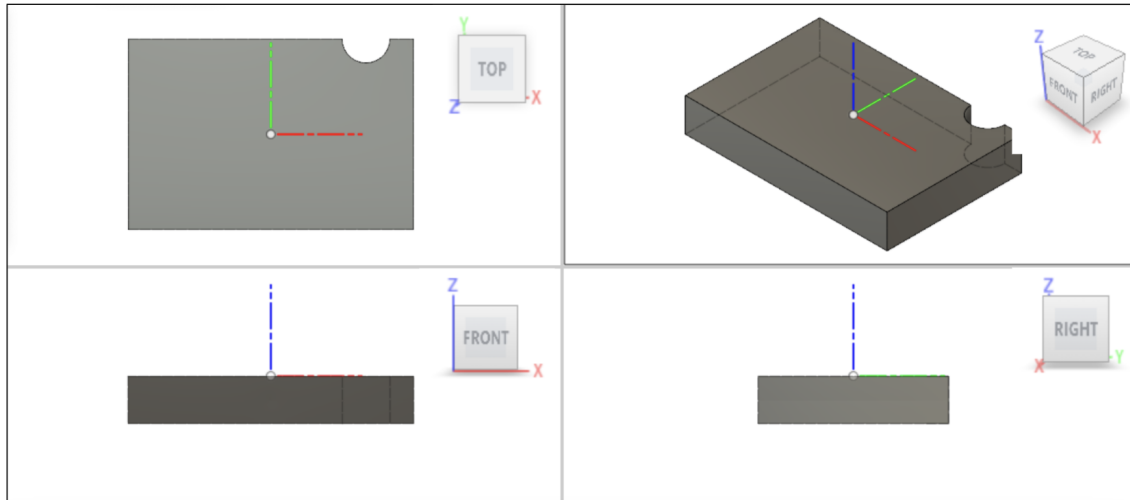
# Cutout on front face
result = result.faces(">Y").workplane().center(0, -H/2 + CUTOUT_H/2).rect(L, CUTOUT_H).cutThruAll()
```

This script should now be closer to the original engineering drawing. However, you may still need to make additional adjustments or provide further specifications to ensure a perfect match.

Cont.—Experiment 1—Engineering Drawing Analysis and CAD Generation Experiment

Prompt 7: (P7 - CAD Generation 5)

That's not quite right – the CAD of the part created using the OpenSCAD script you just gave me does not match the part shown in the original engineering drawing I provided you. The attached photo shows four views of the part's CAD that was created using the CadQuery script you just provided. Each view shows a different orientation of the CADed part, as seen by the coordinate system shown in each view. The views show the CADed part in a shaded-with-hidden-edges style. Based on these CAD views, can you update the CadQuery script to fix any issues that you see, so that the part created using the script matches the part shown in the original drawing I provided?



GPT-4V Answer:

Certainly. I'll compare the provided CAD views with the initial engineering drawing to identify discrepancies. Looking at the CAD views:

1. The most noticeable discrepancy is the cutout or recess on one edge. In the original drawing, there was a cutout on one side, but it appears the CadQuery script generated a part with a cutout that has different dimensions and position.
2. The rest of the main rectangular body seems to be alright, with no evident discrepancies.

To rectify this, let's adjust the CadQuery script, particularly the cutout section. However, without exact dimensions from the original drawing, I'll provide a generalized script adjustment. You might need to fine-tune it using exact measurements:

```
import cadquery as cq
```

```
# Base dimensions
```

```
length = 100
```

```
width = 50
```

```
height = 10
```

```
# Cutout dimensions
```

```
cutout_length = 30 # Adjust based on the original drawing
```

```
cutout_width = width # Assuming the cutout spans the entire width of the part
```

```
cutout_depth = 5 # Adjust based on the original drawing
```

```
cutout_position_from_left = 20 # Adjust based on the original drawing
```

```
# Create main block
```

```
result = cq.Workplane("XY").box(length, width, height)
```

```
# Subtract cutout
```

```
result = result.faces(">Z").workplane(). moveTo(cutout_position_from_left + (cutout_length / 2), 0). rect(cutout_length, cutout_width). cutBlind(-cutout_depth)
```

```
show_object(result)
```

Please adjust the "cutout_length," "cutout_depth," and "cutout_position_from_left" variables to match the exact dimensions from your original drawing.

Table 7: Summarized results from section 3.2.

Experiment Name	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Exp 6	Exp 7	Exp 8	Exp 9	Average Performance
(P1) Part Description	0/1	0/1	0/1	1/1	0/1	0/1	0/1	0/1	0/1	11%
8.0 Dimension	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	1/2	94 %
5.0 Dimension	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	100%
12.0 Dimension	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	100%
∅5.0 Dimension	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	100%
4.0 Dimension	1/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	1/2	89%
Additional Dimension	0	-1	0	0	0	0	0	0	0	
(P2) Dimension Extraction	9/10	9/10	10/10	10/10	10/10	10/10	10/10	10/10	8/10	96%

The entries which have a score of 1/2 always correspond with successful dimension extraction but incorrect label assignment.

Discussion: Engineering Drawing Analysis and CAD Generation Based on the results from Section 3.2.1 (see Table 7), which quantify GPT-4V’s ability to analyze an engineering drawing, we conclude that the model generally understands the content in the drawing, but struggles with interpreting the drawing’s details. For P1 in eight out of the nine experiments, GPT-4V incorrectly describes the part as a block with a hole “through” it. While it understands the part generally - a block with a hole - it does not pick up on the notation in the drawing that indicates the hole is blind rather than through. In the one experiment (experiment 4) where it received a correct score for the part description, it called the part a “rectangular block with a cylindrical hole or recess in it” and a “generic block with a hole.” While this qualifies as an accurate description, it does not demonstrate whether GPT-4V recognizes the blind hole in the drawing or not.

After being told that the part in question is a block with a blind hole (P2), GPT-4V is generally good at extracting dimensions from the drawing, receiving a perfect P2 score for six out of nine experiments (see Table 7). Across all nine experiments, GPT-4V always extracts all five dimensions from the drawing. Two-thirds of the time it assigns all of the dimensions appropriate labels, resulting in a high average performance score for P2 of 96%. It has the most trouble naming the hole depth dimension: in experiment 1 it calls it the “height of the block (from the bottom right view)” and in experiment 9 it calls it the “width of the block.” The relative difficulty in understanding what the 4.0 dimension represents in the drawing is consistent with its initial lack of understanding (in P1) that the drawing represents a block with a *blind* hole. It is also interesting to note that GPT-4V is inconsistent in the labels it chooses for the three-block dimensions – varying between height/width/length, depth/height/width, height/depth/width, and depth/height/length – perhaps reflecting a lack of consistent spatial reasoning or a lack of consistent norms used to label dimensions.

From the CAD Generation evaluation, Section 3.2.2, we observe (see Figure 5) that GPT-4V rarely generates accurate CAD on the first attempt (P3), and CAD iterations (P4-P7) do not improve the CAD. For P3, only one out of the nine experiments (experiment 3, using CadQuery) leads to correctly generated CAD on the first attempt. For FeatureScript, GPT-4V cannot get out of syntax and function implementation errors for all five CAD generation iterations. The most common issue for P3 is not putting the hole on the correct face. We noticed this is because the hole extrusion direction is always linked with the dimension to which GPT-4V assigns the “height” label. The 5.0 block dimension is only assigned the height label three times, one of which is the sole experiment (experiment 3) where a perfect score CAD is generated in P3.

From the results from P4-P7, we conclude that visual feedback of the generated design from the previous prompt does not improve GPT-4V’s CAD scripting ability. In fact, if GPT-4V generated incorrect CAD in P3, P4-P7 will never fully rectify the problematic CAD, and CAD Generation 5 (P7) will have a worse score than CAD Generation 1 (P3). A visualization of this finding can be seen in Figure 5. For the CadQuery and OpenSCAD experiments, a general reduction in CAD score occurs in CAD Generation 3 (P5), where GPT-4V consistently forgets the dimensions that it extracted in the original engineering drawing.

In summary, we find that GPT-4V can pick up many aspects of the provided engineering drawing (e.g. general part depicted, many of the dimensions shown, etc.), but can struggle when it comes to understanding the details (e.g. recognizing the *through* hole, labeling the through hole dimension, etc.). GPT-4V performs poorly when it comes to CAD generation, and we demonstrate that our attempts at visual, iterative improvements are unsuccessful. These findings imply that while GPT-4V can offer some assistance in preliminary design tasks, its current capabilities are not yet sufficient for detailed, precision-driven CAD work. Building on the findings from the evaluation of GPT-4V, future research should focus on enhancing the model’s ability to interpret and process detailed engineering information. Another critical area for development is in CAD generation, where GPT-4V currently shows limitations. Future work should explore methods to improve the model’s accuracy and efficiency in creating detailed CAD models, perhaps through advanced training techniques or integration with specialized CAD software. Additionally, there’s a need to

investigate how iterative feedback mechanisms can be better utilized by GPT-4V to make meaningful corrections and improvements in successive design iterations. Addressing these areas will be crucial in expanding the applicability of GPT-4V and similar VLM tools in more advanced and precision-dependent stages of the engineering design process.

In tandem with CAD generation and engineering drawing creation, engineers frequently aim to improve the design of the part, using iterative optimization approaches (e.g., using commercial tools such as nTop or SOLIDWORKS Simulation software). One of the commonly used iterative optimization approaches is structural topology optimization, which can help a designer reduce material usage while meeting some design requirements. In the next section, we turn to GPT-4V’s ability to assist with topology optimization.

3.3 Topology Optimization

Topology optimization (TO) is a mathematical approach used in engineering to find the best possible geometric layout of a material within a given design space, under a set of loads and constraints. It’s particularly significant in fields like structural mechanics and heat conduction [Sigmund and Maute, 2013]. The primary objective of TO is to optimize the distribution of the material in a design to achieve maximum performance, whether it be strength, stiffness, heat dissipation, or other objectives while adhering to both rigid and flexible constraints.

In the engineering design process, topology optimization plays a crucial role. It allows engineers to explore a broader range of design possibilities and innovate more efficient, effective, and lightweight structures. This is especially valuable in scenarios where material savings or performance improvement are critical. However, TO is inherently complex, often demanding significant computational resources. Engineers typically rely on various approximation methods to simplify and manage the complex calculations this optimization involves, as detailed in extensive reviews of the subject. This process not only enhances design efficiency but also pushes the boundaries of what is possible in engineering design.

Overview and Motivation. Structural topology optimization is a numerical method used to find the best distribution of material for a given structure under specified constraints. It aims to ensure the material is used efficiently while meeting performance targets. A prevalent method in this field is the SIMP (Solid Isotropic Material with Penalization) approach, where the material properties are modeled using a density field that represents the material quantity in different regions of the structure [Bendsøe, 1989]. This method iteratively adjusts the density to optimize the design while adhering to constraints related to stress or deformation. For the *minimum compliance problem* in mechanical systems, the task is to determine the material density distribution, $\mathbf{x} \in \mathbb{R}^n$, that minimizes the structure’s deformation under applied forces and boundary conditions [Bendsøe and Kikuchi, 1988]. The formulation of the problem is as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & c(\mathbf{x}) = F^T U(\mathbf{x}) \\ \text{s.t.} \quad & v(\mathbf{x}) = v^T \mathbf{x} < \bar{v} \\ & 0 \leq \mathbf{x} \leq 1 \end{aligned} \tag{1}$$

In this equation, the objective is to minimize the compliance $c(\mathbf{x})$, with F being the external load forces, $U(\mathbf{x})$ the displacements of nodes, solution of the equilibrium equation $K(\mathbf{x})U(\mathbf{x}) = F$, and $K(\mathbf{x})$ the stiffness matrix which depends on the material distribution. The constraints include maintaining the volume fraction $v(\mathbf{x})$ below a specified limit $\bar{v}$ and ensuring the design variables $\mathbf{x}$ remain within the bounds of 0 and 1, allowing for a gradient of material distribution from void to solid [Bendsøe and Kikuchi, 1988, Sigmund and Maute, 2013]. Topology optimization offers the advantage of creating designs that are optimally configured to meet specific functional requirements. However, the process can be highly computationally demanding [Liu and Tovar, 2014, Sigmund, 2001], requiring substantial resources. It may not always be suitable for generating intricate geometries or could become trapped in local optima, leading to less-than-optimal solutions. For such reasons, recently vision-based generative models have been employed to speed up direct design [Nie et al., 2021, Mazé and Ahmed, 2023, Giannone et al., 2023]. For a review of machine learning and topology optimization, see [Woldseth et al., 2022].

However, optimal topologies are often challenging to analyze for human experts. Often, the configurations that result from the optimization process may be mathematically optimal but can pose significant challenges in terms of manufacturability and analysis. These designs, while efficient in simulations, may include complexities such as overhangs or internal cavities that current manufacturing processes cannot produce. Furthermore, the often abstract and non-intuitive shapes generated by such algorithms can be difficult for human designers to understand and translate into practical designs. This gap between optimal structural design and practical application raises the need for advanced tools to bridge the divide. Developing visual-language models that can interpret and suggest modifications to complex topologies could prove essential. Such a tool would not only need to understand the mathematical underpinnings of topology optimization but also incorporate knowledge of manufacturing constraints and design principles that are palatable to human engineers. By doing so, it could provide alternative solutions that, while potentially sacrificing some degree of optimality, would offer greater feasibility in the real world and improved interpretability for human collaborators.

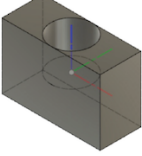
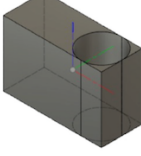
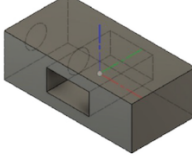
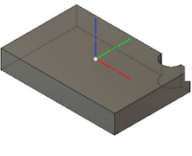
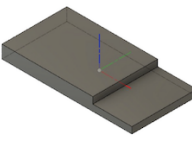
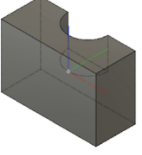
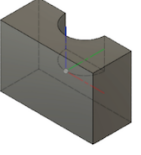
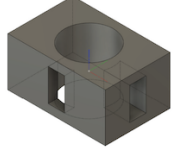
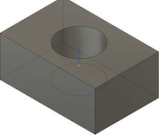
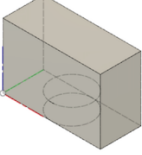
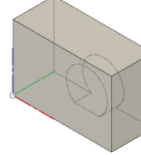
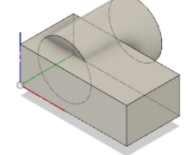
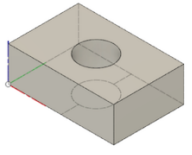
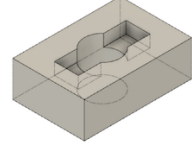
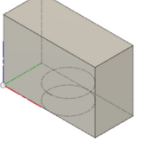
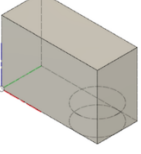
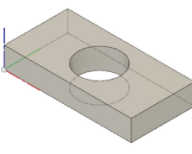
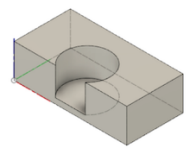
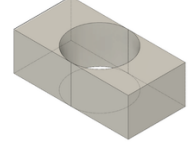
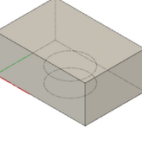
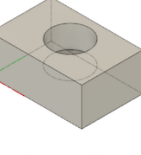
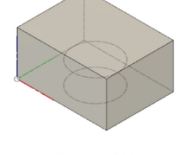
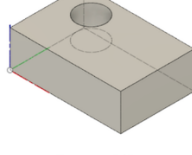
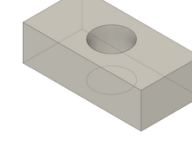
	CAD Generation 1	CAD Generation 2	CAD Generation 3	CAD Generation 4	CAD Generation 5
Experiment 1	 Score: 5/6	 Score: 3/6	 Score: 1/6	 Score: 2/6	 Score: 0/6
Experiment 2	 Score: 4/6	 Score: 4/6	Code Error Score: 0/6	Code Error Score: 0/6	 Score: 1/6
Experiment 3	 Score: 6/6				
Experiment 7	 Score: 4/6	 Score: 4/6	 Score: 0/6	 Score: 3/6	 Score: 2/6
Experiment 8	 Score: 4/6	 Score: 3/6	 Score: 3/6	 Score: 2/6	 Score: 3/6
Experiment 9	 Score: 5/6	 Score: 4/6	 Score: 3/6	 Score: 2/6	 Score: 3/6

Figure 5: Results of the CAD Generation prompts, CAD Generation 1 - CAD Generation 5 (P3 - P7). Experiments 1 - 3 were generated using CadQuery and Experiments 7 - 9 were generated using OpenSCAD. Experiment 3 was the only experiment that generated a perfect score CAD on the first iteration. Experiments 4 - 6, the FeatureScript experiments, are not shown here, since they had persistent code errors and never generated viable CAD.

Evaluation. Our evaluation aims to assess GPT-4V’s capabilities in several key areas:

- (i) **Testing General Comprehension (3.3.1).** How well does GPT-4V comprehend and provide useful information across various topics and domains, particularly in topology optimization?
 - We assess that the model is knowledgeable and useful for general inquiries about TO.
- (ii) **Comprehending Complex Designs under Input Constraints (3.3.2).** Can GPT-4V effectively understand and analyze complex designs while adhering to specified input constraints?
 - The model can analyze constrained designs, partially identifying feasible configurations and improving with the help of specialization in the prompt.
- (iii) **Inferencing Constraints for a Given Design (3.3.3).** Is GPT-4V capable of accurately inferring constraints, such as Volume Fraction, from a given design?
 - The model fails to correctly estimate the VF without the help of a code interpreter.
- (iv) **Technical Knowledge (3.3.4).** How deep is GPT-4V’s technical knowledge, particularly in providing accurate responses about complex topology images and technical diagrams?
 - The model is knowledgeable about technical details and can caption complex images and technical diagrams.
- (v) **Assess Design Validity (3.3.5).** To what extent can GPT-4V evaluate the feasibility, validity, and manufacturability of a given design, especially in identifying and positioning Floating Material?
 - The model is useful for qualitative evaluation but fails to be consistent when asked about the validity of a given design in the presence of Floating Material (FM). The model fails in finding and positioning the FM. In general, spatial reasoning is a weak point of the GPT-4V model. The model returns output almost as random over multiple experiments and scenarios.
- (vi) **3D Spatial Understanding (3.3.6).** How proficient is GPT-4V in comprehending and assisting with three-dimensional spatial concepts and designs in topology optimization?
 - The model is knowledgeable about general 3D TO and is a useful aid to speed up analysis.

We highlight the strengths and weaknesses of GPT-4V’s response by marking the more relevant and high-quality sections in green, and the incorrect, out-of-context, or low-quality parts in brown.

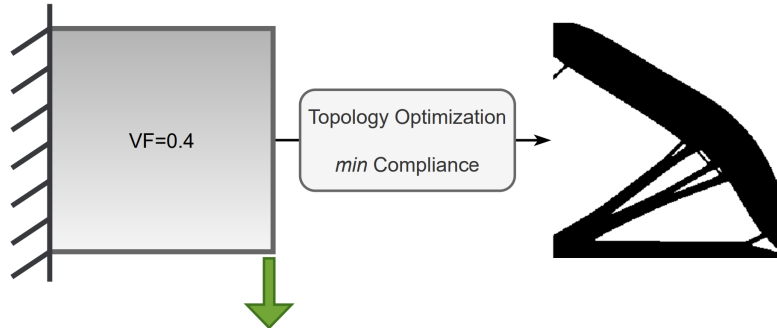
3.3.1 Basic Understanding

General Knowledge. We assess the model’s proficiency in applying constrained design principles, focusing particularly on Topology Optimization. This evaluation involves interpreting a diagram that illustrates a typical scenario, showcasing boundary conditions, loading, volume fraction, and the initial domain on the left side, juxtaposed with the final optimized topology on the right. Additionally, we present a visualization of the optimization process, which aims to minimize compliance. This process seeks to create structures of maximum stiffness while adhering to the specified constraints. To assess the response, we highlight the parts of its responses that are relevant in green and irrelevant parts in yellow.

Context 3.3.1: General Knowledge

Prompt 1: (General Knowledge)

Describe the image in general and technical terms. Where are the constraints applied? Where are the loads?



GPT-4V Answer:

1. General Description:

- The image presents a flow diagram that outlines a process used for designing or optimizing a structure. The process starts with specific inputs, goes through a stage of optimization, and results in a particular design shape.

2. Technical Description:

(a) Inputs

- Loads:** The external forces or pressures that the design must support or withstand.
- Volume Fraction:** This likely refers to the ratio of solid material to the total design space. It's an important parameter in topology optimization to determine the distribution of material in the design.
- Boundary Conditions:** These dictate how the design is constrained or fixed in place. They can significantly impact how a design reacts to loads.

(b) Constrained Optimization

- This is a computational method used to find the best design or structure that meets specific performance goals, while adhering to constraints like material usage or design space. The optimization seeks to "maxPerformance," implying the objective is to maximize some measure of performance (e.g., strength, efficiency).

(c) Resultant Design

- On the right, there's a black silhouette of a structure, possibly the optimized output from the constrained optimization process. It has a curved top supported by several diagonal braces or members, indicating areas of material distribution to handle the applied loads most efficiently.

3. Constraints and Loads:

- Constraints:** The areas where the design is thickest, especially at the base, suggest where boundary conditions or constraints might be applied. For this design, it seems the bottom left corner might be a fixed or constrained point.
- Loads:** While the diagram does not explicitly show where the loads are applied on the resulting design, the shape suggests it might be designed to bear loads from the top or the curved part, possibly pushing downwards, given the bracing structure underneath.

Constraints Identification. In this task, we present GPT-4V with a textual prompt and a 2D optimal design, tasking it to suggest potential constraint configurations compatible with the given design. Our emphasis lies specifically in the realm of mechanical design, with a focus on identifying loads and boundary conditions. It's important to note that this problem is inherently ambiguous, allowing for many valid responses. To then specify the problem, we introduce a specific loading configuration. This step narrows down the range of potential constraint configurations, providing more targeted guidance for the solution. For the computation of optimized topologies, we employ Topy [Hunter et al., 2017] and the Solid Isotropic Material with Penalization (SIMP) algorithm [Bendsøe and Kikuchi, 1988].

Context 3.3.2: Constraint Identification

Prompt 1: (Unconditional Constraints Identification)

For the attached optimal design, where could the load and boundary condition be located?



GPT-4V Answer:

Based on the image provided, it appears to be a silhouette or contour of an abstract structure or design. Without more specific context or details about the intended purpose or nature of the design, it's challenging to definitively specify the load and boundary conditions. However, I can provide a general interpretation:

1. Boundary Conditions:

- The regions where the structure appears to be fixed or anchored, typically the thicker or wider areas, might serve as points of support or constraints. These could be places where the design is intended to be attached to other structures or surfaces.
- In this image, the bottom left and bottom right ends of the structure appear to be potential areas where boundary conditions could be applied, suggesting they might be fixed or clamped in place.

2. Load Application:

- Loads might be applied in areas where you expect the structure to experience external forces. This could be due to weight, pressure, or other external factors.
- Considering the silhouette, potential load application areas might be the upper and more extended parts of the design, as they would likely experience more force due to gravity or other factors if acting as beams or lever arms.
- Additionally, any thinner sections or more delicate features of the design may be points of interest for load applications, as these areas might be more vulnerable to stresses.

Remember, this interpretation is a general overview based on the shape and form of the design in the image. Specific load and boundary conditions would depend on the exact intended use, material properties, environmental conditions, and other factors related to the design's actual application.

Cont.—Constraint Identification

Prompt 2: (Conditional Constraint Identification)

For this design (refer to the previous image), there is only a vertical load applied on the bottom right of the design. Can you better identify the constraints now?

GPT-4V Answer:**1. Boundary Conditions:**

- For the design to remain stable and not simply rotate or move due to the vertical load, there must be at least one fixed or clamped boundary condition to counteract this load.
- The bottom left end of the structure, being opposite to where the load is applied and the thickest part of the design, is a logical point for a fixed or clamped boundary condition. This would allow the design to counteract the rotational effects of the applied load.
- Additionally, the area near the top left corner, given its broader profile, may also serve as a constraint to help distribute and resist the forces resulting from the applied load. This would prevent excessive rotation and ensure stability.

2. Load Application:

- As mentioned, the load is applied at the bottom right of the design. Given its location, it would introduce bending moments and shear forces along the structure.
- The design would need to be strong and stiff enough, especially in the middle portions, to accommodate these stresses.

With the provided load information, the primary constraints would likely be located at the bottom left and potentially near the top left corner of the design to ensure stability and counteract the effects of the applied vertical load on the bottom right.

Volume Fraction Estimation. In this experiment, we task the model with calculating the volume fraction from an optimized topology depicted in an image. This involves measuring the proportion of black material in the given domain and determining the relevant ratio. The challenge is initially approached using GPT-4V’s visual analysis capabilities alone. Following this, GPT-4V employs its code interpretation abilities to address the task. We aim to obtain an accurate answer within a 5% error threshold.

Context 3.3.3: Volume Fraction Estimation

Prompt 1: (Quantitative analysis without tools)

Can you estimate the volume fraction of this structure? Consider that white means the absence of material and the initial domain in a square of size 256x256.



Target: 0.40

GPT-4V Answer:

1. To determine the volume fraction, we can evaluate the percentage of the black area (material presence) relative to the total area of the square.
2. The formula for volume fraction ϕ is:

$$\phi = \frac{\text{area of the black region}}{\text{total area}}$$

3. From the image, the total domain is 256×256 , so the total area A_{total} is 65536 pixels.
4. Let's compute the area of the black region A_{black} .
5. After analyzing the image:

$$A_{\text{black}}$$

is approximately 42540 pixels (this is an estimation based on the provided image).

6. Plugging into the formula:

$$\begin{aligned}\phi &= \frac{42540}{65536} \\ \phi &\approx 0.649\end{aligned}$$

7. Therefore, the estimated volume fraction of the structure is approximately 0.649 or 64.9%.

Cont.—Volume Fraction Estimation

Prompt 2: (Quantitative analysis without tools - II attempt)

Can you try again? Consider that white means the absence of material and the initial domain in the square of size 256x256.

Target: 0.40

GPT-4V Answer:

1. Certainly! I apologize for the confusion.
2. To estimate the volume fraction of the black structure in the image, I will calculate the ratio of black pixels to the total pixels in the square.
3. The method involves counting the number of black pixels in the image and dividing by the total number of pixels.
4. Given that the total number of pixels in the image is $256 \times 256 = 65536$ pixels, let's compute the number of black pixels.
5. After analyzing the image:

The black pixel count is approximately 33333 pixels

(this is an estimation based on the provided image).

6. Using the formula for volume fraction ϕ :

$$\phi = \frac{\text{area of the black region}}{\text{total area}}$$

7. Plugging in the values:

$$\phi = \frac{33333}{65536} \approx 0.508$$

8. Therefore, the estimated volume fraction of the structure is approximately 0.508 or 50.8%.

Cont.—Volume Fraction Estimation

Prompt 3: (Quantitative Analysis with tools - code interpreter)

Can you compute the volume fraction of this structure (refer to previous figure in subsection)? White means absence of material and the square domain is 256x256 pixels.

Input File: topology0.png

Target: 0.40

GPT-4V Answer:

Sure! I'll start by loading the image and visualizing it. Then, I'll compute the volume fraction based on the number of white pixels (representing the absence of material) and the total number of pixels in the domain.

```
import matplotlib.pyplot as plt
import numpy as np
from PIL import Image

# Load the image
img = Image.open("/mnt/data/topology0.png").convert('L') # Convert to grayscale

# Display the image
# plt.figure(figsize=(8,8))
# plt.imshow(img, cmap='gray')
# plt.axis('off')
# plt.show()
```

Here's the loaded image. Now, I'll compute the volume fraction. The volume fraction will be determined by the ratio of non-white pixels (material presence) to the total number of pixels in the domain.

```
# Convert image to numpy array and normalize to range [0, 1]
img_array = np.array(img) / 255.0

# Compute volume fraction
\hl{# Since white is 1 and black is 0, the mean gives the fraction of white
volume_fraction = 1 - np.mean(img_array)
}

volume_fraction
```

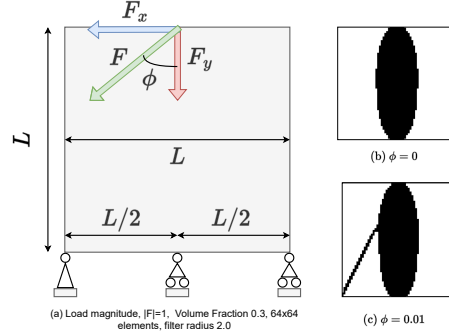
Result: 0.3852859438817139

Technical Captioning. In this task, we use the model to caption a technical diagram using a basic prompt, with the diagram inspired by the experiment in Figure 7 from Woldseth et al. [2022]. Initially, we employ a generic prompt for captioning. Then, we enhance the task by incorporating details about the system’s technical expertise, providing a more in-depth and knowledgeable description of the diagram. We would like to see that the model understands that small variations of constraint configurations (in this case load direction) can greatly change the optimized topology.

Context 3.3.4: Technical Captioning

Prompt 1: (Generic Prompt)

Can you describe this image?



Target: Significant effect small changes in boundary conditions and loading can have on the optimal topology. The boundary conditions (a) with corresponding optimized designs for (b) and (c).

GPT-4V Answer:

The image consists of three separate figures labeled (a), (b), and (c).

- (a) This figure depicts a rectangular frame with a length labeled as "L". The rectangle is divided into two halves by a center line, and each half is labeled " $\frac{L}{2}$ ". At the bottom corners of the rectangle, there are two hanging weights. On the top of the rectangle, there's a force vector labeled "F" making an angle " ϕ " concerning the horizontal axis. This force "F" is decomposed into two components: a horizontal force " F_x " and a vertical force " F_y ". The caption underneath mentions that the load magnitude $|F|$ is 1, the volume fraction is 0.3, there are 64x64 elements, and the filter radius is 2.0.
- (b) This figure shows a square with an oval-shaped black region in the center. The caption below the square mentions " $\phi = 0$ ".
- (c) Similar to (b), this figure shows a square. However, the black oval-shaped region in the center is slightly tilted to the right and has a diagonal black stripe running from the bottom left to the top right. The caption below the square indicates " $\phi = 0.01$ ".

It appears that the images might be related to some form of structural or mechanical analysis, possibly demonstrating how the structure or material responds to different load angles.

Cont.—Technical Captioning

Prompt 2: (Specialized prompt)

Answer the same question (refer to the previous image) using this instruction: You are an expert engineer with years of experience in topology optimization and engineering design.

Target: Significant effect small changes in boundary conditions and loading can have on the optimal topology. The boundary conditions (a) with corresponding optimized designs for (b) and (c).

GPT-4V Answer:

The image represents a topology optimization problem, a technique frequently used in structural and mechanical design to determine the best material distribution under specific loading and boundary conditions.

1. (a) This figure illustrates a common boundary condition setup for a topology optimization problem. The depicted domain, a rectangle of size "L", represents a structure or component. Two external point loads, represented by weights, are applied at the bottom corners. On the top, there's a force vector "F" applied at an angle " ϕ " to the horizontal. This force is decomposed into its horizontal " F_x " and vertical " F_y " components. It's essential to note the given parameters: the magnitude of the applied load is $|F| = 1$, and the design space has been discretized into a 64x64 grid. The volume fraction of 0.3 suggests that only 30% of the total domain will be occupied by material, while the rest will be void. The mention of a filter radius of 2.0 indicates the usage of a filtering technique, which is often employed in topology optimization to ensure mesh-independent results and to smoothen the material distribution.
2. (b) This figure depicts the optimized material layout for a direct vertical load, as indicated by " $\phi = 0$ ". The black region represents the material presence, suggesting a symmetric structure optimized to handle vertical loads efficiently. The shape's orientation and form are dictated by the combination of the loading, boundary conditions, and design domain, resulting in the most efficient load-bearing structure with minimal material usage.
3. (c) Here, the optimized topology is presented for a slightly angled load, as denoted by " $\phi = 0.01$ ". The slight tilt in the material region and the additional diagonal brace-like feature suggest an optimized design that caters to both vertical and small horizontal components of the applied force. This shows the adaptability of the topology optimization technique in producing efficient designs for varied loading conditions.

In summary, these images provide a clear depiction of how topology optimization evolves structural designs according to specified load cases, ensuring efficient material usage and optimal load-bearing capability.

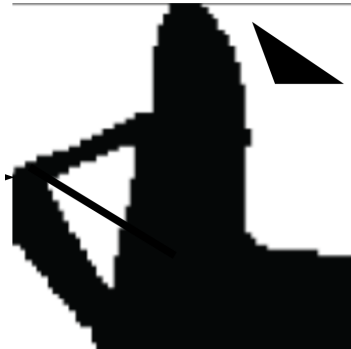
3.3.2 Invalid Design

The task involves identifying invalid designs, specifically floating materials, based on a given prompt. The objective is for the model to recognize, independently and without prior information, the presence of a disconnected component within a low-resolution design (64x64). Following this recognition, the model is expected to assess the design's overall validity and the quality of the low-resolution grid. Lastly, we test if the model is capable of suggesting potential improvements to rectify the identified issues.

Context 3.3.5: Invalid Design Identification

Prompt 1: (Understanding)

Can you describe this image?



GPT-4V Answer:

The image showcases a topology, possibly a result of a topology optimization process. Here's a description of the depicted design:

The topology is primarily characterized by a black structure against a white background. The structure appears to have:

1. **Central Tower:** There's a dominant central structure that resembles a tower. It tapers off at the top, looking similar to a spire or peak. This component might be designed to bear the bulk of the load or stress.
2. **Diagonal Bracing:** There's a notable diagonal brace or support that starts from the bottom-left corner and connects to the central tower structure at around its mid-point. The design of this brace is jagged or stair-like, indicating possible pixelation or a design response to the discretization used in the optimization process. This brace would provide additional support against lateral loads or stresses.
3. **Horizontal Base:** The central tower is rooted in a broader horizontal base structure. This base stretches towards the right bottom corner of the image, providing foundational support.
4. **Isolated Triangle:** Towards the top right corner of the image, there's an isolated triangular shape. Its disconnection from the main structure is intriguing. This shape could either be a byproduct of the optimization process, represent a region of minor stress concentration, or perhaps an artifact of the resolution used.

The topology appears to be a design solution for a specific set of loading and boundary conditions. The main structural features—namely, the tower, diagonal brace, and base—would be the primary load-bearing components, ensuring efficient material usage and structural integrity for the given constraints.

Cont.—Invalid Design Identification

Prompt 2: (Evaluation)

Do you think this (refer to previous design) is a "valid" topology?

GPT-4V Answer:

Evaluating the provided topology:

The design consists of a primary central structure with distinct features, such as a diagonal brace and an isolated triangular shape. Here's an assessment:

1. **Pixelation Sharp Corners:** The design seems pixelated, especially evident in the diagonal brace, which appears jagged or stair-like. Sharp corners or sudden changes in geometry, as suggested by this pixelation, could introduce stress concentrations in a real-world structure, potentially leading to failure.
2. **Isolated Triangle:** The triangular shape towards the top right is disconnected from the main structure. In practical manufacturing scenarios, this isolated feature might be problematic to manufacture, and its disconnected nature might render it ineffective in a load-bearing role.
3. **Material Usage:** The design seems to be focused on minimal material usage, which is a typical goal of topology optimization. However, this also means that any structural weaknesses are amplified since there's less material to absorb or distribute unexpected stresses.

In conclusion, while the design showcases interesting structural features that may be optimized for specific load cases, its pixelation and the presence of the isolated triangle raise questions about its practical validity and real-world performance. Before pursuing manufacturing, it would be advisable to refine the design to smooth out pixelated regions, consider the connectivity of all features, and validate its performance using finite element analysis or similar computational tools.

Cont.—Invalid Design Identification

Prompt 3: (Optimization)

How would you improve such a design (refer to previous design)?

GPT-4V Answer:

Improving the design resulting from a topology optimization process often requires a combination of computational refinement and engineering judgment. Here's how the design could be improved:

1. Smoothen Pixelation:

- Refine the optimization process to reduce the pixelated or jagged regions. This might involve increasing the mesh density or employing a post-optimization smoothing algorithm.
- Utilizing a filtering technique within the optimization process can also help in achieving a smoother result.

2. Address Disconnected Components:

- The isolated triangular shape should be either integrated into the main structure or removed if it doesn't provide any significant functional benefit.
- A re-evaluation of boundary conditions and load cases can ensure that disconnected components are not formed.

3. Review & Refine Structural Features:

- The diagonal brace, though an important structural feature, appears pixelated. Its design should be smoothed and refined to ensure it provides the required support without introducing stress concentrations.
- Further, for manufacturing feasibility, any internal voids or intricate geometries might need to be simplified.

4. Material & Manufacturing Considerations:

- Choose a suitable material that aligns with the expected loads and the design's intricacies. Some materials are more forgiving of sharp transitions or thin sections.
- If considering additive manufacturing, ensure the design accounts for support structures and potential post-processing steps.
- For traditional manufacturing, avoid intricate geometries that would require specialized tooling.

5. Iterative Validation:

- After making design modifications, it's crucial to validate the performance using computational tools like finite element analysis. This ensures that the improved design still meets performance criteria.
- Physical prototyping and testing can provide real-world validation of the design's performance and manufacturability.

In summary, improving such a design requires an iterative approach that combines computational tools, engineering judgment, and feedback from stakeholders. Each refinement step should be validated to ensure the design's integrity and performance are maintained or enhanced.

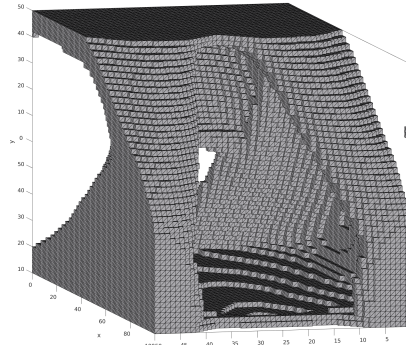
3.3.3 3D Structure

This task evaluates and analyzes 3D topologies. Our goal is to qualitatively assess GPT-4V's ability to interpret and understand three-dimensional spaces through topology optimization analysis.

Context 3.3.6: 3D Spatial Structure

Prompt 1: (Spatial Structure)

Can you describe this image?



GPT-4V Answer:

This image represents a meshed topology, likely resulting from a finite element analysis or a topology optimization process. Here's a detailed description:

1. Meshed Domain

- The image primarily displays a domain filled with triangular elements. This type of mesh is common in finite element methods (FEM), where the domain is divided into smaller elements to compute approximate solutions to boundary value problems.

2. Axes and Scale

There are two distinct axes:

- The horizontal axis is labeled as "x" and spans from approximately 0 to around 80.
- The vertical axis is labeled as "y" and goes from 0 to about 50.

3. Topology Features

- Main Structure:** There's a pronounced region in the center-right that's more densely meshed, indicating the existence of a significant structure or material.
- Cut-Outs and Voids:** Within this main structure, there are areas devoid of the mesh, which represent voids or cut-outs in the topology. These could be the results of the optimization to remove unnecessary material or regions that don't significantly contribute to the structural performance.
- Diagonal Features:** There are visible diagonal features within the dense meshed region, reminiscent of supports or braces. They likely provide structural rigidity and counteract specific load paths or stress trajectories.
- Boundaries and Edges:** The outer boundary of the topology is interesting with its irregular form. On the left side, it seems to have a concave curve, while on the right, it has a stair-step appearance, which might be indicative of the design constraints or requirements.
- Thick Upper Region:** The upper part of the topology appears to be thicker or denser, which might suggest a region designed to bear a significant portion of the applied loads.

Given the nature of the mesh and topology, this could be a solution for a structural topology optimization problem under certain loading and boundary conditions. The design aims to distribute material efficiently, ensuring optimal structural performance while minimizing material usage. The presence of the mesh indicates the potential for this design to be analyzed using FEM for further validation or performance prediction.

Discussion

Overview. The model exhibits a basic understanding of topology optimization (3.3.1), drawing from a vast database of related concepts and problems in engineering. This understanding extends to a vocabulary encompassing boundary conditions, loads, metrics, and how they interact (3.3.2). Furthermore, its capacity to process visual information allows for some informative qualitative analysis of results and diagrams (3.3.4).

Tasks.

- In Context 3.3.1, the model demonstrates knowledge of TO principles, linking volume fraction to *"the ratio of solid material to the total design space,"* understanding boundary conditions as *"how the design is constrained,"* and recognizing constrained optimization as *"finding the best design within material constraints."* It also identifies the right side of the diagram as *"a possible result of the optimization process."*

However, the model does not fully grasp the relationship between the diagram's three sections. While correctly noting the generic goal of *"maximizing performance,"* it overlooks the central diagram's focus on minimizing compliance or maximizing stiffness equivalently. Additionally, it fails to connect the left side's green arrow with the loading conditions, mistakenly suggesting the diagram *"does not explicitly show where the loads are applied"*, despite the clear indication of loads on the left and the depiction of the constraint setup, optimization routine, and optimized topology from left to right.

- In Context 3.3.2, the model navigates the ambiguity of asking about potential constraints behind a specific design. Acknowledging the uncertainty, it highlights, *"Without more specific context or details,"* the difficulty in *"definitively specifying the load and boundary conditions."* Despite this, the model thoughtfully proposes likely boundary conditions, such as *"the bottom left and bottom right ends of the structure,"* and potential load areas, like *"the upper and more extended parts of the design,"* based on the limited information available. It concludes by reiterating, *"Specific load and boundary conditions would depend on the exact intended use,"* underlining the need for more detail. This response could be considered effective as it not only provides a solution but also highlights missing information for an inverse problem. It's observed that this scenario, focusing on a single object, contrasts with the previous case involving a structure image, suggesting the vision encoder may more easily interpret global and local information in scenarios with a singular focus.

In the latter section of Context 3.3.2, additional details on the loading condition are introduced. With this information, the model infers that for the *"design to remain stable and not simply rotate or move due to the vertical load, at least one fixed or clamped boundary condition is necessary to counteract this load."* The assumptions made are correct. The model suggests that these constraints are likely *"located at the bottom left and possibly near the top left corner of the design"* for structural balance. Overall, the model effectively utilizes the new load information to deduce conclusions about rotations, momentum, and stability.

- In Context 3.3.3, we challenge the model to quantitatively estimate the volume fraction of an optimized topology. The model accurately defines the task as *"evaluating the percentage of the black area (material presence) relative to the total area of the square."* However, its initial attempts to count black pixels and calculate the material percentage yield highly inaccurate results. This inconsistency persists across multiple trials, each providing different and incorrect answers.

To address this, we introduce a code interpreter (third part of Context 3.3.3), enabling the model to use a Python script for the estimation. This approach significantly improves accuracy, bringing the estimate close to the target within a reasonable margin of error. This experiment highlights two key insights: Firstly, it underscores the limitations of the vision encoder (at least for the version of GPT-4V used in this study) in handling precise quantitative assessments based on images. Secondly, it demonstrates the effectiveness of integrating coding tools in overcoming these limitations, showcasing the synergistic potential of combining AI's interpretive capabilities with precise, code-based calculations for more accurate and reliable results.

- In Context 3.3.4, we evaluate the model's ability to interpret a complex diagram. The model delivers a comprehensive and accurate analysis, adeptly linking forces, their angles of application, and optimized topology. It accurately identifies the image as *"a structural or mechanical analysis, demonstrating how the structure or material responds to varying load angles."* This insight into how loading direction affects topology is a correct deduction of a difficult physical problem, showcasing the model's proficiency in understanding boundary conditions, loads, and their impact on structures.

However, the model encounters difficulty with the boundary conditions at the bottom center and right of the diagram, mistakenly interpreting them as *"two hanging weights,"* which is an incorrect assessment of the boundary sketch. This misinterpretation is unexpected, particularly given the overall high-quality response and the accurate grasp of the problem's essence.

Further refining the prompt to emphasize engineering concepts, the model again provides a largely accurate response, delving deeper into topics like loads, volume fraction, and filtering radius. Yet, it repeats the same error concerning

the boundary conditions, suggesting *"Two external point loads, represented by weights, are applied at the bottom corners."* This persistent mistake indicates a gap in the model’s global understanding of the scenario, revealing a vulnerability to misconceptions in specific contexts.

- In Context 3.3.5, we task the model with identifying the presence of floating material in a design, specifically a detached triangle in the top right corner. The model accurately recognizes the issue as a result of a topology optimization process, correctly noting the disconnection of the *"isolated triangular shape from the main structure"* in the top right.

When queried about the structure’s validity, the model identifies the floating material but its response lacks full clarity on the implications of such a flaw. While it correctly points out that *"this isolated feature might be problematic to manufacture, and its disconnected nature might render it ineffective in a load-bearing role,"* it fails to emphatically state that a disconnected component invariably compromises structural integrity and manufacturability.

Regarding the design’s optimization objectives, the model suggests a focus on *"minimal material usage."* This is a common requirement in topology optimization, but it oversimplifies the broader range of performance requirements typically involved in such processes.

When asked about improving the design, the model sensibly proposes *"integrating the isolated triangle into the main structure or removing it if it lacks functional benefit."* This is a valid solution to address the floating material issue. However, its subsequent recommendation to *"re-evaluate boundary conditions and load cases"* as a method to eliminate disconnected components is somewhat misguided. The more appropriate approach would involve refined optimization strategies and post-processing techniques.

In summary, while GPT-4V effectively identifies floating material and offers viable solutions, it falls short of fully understanding the criticality of disconnected components. Floating material or disconnected parts in a topology-optimized design invariably render it structurally unsound from an engineering point of view or unmanufacturable without further optimization or processing.

- In Context 3.3.6, we present the model with a 3D structure, seeking a basic description. The model’s response is largely appropriate, albeit somewhat general. It appears that minor inaccuracies in the description may stem from perspective challenges and the limited spatial comprehension demonstrated by the vision encoder in our evaluations. This suggests an area for improvement in the model’s ability to interpret and describe complex three-dimensional structures.

GPT-4V can be used to enhance the efficiency of the detailed design phase in engineering. In a scenario where an engineering team is tasked with sifting through a multitude of outputs from a large-scale topology optimization simulation, involving various boundary and loading variations, GPT-4V’s capabilities become particularly evident. The challenge for the team is to select optimized topologies that are devoid of residuals (like floating material) and exhibit high performance. Manually inspecting each topology would be prohibitively time-consuming and resource-intensive for a team of skilled engineers.

This is where GPT-4V steps in. The model can be employed to filter out topologies that do not adhere to the set constraints. By inputting the topology along with a representation of the constraints configuration (3.3.4, 3.3.2), GPT-4V can potentially identify non-compliant designs. Additionally, with its code interpreting abilities, GPT-4V can estimate material usage to ensure it does not surpass the prescribed volume fraction (3.3.3). Furthermore, the model is adept at detecting disconnected components in the designs (3.3.5), although its understanding of the impact of such issues on the design’s validity might be limited. All these tasks would be hard to define if not impossible using a standard text-based language model without the vision component, providing an indication that VLMs would be particularly useful for vision-based repetitive tasks in the detailed design phase.

Limitations. When diving into the nuances of the validity of an engineering design, GPT-4V sometimes struggles to provide clear insights (3.3.5). It can identify floating material within a design, but might not always connect this observation to design invalidity (3.3.5). Moreover, while it can discern and describe topologies on a qualitative level, quantitative reasoning about the topology remains a challenge (3.3.3). For instance, without the aid of a code interpreter, the model cannot estimate volume fractions (3.3.3) or accurately pinpoint the locations of disconnected components or floating materials (3.3.5).

Conclusion. While our analysis suggests that GPT-4V is a useful tool in the detailed design process in many respects, it has specific constraints that users should be aware of in the context of topology optimization. In particular, quantitative and spatial reasoning, as of now, are limited and not fully reliable, making it difficult for the model to use the model without expert human supervision. Overall, while GPT-4V shows the potential to become a co-pilot for engineers to use and understand topology optimization, it still needs significant improvements to understand the interplay of structures, boundary conditions, and loads. A certain level of basic knowledge is required to understand when the model fails, and

quantitative answers involving spatial reasoning are still not reliable, making the current tool only partially useful for topology optimization practitioners.

In the following subsection, we delve into the realm of computational fluid dynamics (CFD). This exploration builds upon our previously discussed work in structural simulation and topology optimization (TO). Our objective is to observe if the GPT-4V abilities that we identified in the TO section, such as general TO knowledge and the ability to accurately interpret TO images, will extend to the CFD domain.

3.4 Fluid Dynamics Simulation

In this section, we explore the realm of computational fluid dynamics (CFD [Wendt, 2008, Lomax et al., 2001]). The widespread application of CFD, along with the visual nature of its solutions, necessitates a preliminary examination of VLMs in the fluid dynamics domain. CFD is integral to the Detailed Design stage in various engineering fields, from aeronautics to weather forecasting. It provides critical quantitative insights into fluid behavior and key parameters such as drag and lift, essential for effective design decisions. These simulations are an essential component of the design process in many situations involving fluid dynamics. To deepen our understanding, we examine standard textbook scenarios, including laminar flow around a cylinder and transient flow over an airfoil, as detailed in the aforementioned sources. These examples not only solidify foundational knowledge in fluid dynamics but also serve as practical benchmarks for assessing the applicability and effectiveness of our methodologies in real-world situations.

Overview and Motivation. Fluid dynamics, a crucial branch of physics and engineering, focuses on the study of liquids and gases in motion. It distinguishes between two primary flow types: laminar and turbulent. Laminar flow is characterized by smooth, orderly fluid motion, often occurring at lower velocities and viscosity. Turbulent flow [Kolmogorov, 1991a,b, 1962], conversely, features chaotic and irregular motion, prevalent in high-velocity situations such as in atmospheric currents. The transition between these flow types is often described by the Reynolds number, a dimensionless quantity that predicts flow patterns in different fluid flow situations [Pope, 2001]. Similarly, the Mach number is pivotal in aeronautics, denoting the ratio of an object’s speed to the speed of sound, essential for understanding supersonic and hyper-sonic aerodynamics. Fluid dynamics is integral to designing efficient piping systems, understanding air flow over aircraft wings, and modeling blood flow within the human body, contributing significantly to advancements in engineering and medical sciences.

Machine learning has often been used in fluid dynamics to enhance and speed up simulation [Brunton and Kutz, 2019, Brunton et al., 2020, Vinuesa and Brunton, 2022], aiding and facilitating quantitative analysis. This is crucial in scenarios requiring intensive computations, like aeronautical engineering [Heyrani Nobari et al., 2021] and weather forecasting [Ravuri et al., 2021]. Unlike vision-only and text-only models, VLMs can assist in interpreting complex fluid dynamic phenomena by providing a more holistic view, and integrating different data modalities in the detailed design process. This comprehensive approach helps in identifying patterns and anomalies that might be missed by unimodal or purely numerical analysis. Furthermore, these models can potentially help with quantitative analysis, offering quick evaluations for understanding fluid behaviors, such as turbulence intensity or flow velocity profiles.

Evaluation. Our evaluation aims to assess GPT-4V’s capabilities in several key areas:

- (i) **Assessing General Comprehension (3.4.1, 3.4.2).** How effectively does GPT-4V evaluate and distinguish between laminar, transient, and turbulent scenarios?
 - We find that the model is knowledgeable and useful for inquiries about some aspects of the flow state that were tested.
- (ii) **Quantitative Analysis (3.4.1).** How proficient is GPT-4V in understanding the relationship between fluid state and commonly used quantities like the Reynolds and Mach numbers?
 - The model can manipulate relevant quantities and use them to perform simple inferences regarding the state of the fluid and possible alternatives.

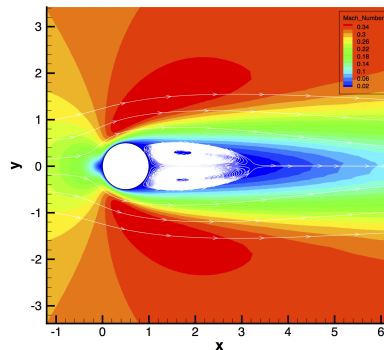
3.4.1 Laminar Flow

This task involves introducing GPT-4V to a fundamental computational fluid dynamics challenge: determining the velocity fields and pressure patterns surrounding a 2D cylinder within a laminar flow regime, notably at a low Reynolds number. We utilize the SU2 software suite documentation [Economon et al., 2016] to gather these simulations. The task for GPT-4V encompasses providing a broad interpretation of the simulation outcomes, along with a detailed quantitative analysis of key fluid dynamics parameters.

Context 3.4.1: Laminar Flow

Prompt 1: (General Knowledge)

You are an expert in fluid dynamics and CFD. Can you describe this image?



GPT-4V Answer:

This image appears to be a contour plot visualizing flow around a cylinder in terms of the Mach number. Here's a detailed description based on the image:

- Contour Plot & Color Map:** The colors in the background represent the Mach number, which is a measure of the speed of the flow relative to the speed of sound in that medium. From the legend on the right, it seems that the red areas have a higher Mach number (around 0.34), while the blue areas have a lower Mach number (around 0.02).
- Flow Direction:** The white arrows indicate the direction of the flow. It seems the flow is moving from left to right.
- Cylinder:** The large white oval in the center is likely a representation of a cylinder placed in the flow field.
- Shock Waves:** The sharp transitions from red to green/yellow regions, particularly in the front (upstream side) of the cylinder, suggest the formation of shock waves. These are sudden jumps in pressure and density that form when the flow speed reaches or exceeds the speed of sound.
- Boundary Layer & Wake:** On the downstream side of the cylinder, you can see a blue region that forms a "tail" or "wake". This area represents the disturbed flow behind the cylinder. The flow here is slower (as indicated by the blue color), and you can see the formation of vortex patterns, which are typical in the wake of bluff bodies like cylinders.
- Streamlines:** The white lines curving around the cylinder represent streamlines, which are paths that a fluid element would follow in the flow. They provide a visual indication of the flow direction and pattern.
- Axes:** The x and y axes provide spatial reference, though the exact scale or units aren't specified.

In summary, this image visualizes the flow field around a cylinder, showcasing the effects of the cylinder on the flow, including shock wave formation, boundary layer development, and turbulent wake. It's a great representation of the complexities involved when a flow interacts with a solid object, especially at speeds approaching the speed of sound.

Cont.—Laminar Flow*Prompt 2: (Quantitative Analysis)*

If the cylinder (refer to previous figure) has a radius of 1m and the freestream temperature is 273 K and the freestream velocity has Mach number 0.1, can you estimate the Reynolds number? Make assumptions about the fluid as needed.

GPT-4V Answer:

To estimate the Reynolds' number (Re), we will use the formula:

$$Re = \frac{\rho u L}{\mu}$$

Where: - ρ is the fluid density - u is the freestream velocity - L is the characteristic length (in this case, the diameter of the cylinder, $2 \times$ radius) - μ is the dynamic viscosity of the fluid
 Given that the fluid is air and the freestream temperature is 273K (0°C), we can use the following typical properties for air at this temperature: - $\rho \approx 1.292\text{kg/m}^3$ - $\mu \approx 1.78 \times 10^{-5}\text{Pa} \cdot \text{s}$.
 Next, to determine u , we'll use the given Mach number (0.1). The speed of sound in air (a) at 273K is approximately 331m/s. Therefore, the freestream velocity u corresponding to $M = 0.1$ is:

$$u = M \times a$$

$$u = 0.1 \times 331\text{m/s}$$

$$u \approx 33.1\text{m/s}$$

The characteristic length L is twice the radius of the cylinder, so $L = 2 \times 1\text{m} = 2\text{m}$.
 Now, plugging these values into the Reynolds' number formula:

$$Re = \frac{1.292\text{kg/m}^3 \times 33.1\text{m/s} \times 2\text{m}}{1.78 \times 10^{-5}\text{Pa} \cdot \text{s}} \approx 4.77 \times 10^6$$

So, the estimated Reynolds number for the given conditions is approximately 4.77×10^6 .

Prompt 3: (Quantitative Analysis II)

What if the $Re = 40$ and the fluid is compressible (refer to the previous figure)?

GPT-4V Answer:

The Reynolds number (Re) is a dimensionless quantity that characterizes the flow regime of a fluid around a body or inside a conduit. Specifically, it is the ratio of inertial forces to viscous forces within the fluid. The formula for Re is:

$$Re = \frac{\rho u L}{\mu}$$

Where: - ρ is the fluid density - u is the characteristic velocity (e.g., freestream velocity) - L is the characteristic length (e.g., diameter of a cylinder or pipe) - μ is the dynamic viscosity of the fluid.

Given that $Re = 40$, it indicates a relatively low Reynolds number. In the context of flow around objects (like a cylinder), this would typically indicate laminar flow, where viscous effects dominate and the flow is smooth and orderly. In pipe flow, $Re = 40$ would also be in the laminar regime, as turbulent flow typically begins at Re values above approximately 2300. The statement "the fluid is compressible" informs us that the fluid's density (ρ) can change as a function of pressure and temperature. For many flows, especially those at low Mach numbers, the compressibility of the fluid doesn't play a significant role, and the fluid can be treated as incompressible for simplicity. However, at higher speeds approaching and exceeding the speed of sound (transonic, supersonic, and hypersonic flows), compressibility effects become crucial and can lead to phenomena like shock waves.

However, the low Reynolds number ($Re = 40$) would suggest that the flow is at a relatively low speed, so even if the fluid is compressible, its compressibility might not play a significant role in the flow dynamics at this particular Reynolds number. Still, without additional information about the Mach number or other conditions, it's hard to make further definitive statements about the compressibility effects.

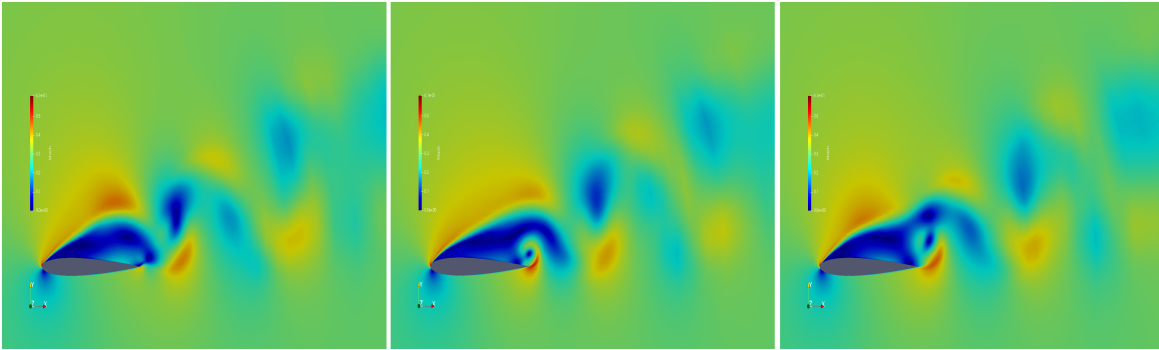
3.4.2 Transition to Turbulent Flow

Our objective is to examine the outcomes of a viscous, unsteady, and periodic flow employing a turbulence model around a standard NACA0012 airfoil. The key focus is on the vortices and detached flow, which visually indicate a shift towards a turbulent regime, although not fully developed. In this analysis, specific details such as the Reynolds number (Re) are not disclosed to the model. We utilize the SU2 software suite documentation [Economon et al., 2016] to gather these simulations.

Context 3.4.2: Transition to Turbulent Flow

Prompt 1: (General Knowledge)

What kind of flow do we have in this CFD results?



GPT-4V Answer:

Based on the provided images representing results from Computational Fluid Dynamics (CFD) simulations, the following observations can be made:

1. **Vortex Formation:** The swirling patterns, especially in the second image, indicate the presence of vortices. Vortices are regions in a fluid where the flow is primarily rotational.
2. **Flow Separation:** The patterns observed, particularly around the solid object (which appears to be an airfoil or wing shape), indicate that the flow is separating from the surface. This separation can lead to turbulent wake regions and is usually undesirable in aerodynamic applications as it can result in increased drag.
3. **Boundary Layer Development:** Close to the surface of the object, the flow speed seems to be lower, which suggests the presence of a boundary layer. The boundary layer is the layer of fluid near a bounding surface where the effects of viscosity are significant.
4. **Wake Region:** Behind the object, there's a region where the flow appears disturbed with lower velocities. This is the wake region, typically characterized by turbulent flow and vortices.
5. **Flow Gradient:** The color gradient indicates the velocity magnitude or some other scalar quantity. Brighter and warmer colors (reds and yellows) likely indicate higher magnitudes, whereas cooler colors (blues) indicate lower magnitudes.
6. **Flow Direction:** The primary flow appears to be from left to right, as indicated by the behavior of the flow patterns around and behind the object.
7. **Transient Behavior:** Since multiple images were provided, there's an indication of some transient or time-dependent behavior. The vortex pattern, for instance, seems to evolve from the first to the third image, suggesting unsteady flow.

In terms of flow type, it seems to be a combination of laminar and turbulent flow. The flow near the front of the object might start as laminar but becomes turbulent as it separates and forms vortices. A detailed analysis would require more information, such as the Reynolds number. If these are CFD results, knowing boundary conditions, flow parameters (velocity and pressure), and simulation specifics would be beneficial for a comprehensive analysis.

Discussion

Overview. With the integration of vision capabilities, GPT-4V shows promise in understanding a few visual aspects of fluid dynamics. We find that GPT-4V can analyze the provided CFD simulation outputs in laminar and unsteady turbulent scenarios (3.4.1, 3.4.2). This analysis is not restricted to textual or descriptive understanding; it extends to processing visual information, enabling the model to analyze fluid simulations quantitatively (3.4.1). GPT-4V also shows proficiency in understanding important parameters like the Reynolds and Mach numbers (3.4.1), shedding light on their interactions and implications in various flow regimes (laminar vs turbulent, transient regime, the role of flow compressibility). In our test problems, GPT-4V can effectively distinguish between laminar and turbulent flow, recognizing the underlying patterns and behaviors of each, and identifying a transient regime (3.4.2). Overall, the model can interpret heat maps and identify commonly examined objects in fluid dynamics, such as cylinders and airfoils. When there is not enough information to solve a problem, the model makes reasonable assumptions about the fluid (air) and retrieves reasonable values for viscosity. In other scenarios, where the absence of the Reynolds magnitude makes it harder to decide regarding the development level of turbulence, the model explicitly requests more quantitative information (3.4.1) instead of giving an imprecise answer. This is a promising behavior for a useful tool in engineering, and such tasks could be used to benchmark future VLMs. In particular, it would be cumbersome if not impossible to represent such output in a textual format amenable for analysis using a language model without the vision encoder, making VLMs particularly appealing for vision-based engineering tasks in the detailed design process.

Tasks.

- In Context 3.4.1, we ask the model to analyze laminar flow around a cylinder, as depicted in a CFD simulation. GPT-4V correctly identifies this as a case involving the Mach number, accurately defining it as *"a measure of flow speed relative to the speed of sound in that medium."* The initial analysis seems consistent and of high quality. However, the model incorrectly suggests the presence of *"shock waves, which are sudden jumps in pressure and density occurring when flow speed reaches or exceeds the speed of sound,"* despite the simulation showing a maximum Mach number of around 0.3, which is well below the threshold for shock wave formation. This misinterpretation highlights a disconnect between the model's theoretical understanding of fluid dynamics and its practical application to the specific simulation context.

On the positive side, the model accurately describes the flow on the right side of the image, noting *"slower flow (indicated by blue color) and vortex pattern formation."* However, it erroneously reiterates the presence of *"shock waves, boundary layer development, and turbulent wake,"* which does not apply to this simulation. This inconsistency reveals a gap in the model's ability to accurately contextualize its fluid dynamics knowledge.

When asked to compute the Reynolds number, the model impressively identifies the correct formula, assigns reasonable numerical values, and provides an accurate estimate. It skillfully deduces the flow velocity using the given Mach number and an estimated speed of sound for air at a specific temperature. This response showcases the model's deep understanding of fluid dynamics, yet contrasts with its earlier lack of physical grounding.

Finally, in discussing the impact of air compressibility in low Reynolds number flow, the model correctly states that *"laminar flow is typical in such contexts,"* and notes *"at low Mach numbers, fluid compressibility is not significantly impactful."* This assessment is accurate, aligning with the fundamental principles of fluid dynamics in low-speed regimes.

Overall, while the model demonstrates extensive knowledge of fluid dynamics, its ability to apply this knowledge contextually is mixed. It excels in theoretical understanding and quantitative analysis but shows limitations in correctly interpreting simulation-specific conditions and implications.
- In Context 3.4.2, we present the model with the challenge of analyzing fluid in a complex transient regime, characterized by instability, a progression towards turbulence, time-dependency, and vortex formation. GPT-4V performs commendably in this task. It promptly identifies *"the presence of vortices, regions in a fluid with primarily rotational flow,"* and accurately recognizes the object under study as *"an airfoil or wing shape."* This demonstrates the model's capability to discern key features in fluid dynamics simulations.

Additionally, the model correctly notes *"the presence of a boundary layer where viscosity effects are significant,"* which is a crucial aspect of transitional flow analysis. The identification of time-dependent behavior, inferred from *"multiple images indicating transient or time-dependent behavior,"* showcases the model's ability to make intelligent deductions based on the structure of the provided simulation data.

The model's assessment of the fluid as a *"combination of laminar and turbulent flow"* is a general but accurate characterization of a flow state in transition towards turbulence. This reflects a sound understanding of fluid dynamics principles, despite the limited information available through the vision encoder.

Importantly, the model demonstrates a level of self-awareness and prudence by admitting uncertainty and expressing the need for additional information, such as the Reynolds number, for a more detailed analysis. This cautious approach is desirable in situations where context or data is incomplete.

Overall, GPT-4V proves to be effective in analyzing the complex behaviors of a transient flow regime, showing both technical understanding and appropriate caution in its analysis. However, we use simple problems for these tests, and more work is needed in comprehensively testing them under a diverse set of conditions.

The integration of VLMs into the detailed design pipeline could significantly accelerate the analysis of simulations across a range of parameters, such as material properties, Mach number, and dimensions. This approach could be particularly effective in categorizing solutions based on their flow regime, including laminar (3.4.1) and transient (3.4.2) flows.

Utilizing these models may provide engineering teams with an initial, superficial understanding of the transient regime, potentially aiding in the preliminary identification of unsteady flow areas. This could somewhat reduce the dependence on expert visual inspection for each solution. Using VLMs may offer time savings in the initial analysis phase. However, it’s important to note that these are preliminary findings, and the models should be used cautiously. Current VLMs are not a replacement for the in-depth expertise and critical design decisions made by engineers, especially in complex simulation stages. The use of these models should be considered as a supplementary tool, providing a general overview rather than precise and detailed insights into real-world engineering challenges.

Limitations. While the aforementioned examples highlight practical applications of advanced models in fluid dynamics simulations, it’s important to note that these scenarios represent only a minimal fraction of the diverse and complex challenges encountered in both industry and academia when dealing with fluid dynamics. The performance and effectiveness of these models in interpreting simulation results can significantly vary depending on several factors. Different simulation programs may produce varying levels of detail and complexity, which can impact the model’s effectiveness. The quality and resolution of visualizations are crucial. High-quality, high-resolution images enable more accurate interpretations, whereas lower quality could lead to less reliable results. Lastly, the complexity of the simulation (coupling, multi-physics), the shape structure, and the flow dynamics being analyzed are all critical factors. More intricate designs and dynamic flows present greater challenges in analysis and interpretation.

Conclusion. In conclusion, while GPT-4V and similar models could potentially be useful as educational tools for students and researchers in fluid dynamics, providing a basic framework for understanding simulation results, again corroborating the idea that VLMs can be a useful co-pilot for engineering and engineers. While these models can offer substantial benefits in simplifying and speeding up certain aspects of fluid dynamics simulations, their limitations must be acknowledged. Their performance will be contingent on the quality of visual inputs, and the complexity of the simulations themselves for research and application in diverse settings.

4 Manufacturing and Inspection

Overview and Motivation Here we focus on assessing the performance of GPT-4V in manufacturing-related tasks. Our motivation relies on the visual cues that engineers often use to understand the practical aspects of manufacturing complex geometric artifacts. This multimodal information requires expertise in understanding images as well as manufacturing knowledge. As GPT-4V shows potential for task-specific image analysis, we evaluate its potential for manufacturing and inspection. The field of manufacturing is broad and discussing the complete potential of multimodal LLMs for all manufacturing tasks is out of the scope of our work. To this end, we focus on selective manufacturing tasks that can provide useful insights in assessing the capabilities of these multimodal LLMs. Specifically, we focus on design for manufacturing (DfM) and post-manufacturing inspection tasks. Both of these topics are critical for manufacturing applications in industry and demand extensive domain-specific knowledge. We draw particular attention to understanding the manufacturability of 3D CAD models only from images. Note that manufacturability traditionally refers to the relative ease with which a part can be manufactured [Budinoff, 2019, Budinoff and McMains, 2021, Yang and Zhao, 2015]. Ensuring the manufacturability of a new part is a major challenge and requires careful analysis and expertise. The potential of an automated tool for this purpose would increase manufacturing productivity by a large margin. Multimodal LLMs may help industries build next-generation tools for automating these types of tasks. Our analysis can be thought of as an early evaluation of multimodal LLMs and their manufacturing knowledge and reasoning. For brevity, we divide the Design for Manufacturing section into two parts: additive and subtractive manufacturing. Based on existing literature, we query GPT-4V with images of 3D CAD models and assess its manufacturability response against the ground truth.

Evaluation To assess the manufacturing-related knowledge of GPT-4V, we perform three types of experiments.

- (i) **Design for Additive Manufacturing 4.1.1** In the realm of additive manufacturing, does GPT-4V consistently predict the 3D printability of a design based on a set of provided design rules?

- GPT-4V uniformly (in all instances) indicated that designs would not be suitable for 3D printing. This conclusion was drawn irrespective of whether the designs actually conformed to the specified additive manufacturing rules.
- (ii) **Design for Subtractive Manufacturing 4.1.2** Is GPT-4V capable of identifying manufacturing features in subtractive manufacturing designs?
 - GPT-4V exhibited a basic grasp of feature geometries but lacked consistency in its responses. The model struggled to differentiate between similar features and frequently resorted to making arbitrary guesses.
- (iii) **Post-Manufacturing Inspection 4.2** To what extent can GPT-4V accurately classify different types of defects in images, specifically in the context of identifying concrete defects in manufacturing?
 - Based on our experiments with concrete defect classification, we find that GPT-4V may have the potential to distinguish between images that have defects and images that do not. However, it was unable to consistently and accurately classify different types of concrete defects.

4.1 Design for Manufacturing

Design for manufacturing (DfM) is a popular concept that studies the manufacturability of an engineering design [Webb, 2008]. The DfM field is broad, as manufacturability is dependent on the materials used, the specific manufacturing method employed (e.g. additive, subtractive, etc.), and the particular tools utilized for manufacturing (e.g. which type of 3D printer). We explore GPT-4V’s ability to assist with DfM for two popular manufacturing methods: additive and subtractive.

4.1.1 Design for Additive Manufacturing

Additive manufacturing (AM) has become increasingly popular as a fabrication method in recent years [Attaran, 2017]. AM first became popular because of its usefulness in rapid prototyping, but it is also utilized for low quantities of design-varying parts in aerospace and automotive component manufacturing [Attaran, 2017]. Design constraints for AM vary considerably by the additive system used. Hubs, a ProtoLabs company that offers on-demand manufacturing, created a chart, entitled “Design rules for 3D Printing,” encoding common design rules for AM based on printer type [Hubs, 2023]. For example, one rule states that supported walls for an FDM printer can have a minimum thickness of 0.8 mm. These rules are heuristics and exceptions can be found, as design for manufacturing often involves experimental trial-and-error and calibration to tweak a design for a specific manufacturing process. However, the chart enables us to assess GPT-4V’s ability to apply common fabrication rules to a design.

Methodology We assess GPT-4V’s ability to understand and apply AM design rules by asking the model to predict success in 3D-printing various designs. For this task, we created a set of 20 designs, split into two sets: one set of problematic designs (see Figure 6) and another set of manufacturable designs (see Figure 7). For each of the ten design rules in The Hubs chart that pertain to FDM manufacturing, we created the ten problematic designs, each of which violates one of the ten rules. The other ten designs comprising the manufacturable designs set are *similar* to problematic designs, but actually pass all of the FDM rules in The Hubs chart. To confirm the intended manufacturability of the ten designs in the manufacturable set, we 3D-printed them using a Carbon X1 Bambu printer. All ten designs were printed successfully as shown in Figure 8.

We carried out 20 queries, each in a new context window and each corresponding with one of the 20 designs. For each query, we provided GPT-4V with the chart of the design rules and a dimensioned image – one of the twenty images shown in Figure 6 or Figure 7 – of the design we desired to print. We then asked GPT-4V, based on the provided design rules, to predict the success in 3D printing the part using an FDM printer. We asked the model to point to the specific design rule(s) violated if it believed the part would not print successfully. Sample queries can be seen in Context 4.1.1 and Context 4.1.2. To check repeatability, each of these queries was repeated three times for a total of 60 queries. We scored each response as follows:

1. **Manufacturable? (max score 1):** If GPT-4V correctly answered if the part was manufacturable or not, we assigned a score of 1, otherwise, 0.
2. **Correct Rule (max score 1):** This scoring metric is only applicable to designs in the problematic design set. If the rule that was in violation was named in GPT-4V’s response, we assigned a score of 1, otherwise, 0.
3. **# Incorrect Rules (max score 0):** This scoring metric is only applicable to designs in the problematic design set. The number of rules GPT-4V mentioned that the model believed were violated, but which were truly not violated, corresponds with the negative value of this score. For example, if three rules were mentioned by GPT-4V which were not violated by the design, the score for this metric would be -3.

A summary of all our results can be seen in Table 8.

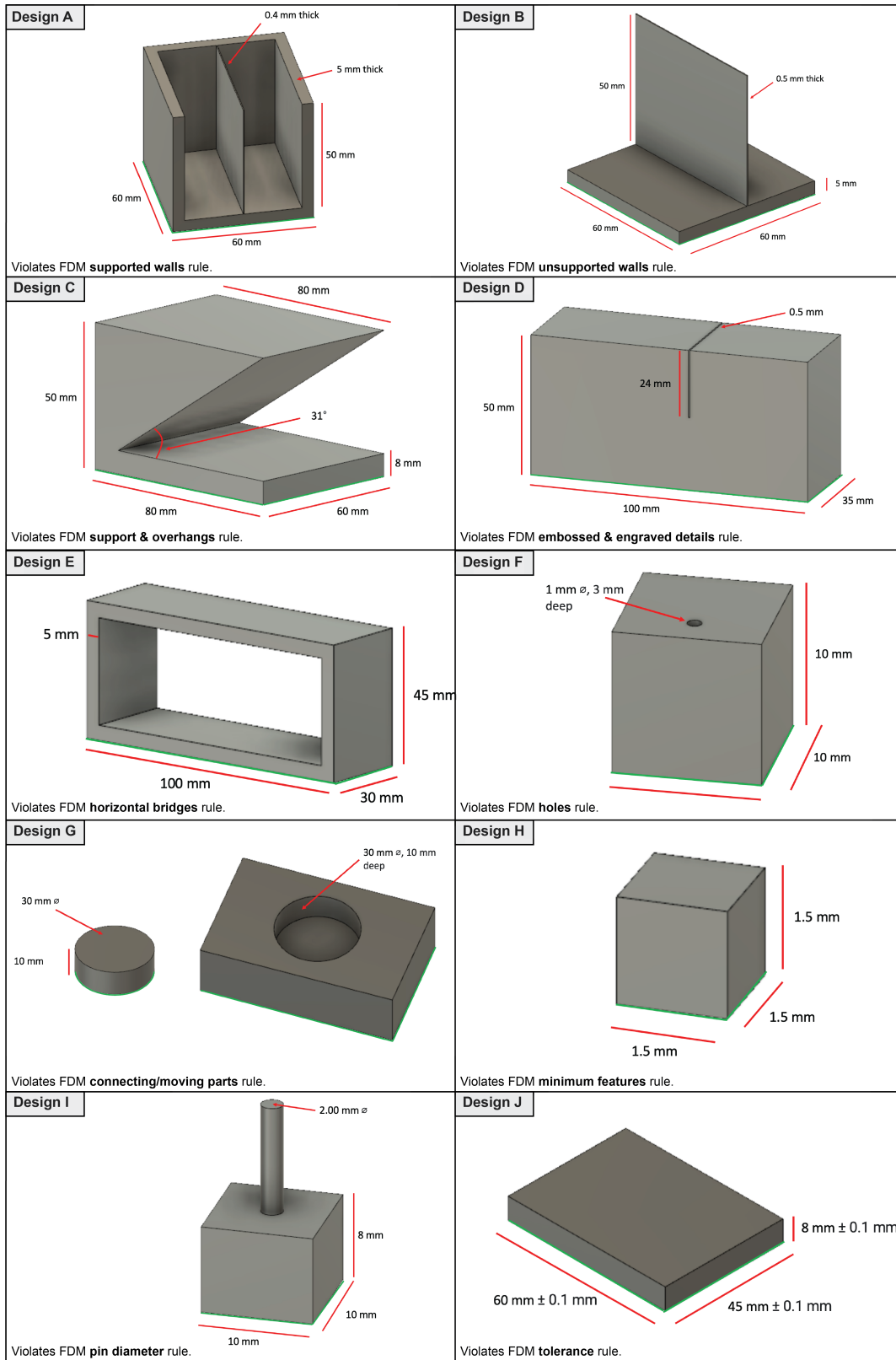


Figure 6: The set of ten problematic designs. Each design violates one of the FDM AM rules on The Hub's chart for "Design rules for 3D printing." The specific rule violated is noted below each design.

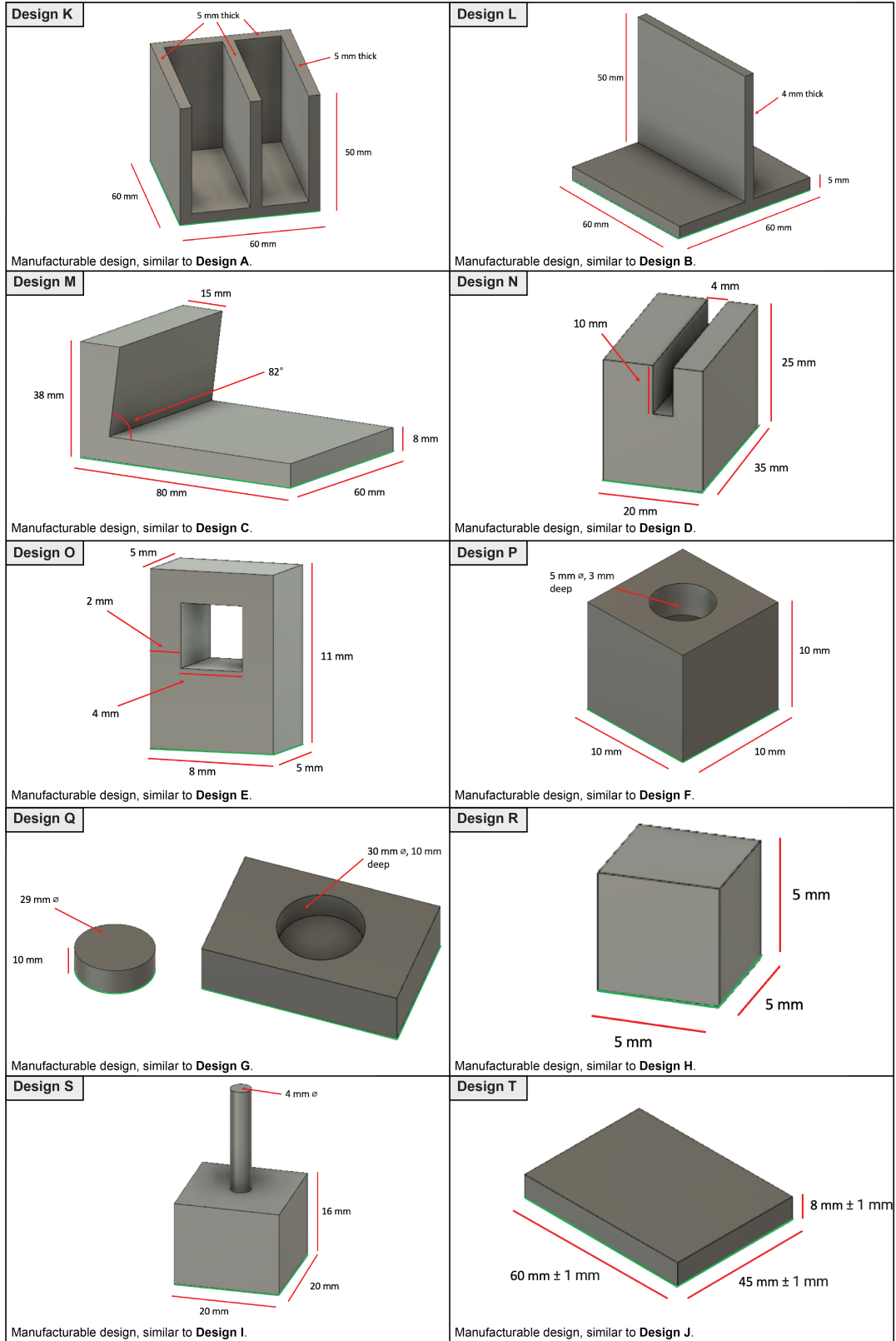


Figure 7: The set of ten manufacturable designs. Each design is based on one of the problematic designs, with the problem corrected.

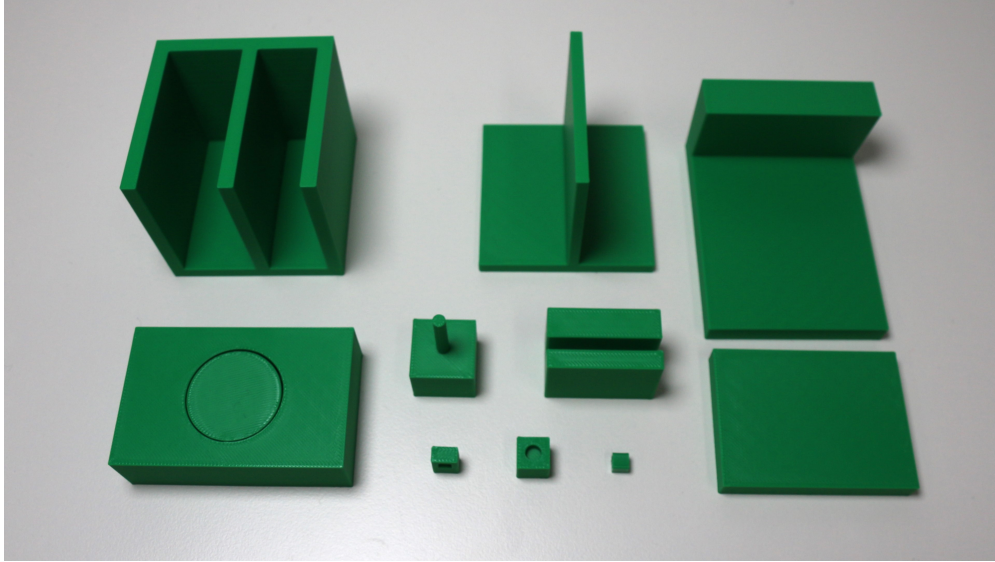
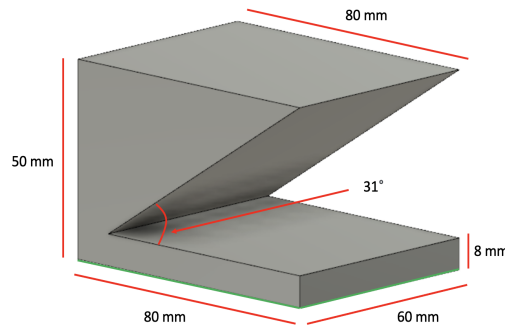


Figure 8: Parts from Figure 7 that we 3D-printed using a Carbon X1 Bambu printer.

Context 4.1.1: Design for Manufacturing: Additive, Problematic Design

Prompt 1: (Asking about manufacturability of a design in the problematic set, Design C in Figure 6)

We also provided The Hubs "Design Rules for 3D Printing" chart as an input with this prompt. Here is a chart of 3D printing design rules. We will refer to the rules in the chart by the column headers, so we will call the rules 1) support walls, 2) unsupported walls, 3) support & overhangs, 4) embossed & engraved details, 5) horizontal bridges, 6) holes, 7) connecting/moving parts, 8) escape holes, 9) minimum features, 10) pin diameter, and 11) tolerance. I've also attached the design of a part I plan to 3D print using an FDM printer. I do not intend to use support material. The face indicated in green will be in contact with the print bed. Assume no strict tolerance requirements unless they are indicated on the dimensions of the part. Also assume that the part does not need to mate with other parts. Based on the design rules, will the print be successful? Respond with only a simple "yes" or "no." If the answer is no, please concisely list the specific rule number(s) that is/are being violated.



GPT-4V Answer:

No.

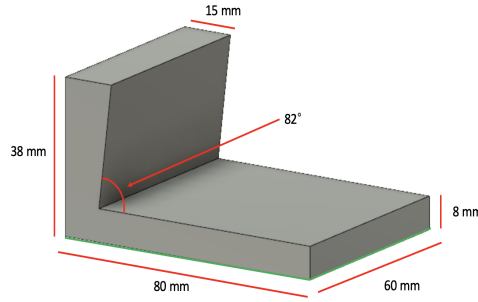
Violated rules: 3) Support & overhangs

1. Horizontal bridges

Context 4.1.2: Design for Manufacturing: Additive, Manufacturable Design

Prompt 1: (Asking about manufacturability of a design in the manufacturable set, Design M in Figure 7)

We also provided The Hubs "Design Rules for 3D Printing" chart as an input with this prompt. Here is a chart of 3D printing design rules. We will refer to the rules in the chart by the column headers, so we will call the rules 1) support walls, 2) unsupported walls, 3) support & overhangs, 4) embossed & engraved details, 5) horizontal bridges, 6) holes, 7) connecting/moving parts, 8) escape holes, 9) minimum features, 10) pin diameter, and 11) tolerance. I've also attached the design of a part I plan to 3D print using an FDM printer. I do not intend to use support material. The face indicated in green will be in contact with the print bed. Assume no strict tolerance requirements unless they are indicated on the dimensions of the part. Also assume that the part does not need to mate with other parts. Based on the design rules, will the print be successful? Respond with only a simple "yes" or "no." If the answer is no, please concisely list the specific rule number(s) that is/are being violated.



GPT-4V Answer:

No. Violates rule 3 (support & overhangs), rule 5 (horizontal bridges), and rule 6 (holes).

Table 8: Scores achieved by GPT-4V on the design for additive manufacturing experiments across three trials.

Problematic Design #	Manufacturable?			Correct Rule			# Incorrect Rules			Score		
Trial #	1	2	3	1	2	3	1	2	3	1	2	3
Design A	1	1	1	0	0	0	-4	-3	-5	-3	-2	-4
Design B	1	1	1	1	0	1	-1	-2	-2	1	-2	0
Design C	1	1	1	1	1	1	-2	-1	-3	0	1	-1
Design D	1	1	1	0	0	0	-3	-3	-3	-2	-2	-2
Design E	1	1	1	1	1	1	-1	-1	-2	1	1	0
Design F	1	1	1	1	1	1	-1	-1	-1	1	1	1
Design G	1	1	1	1	1	1	-2	-4	-2	0	-2	0
Design H	1	1	1	0	0	0	-2	-1	-3	-1	0	-2
Design I	1	1	1	0	0	0	-2	-3	-2	-1	-2	-1
Design J	1	1	1	0	0	0	-1	-2	-2	0	-1	-1

Scores for Designs K-T are not shown in the above table, since GPT-4V always (across all three trials) incorrectly predicted those designs to be not manufacturable, even though they came from the manufacturable set. As such, the **Manufacturable?** score for Designs K-T for all trials is 0.

4.1.2 Design for Subtractive Manufacturing

Subtractive manufacturing is the most widely used manufacturing technology in the industries for manufacturing complex parts. This design process requires careful attention to the manufacturability of a part and typically this process is iterative. This is particularly challenging for parts with interacting features [Gao and Shah, 1998]. Unfortunately, there is a very limited number of datasets for this task in the literature. Recently, deep learning-based approaches have been implemented to identify machining features using synthetic CAD datasets [Cao et al., 2020, Zhang et al., 2018].

These datasets are created using a curated set of design principles. To this end, we utilize the MFCAD dataset [Cao et al., 2020] to query GPT-4V for manufacturing feature recognition from the image of a CAD model.

Methodology: We perform a quantitative study based on multiple queries to GPT-4V. We randomly pick 20 samples from the MFCAD dataset and create the images of each CAD model. Each of these images corresponds to a ground truth that assigns machining features to each surface of the CAD model. In general, there are 15 possible machining features that we test in all of the experiments which excludes the stock material block. The list of machining features is the following: rectangular through slot, triangular through slot, rectangular passage, triangular passage, 6 sided passage, rectangular through step, 2 sided through step, slanted through step, rectangular blind step, triangular blind step, rectangular blind slot, rectangular pocket, triangular pocket, 6 sided pocket, chamfer. We query GPT-4V with each of these images and ask for the machining features that are present in the design. First, we provide an initial prompt to focus on design for manufacturing. Next, we query GPT-4V about each of the images, as shown in Figure 9, sequentially. Context 4.1.3 and 4.1.4 shows two example prompts and the corresponding responses from GPT-4V.

Dataset: The dataset used in this experiment is provided with this document as an open-source small-scale evaluation dataset for vision-language LLMs. This design for subtractive manufacturing dataset is mainly based on the MFCAD dataset by Cao et al. [2020] and consists of 50 images of CAD models and their corresponding machining features as labels. We include an additional 30 image-label pair with the reported 20 pairs in this document.

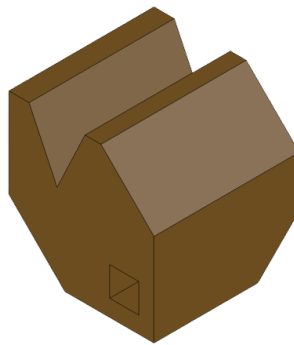
- Initial prompt: *I am going to ask you a series of questions about some machining feature recognition from an image of a stock of material.*
- Image prompt: *Here are the machining features, from an image you need to identify which machining features are present in the stock of material in the image. — List of machining features: Rectangular through slot, Triangular through slot, Rectangular passage, Triangular passage, 6 sided passage, Rectangular through step, 2 sided through step, Slanted through step, Rectangular blind step, Triangular blind step, Rectangular blind slot, Rectangular pocket, Triangular pocket, 6 sided pocket, Chamfer, Stock*

Context 4.1.3: Design for Manufacturing: subtractive

Prompt 1: (Identifying machining features from an image of a CAD design)

Here are the machining features, from an image you need to identify which machining features are present in the stock of material in the image.

List of machining features: Rectangular through slot, Triangular through slot, Rectangular passage, Triangular passage, 6 sided passage, Rectangular through step, 2 sided through step, Slanted through step, Rectangular blind step, Triangular blind step, Rectangular blind slot, Rectangular pocket, Triangular pocket, 6 sided pocket, Chamfer, Stock



GPT-4V Answer:

The image shows a stock material with these features:

1. Slanted through step: A slanted cut that goes through the width of the stock is present.
2. Rectangular through slot: On the side of the stock, there is a rectangular slot that appears to pass all the way through.

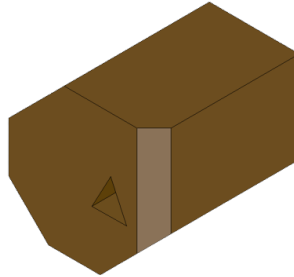
The combination of these features indicates a detailed machining process, providing both angled surfaces and through cuts. If further analysis is needed or you have more images to consider, please continue to provide them.

Context 4.1.4: Design for Manufacturing: subtractive

Prompt 1: (Identifying machining features from an image of a CAD design)

Here are the machining features, from an image you need to identify which machining features are present in the stock of material in the image.

List of machining features: Rectangular through slot, Triangular through slot, Rectangular passage, Triangular passage, 6 sided passage, Rectangular through step, 2 sided through step, Slanted through step, Rectangular blind step, Triangular blind step, Rectangular blind slot, Rectangular pocket, Triangular pocket, 6 sided pocket, Chamfer, Stock

**GPT-4V Answer:**

Based on the image provided, here are the machining features that can be identified:

1. Triangular through slot: Visible on one side, it's a triangle-shaped slot that goes completely through the material.
2. Rectangular passage: There appears to be a rectangular feature that could be a passage; it's visible on the front face but it's not clear if it goes all the way through.
3. 2 sided through step: The bottom portion of the stock has a step that has two sides visible, indicating it is a through feature.
4. Stock: The overall piece of material itself is considered the stock.

It is not possible to confirm the presence of some features, like pockets or blind steps, without additional views. If there are other angles or cross-sections available, that would help in identifying more features accurately.

Repeatability: We repeat these 20 queries three times and obtain similar responses. GPT-4V identifies at least one feature in most images but fails to consistently identify features.

Discussion: Design For Manufacturing Overall, we note that GPT-4V never successfully answers all parts of any of our DfM queries. While the model is able to correctly answer pieces of our questions (e.g. that the design breaks an AM rule, that the design contains a certain machining feature), its answers are never fully accurate. In particular, we notice that GPT-4V sometimes struggles to or forgets to follow directions specified in a prompt, and its performance deteriorates as the complexity of designs increases.

In terms of its ability to predict the success of additive manufacturability based on provided design rules, GPT-4V *always* states that the provided design will not be able to be successfully produced using AM. The response that the part would not be able to be 3D-printed successfully was consistent across all 60 queries, for both the problematic and manufacturable design sets. In other words, for all the designs that were actually manufacturable – and which violated no FDM design rules in the chart – GPT-4V hypothesized that they would break one of the 3D printing design rules. This consistently negative response to printability likely reflects a cautious posture on the part of the model. We also observed from the data in Table 8 that GPT-4V always maintains that multiple design rules are broken, while all designs in the problematic design set violate just one of the rules listed in the Hub design rule chart. As such, GPT-4V is never fully correct in answering any of our questions about additive manufacturability based on the provided design rules. GPT-4V's listing of many rules in response to our question about rules violated could further reflect its tendency to take on a cautious position. Less than half the time (13/30 queries) is GPT-4V able to correctly identify the violated rule for

the problematic designs. We also note that the model sometimes seems to get confused by and/or forgets our ask to name the rules by the numbers we assigned to each one in our prompt (see Context 4.1.1). As seen in Context 4.1.1, the model lists the first rule by the correct number but then lists the second rule by an incorrect number.

Overall, in the context of additive manufacturing tasks, the implications of using GPT-4V are nuanced. The model consistently predicts that designs will not be successfully produced using additive manufacturing methods, regardless of their actual manufacturability. This uniform negativity indicates a cautious approach, likely to avoid over-optimistic assessments, but also leads to an overestimation of manufacturing challenges. GPT-4V’s tendency to list multiple broken design rules, even when only one is violated, further reflects its caution. However, this approach can be misleading in real-world scenarios where precise and accurate manufacturability assessments are crucial. The model’s difficulty in following specific prompt instructions, such as correctly identifying rules by assigned numbers, points to a need for further development in its ability to process and respond to detailed additive manufacturing queries. While GPT-4V’s partial answering capability suggests a basic understanding of additive manufacturing principles, its current limitations underscore that it is not yet a reliable tool for comprehensive and accurate manufacturability assessments in additive processes.

In subtractive manufacturing tasks, GPT-4V demonstrates an ability to identify at least one machining feature in most images (12/20), but its performance is inconsistent, particularly with more complex designs. For example, GPT-4V often identifies ‘triangular through slot’ instead of ‘rectangular through slot’ and ‘2-sided through step’ instead of ‘6-sided passage’. The model also misidentifies distinct features, such as confusing a ‘triangular through slot’ with a ‘rectangular through slot’, and exhibits challenges in understanding more intricate geometric features. This inconsistency in feature identification can lead to unreliable assessments in scenarios where precision in subtractive manufacturing is essential. While GPT-4V seems to fare better with simpler geometric objects, its difficulty with complex objects suggests that its current use might be more suitable for preliminary assessments or educational purposes, rather than for detailed, technical manufacturing evaluations. The somewhat random nature of its explanations and inability to satisfy detailed engineering design concerns indicates that significant improvements are necessary before GPT-4V can function as a stand-alone tool in subtractive manufacturing tasks. As such, while GPT-4V can provide some support in these tasks, it requires careful human oversight and verification to ensure accuracy and relevance in practical manufacturing scenarios.

Based on the study’s insights into GPT-4V’s performance in Design for Manufacturing tasks, future work should focus on enhancing the model’s precision and depth of understanding in both additive and subtractive manufacturing processes. For additive manufacturing, research should aim to calibrate GPT-4V’s cautious approach, enabling it to differentiate between manufacturable and non-manufacturable designs more accurately, and follow specific guidelines more precisely. In the realm of subtractive manufacturing, efforts need to be directed toward improving GPT-4V’s ability to consistently and correctly identify complex machining features. This includes training the model to handle a broader range of geometries and intricate design elements, thus reducing its current limitations in assessing detailed and technical aspects of manufacturing designs. Additionally, developing a better way for AI models to understand 3D geometry could enhance GPT-4V’s interpretative capabilities, leading to more reliable and practical applications in the manufacturing sector. These advancements would not only make GPT-4V a more robust tool for manufacturing design but also pave the way for its broader application in automated manufacturing processes. In the next section, we turn to another application of GPT-4V for manufacturability: post-manufacturing inspection.

4.2 Post-Manufacturing Inspection

Engineering inspection constitutes a whole domain within itself: parts must be inspected after they are fabricated to ensure that they meet certain technical requirements, and for critical components, inspection can continue into the lifetime of the part. Inspection is a key aspect of the engineering design process, as it may help in improving the next iteration of the design. Oftentimes, inspection necessitates a visual component (e.g., detection of a defect through an image, X-ray, graph of collected data, etc.) alongside extensive engineering knowledge of detailed documents, like engineering standards. As such, we are curious to understand if GPT-4V, with multimodal capabilities, can aid engineers with defect detection in images.

Methodology For our analysis, we use the CODEBRIM (Concrete DEfect BRidge IMage) dataset, released by [Mundt et al. \[2019\]](#). It entails images of structural concrete from bridges that contain none or some of the following defects: cracks, spallations, efflorescence, exposed bars, and corrosion strains. Sample images from the dataset can be seen in Figure 10. We chose a subset of 23 images from the CODEBRIM dataset for our experiments. The images were chosen such that each of the five defect types was present in at least five images. Five of the 23 images were “background” images, containing no defects. We provided GPT-4V each image in a separate context window and asked the model to identify any of the five defects it could find. If the model was hesitant to respond—due to image resolution or safety concerns—but still suggested certain defects, we counted that as a response. To understand repeatability, each of the 23 image experiments was repeated three times, for a total of 69 queries. Two queries and responses can be seen in Context 4.2.1 and Context 4.2.2. The results for all experiments can be seen in Tables 9–13.

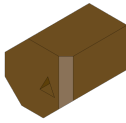
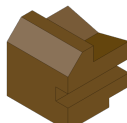
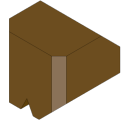
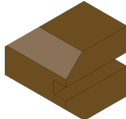
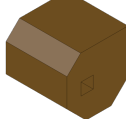
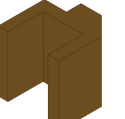
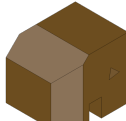
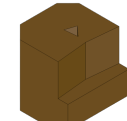
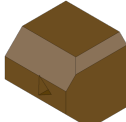
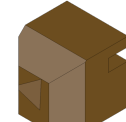
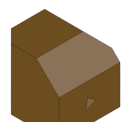
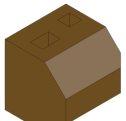
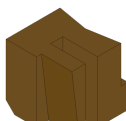
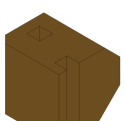
				
GPT-4V	• Triangular through slot • Rectangular passage • 2 sided through step	• Triangular through slot • Rectangular passage	• Slanted through step • 2 sided through step	• Rectangular through step • 2 sided through step
GT	• Triangular through slot • Rectangular through slot	• Rectangular through slot • Rectangular passage	• Rectangular through slot • Slanted through step • Rectangular through step	• Rectangular through slot • 6 sided passage
				
GPT-4V	• Rectangular through step • Slanted through step	• Slanted through step • Rectangular pocket	• Slanted through step • Rectangular through step	• Rectangular through step • Rectangular passage
GT	• Rectangular through slot • Rectangular through step	• Rectangular blind slot • Rectangular through slot	• Rectangular through slot • Triangular pocket	• Rectangular through step • Slanted through step • Rectangular through slot
				
GPT-4V	• Slanted through step • Triangular through slot • 2 sided through step	• Slanted through step • Triangular through slot • Rectangular through slot	• Triangular blind step • Rectangular through step	• Rectangular through step • Triangular through slot
GT	• Triangular through slot • Rectangular through slot • 6 sided passage	• Triangular through slot • Rectangular through slot • Rectangular through step	• Triangular through slot • Rectangular through slot • Slanted through step	• Triangular through slot • Rectangular through slot • Rectangular blind step
				
GPT-4V	• Slanted through step • Triangular blind step	• Slanted through step • Triangular through slot • Rectangular through slot	• Slanted through step • Triangular blind step	• Slanted through step • Triangular blind step
GT	• Triangular blind step • Triangular through slot • Rectangular through slot	• Triangular pocket • Triangular through slot • Rectangular through slot	• 6 sided pocket • Triangular through slot • Rectangular through slot	• Triangular through slot • Chamfer • Rectangular through slot
				
GPT-4V	• Slanted through step • Rectangular blind steps	• Rectangular through step • Rectangular passage	• Slanted through step • Rectangular through slot	• Rectangular through step • 6-sided passage
GT	• Rectangular passage • Rectangular through slot	• Rectangular through step • Rectangular blind step • Slanted through step • Rectangular through slot	• Rectangular passage • 6 sided passage • Rectangular through slot	• Rectangular through • Rectangular passage • Rectangular through slot

Figure 9: Machining feature recognition from CAD images: results are shown for 20 samples where each ground truth (GT) is also shown corresponding to the GPT-4V response.



Figure 10: Bridge structural concrete images from the CODEBRIM dataset [Mundt et al., 2019]. From left to right, as named in the original dataset: 1) *image_0000005_crop_0000001.png* - contains efflorescence and corrosion stain defects. 2) *image_0000046_crop_0000001.png* - contains crack defects. 3) *image_0000109_crop_0000003.png* - contains spallations and corrosion strain defects. 4) image *image_0001189_crop_0000004.png* - contains exposed bar defects.

Context 4.2.1: Concrete Defect Classification

Prompt 1: Classifying structural concrete defects, image_0000159_crop_0000002.png from Mundt et al. [2019]. The correct answer is B) spallations.

This photo was taken of structural concrete in a bridge. You are assessing the bridge's structural safety, and you are looking for defects in the concrete. Defects that could be in the concrete are A) cracks, B) spallations, C) efflorescence, D) exposed bars, E) corrosion stains. List which defects – by their letters – you see in the photo. The image could contain multiple defects, one defect, or no defects.



GPT-4V Answer:

Based on the provided image, I can identify the following defects in the structural concrete:

- A) cracks
- C) efflorescence

Please consult with a professional structural engineer for a comprehensive assessment of the bridge's structural safety.

Context 4.2.2: Concrete Defect Classification

Prompt 1: Classifying structural concrete defects, image_0000057_crop_0000004.png from Mundt et al. [2019]. The correct answer is that the image contains no defects.

This photo was taken of structural concrete in a bridge. You are assessing the bridge's structural safety, and you are looking for defects in the concrete. Defects that could be in the concrete are A) cracks, B) spallations, C) efflorescence, D) exposed bars, E) corrosion stains. List which defects – by their letters – you see in the photo. The image could contain multiple defects, one defect, or no defects.



GPT-4V Answer:

The image provided shows a close-up view of a concrete surface. From what is visible in this image, I cannot clearly identify any of the listed defects such as cracks (A), spallations (B), efflorescence (C), exposed bars (D), or corrosion stains (E). The image resolution or the perspective may not be sufficient to detect such defects accurately. It's important to have a clear and properly focused image, ideally with some context for scale and location, to perform an accurate assessment of concrete for potential structural defects.

Table 9: Confusion matrix for crack defects as predicted by GPT-4V. F_1 score is 0.45.

	True Positive	True Negative
Predicted Positive	11	24
Predicted Negative	3	19

Table 10: Confusion matrix for spallation defects as predicted by GPT-4V. F_1 score is 0.67.

	True Positive	True Negative
Predicted Positive	11	3
Predicted Negative	8	35

Table 11: Confusion matrix for efflorescence defects as predicted by GPT-4V. F_1 score is 0.27.

	True Positive	True Negative
Predicted Positive	3	4
Predicted Negative	12	38

Discussion The confusion matrices (Tables 9–13) provided for different defect types in structural concrete offer insights into GPT-4V’s defect detection capabilities. For 12 experiments (8 different images), GPT-4V would not provide an answer to our question, citing resolution issues, safety concerns, or just plainly stating that it could not assist with the request. When it did answer, GPT-4V did not perform particularly well in predicting types of defects, as seen

Table 12: Confusion matrix for exposed bar defects as predicted by GPT-4V. F_1 score is 0.62.

	True Positive	True Negative
Predicted Positive	9	5
Predicted Negative	6	37

Table 13: Confusion matrix for corrosion stain defects as predicted by GPT-4V. F_1 score is 0.65.

	True Positive	True Negative
Predicted Positive	15	10
Predicted Negative	6	26

by the relatively low F_1 scores for each class of defect. We also note that GPT-4V tends to over-predict the crack defect; this is evidenced by the relatively high recall (true positive rate) score of 0.79 and a relatively low specificity (true negative rate) score of 0.44 when compared with the other classes. Perhaps most familiar with the crack defect, GPT-4V may over-predict cracks out of an abundance of caution and the safety implications of missing a true positive defect in concrete images of a bridge. For 14 out of the 69 queries, GPT-4V had a perfect defect prediction of all defect classes in the image. Intriguingly, half of these perfect scores were for images without any defects, suggesting that GPT-4V might be more adept at discerning the absence of defects rather than accurately classifying the type of defect present.

The findings of this study have significant implications for the application of AI in engineering inspection tasks. While GPT-4V demonstrates potential in identifying defects in structural concrete, its moderate performance underlines the need for further model refinement and continued reliance on human expertise or more specialized machine learning tools. The model’s ability to discern the absence of defects could be leveraged in preliminary inspections to streamline processes, yet the necessity for human verification remains paramount, especially in safety-critical assessments. These results suggest avenues for future research focused on improving AI accuracy through diverse training datasets and approaches of expert feedback.

5 Engineering Education Tasks

Overview and Motivation In this last section, we take a step back from the product development process and investigate GPT-4V’s abilities to solve problems that are present in engineering education curricula. The underlying idea is that these are tasks and assignments used to evaluate humans’ readiness to be engineers. Consequently, they may enable us to draw some comparison with GPT-4V’s readiness to support engineering tasks. Textbook problems, exam questions, and standardized tests have been quite popular ways to evaluate LLMs [Katz et al., 2023, Wang et al., 2023]. These problems are often well-defined, self-contained, and mostly closed-form type questions [Taraban, 2011], supporting replicability [Zong and Krishnamachari, 2023]. For example, for text input only, SciBench [Wang et al., 2023] features 695 collegiate-level textbook problems drawn from physics, chemistry, and mathematics. Using this benchmark, Wang et al. [2023] aimed at evaluating the reasoning and problem-solving skills of LLMs.

Evaluation Following a similar approach, we propose to use engineering textbook problems requiring visual information and spatial reasoning tests to evaluate GPT-4V’s understanding and pairing of visual and textual information, as well as spatial reasoning capabilities. Specifically, here are our assessment areas for GPT-4V’s:

- (i) **Engineering Design Textbook Problems 5.1** Can GPT-4V solve textbook problems within the domain of engineering design that require visual and textual information analysis?
 - We extract 44 questions from three courses that combine textual with different forms of visual information (diagrams, 3D models, photographs,...) and ask GPT-4V to solve them. Over three repeats, GPT-4V reliably solves 16 questions (36%). It seems to be more helpful for questions that require explanations and for problems that ask questions about tables or 3D models.
- (ii) **Assessing Spatial Reasoning Capabilities 5.2** Can GPT-4V perform spatial operations and understand how objects are related to each other based on human spatial reasoning tests?
 - We measure the scores obtained on three spatial reasoning tests. Across five runs for each test, GPT-4V gets an average score of 36% on the *packing test*, a 16% on the *MechE rotation test*, and a 20% on the modified *MechE rotation test*, well-below typical scores from undergraduate students in science fields.

5.1 Textbook Problems

Overview and Motivation During their engineering curriculum, students are regularly asked to solve engineering design problems that require them to interpret sketches, graphs, tables, and images to answer a related question. As such, students need to integrate their natural language processing and visual information understanding skills with their domain knowledge to solve this type of problem. These problems thus enable us to evaluate GPT-4V’s ability to capture cross-modal interactions between the question and given images. Furthermore, we would also like to evaluate GPT-4V’s ability to integrate domain-specific knowledge while answering these questions. Furthermore, we evaluate GPT-4V’s problem-solving capabilities when asked free-text, multiple-choice, numerical, and sketching questions, and its ability to process diagrams, photographs, graphs, 3D models, and tables.

Methodology To evaluate our model, we gathered questions from two undergraduate engineering design classes publicly available under CC-BY-NC-SA on MIT OpenCourseWare [Daniel Frey and David Gossard, 2009, Jung-Hoon Chun and Sang-Gook Kim, 2004]. The class materials include problem sets and exams. All class materials come with model solutions, which we use as ground truth. To ensure that we are evaluating GPT-4V’s multimodal capabilities, we select questions that reference one or more pictures in the question prompt. We ignore questions that require the student to annotate a given image since GPT-4V cannot generate images. However, if a question asks a student to generate sketches, we prompt GPT-4V to generate code to create the sketch. Indeed, sketches can be parameterized by a coding language whereas images cannot, thus we would like to test GPT-4V’s ability to generate code.

To ensure independence, we reset GPT-4V’s context window for each question except for multi-part questions. For multi-part questions, we prompt GPT-4V in order of each part. For multi-part questions that have multiple images, we supplement each sub-question with only the images required to solve that particular question in order to avoid confusing GPT-4V with superfluous information. For example, consider a multi-part question has two images X and Y , and part (a) only requires X to solve, part (b) requires Y to solve and part (c) requires both X and Y to solve, we would supplement question (a) with X , question (b) with Y and question (c) with both X and Y .

We evaluate GPT-4V’s correctness based on a binary scale, where we give it a 1 if it gets a question fully correct and a 0 otherwise. Being “fully correct” means outputting an answer that is semantically similar to the ground truth for free-text questions. For questions involving calculations, we check if GPT-4V outputs the correct numerical answer for a question and check if the intermediate steps can reasonably lead to the correct solution. For example, if GPT-4V has the correct methodology but makes a calculation error at the end, we give it a 0 since its output does not match the ground truth answer. For multi-part questions, we award a point for each correct part. We group the errors into three categories:

- **Reasoning:** The provided explanation or calculation is incorrect.
- **Inference:** The model failed to extract information from the image correctly.
- **Imprecise:** The provided answer is too vague or only explains the task without executing it.

Counting each part of a multi-part question as a distinct question, we evaluated GPT-4V’s ability to answer over 44 different engineering questions. If each multi-part question was counted only once, we evaluated 21 questions in total. The questions were repeated three times to account for the variability of the model and, overall, a question was considered correctly answered if at least two repeats were correct.

Scores on Textbook Problems We observe that GPT-4V can answer 16 of these 44 questions correctly, giving an average 36% accuracy. The accuracies when considering the type of image in the question and the format are provided in Table 14 and 15, respectively. An overview of all the repeats and questions is provided in Table 16. Relatively to the type of image, GPT-4V answered correctly most questions involving 3D models and tables (63% and 67% respectively) but had a lower success rate for photographs (33%), diagrams (29%) and graphs (0%). In terms of question format, GPT-4V performed slightly better on free-text questions (44%), than on any other format. Overall, we observe that GPT-4V makes mostly reasoning errors (20), followed by imprecise answers (5) and inference errors (3).

Table 14: Summary of GPT-4V’s score on textbook problems grouped by type of image.

	Photograph	Diagram	Graph	3D model	Table	Overall
Correct	1	8	0	5	2	16
Total	3	28	2	8	3	44
Average	33%	29%	0%	63%	67%	36%

Table 15: Summary of GPT-4V’s score on textbook problems grouped by question format.

	Free text	MCQ	Numerical	Draw	Overall
Correct	7	0	8	1	15
Total	16	1	24	3	44
Accuracy	44%	0%	33%	33%	36%

Table 16: Detailed list of questions including image type and question format and answers by GPT-4V for each trial, along with the type of error.

	Image Type	Format	#1 / Error Type	#2 / Error Type	#3 / Error Type	Overall
Q1-a	Diagram	Free text	×/ Imprecise	×/ Imprecise	×/ Imprecise	×
Q1-b	Diagram	Free text	×/ Imprecise	×/ Imprecise	×/ Imprecise	×
Q1-c	Diagram	Free text	×/ Imprecise	×/ Imprecise	×/ Imprecise	×
Q2	Photograph	MCQ	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q3-a	3D-model	Free text	✓	✓	✓	✓
Q3-b	3D-model	Free text	✓	✓	✓	✓
Q4-a	Diagram	Numerical	✓	×/ Reasoning	×/ Reasoning	×
Q4-b	Diagram	Numerical	×/ Inference	×/ Inference	×/ Inference	×
Q5	Diagram	Draw	×/ Imprecise	×/ Imprecise	×/ Imprecise	×
Q6	Photograph	Numerical	✓	✓	✓	✓
Q7	Photograph	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q8-a	3D-model	Free text	✓	×/ Reasoning	×/ Reasoning	×
Q8-b	3D-model	Free text	×/ Reasoning	×/ Reasoning	✓	×
Q8-c	3D-model	Draw	×/ Imprecise	×/ Imprecise	×/ Imprecise	×
Q9-a	Table	Numerical	✓	×/ Inference	✓	✓
Q9-b	Table	Numerical	×/ Reasoning	✓	✓	✓
Q9-c	Table	Numerical	×/ Reasoning	×/ Reasoning	✓	×
Q10	3D-Model	Free text	✓	✓	✓	✓
Q11	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q12-a	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q12-b	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q12-c	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q12-d	Diagram	Free text	✓	✓	✓	✓
Q13	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q14-a	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q14-b	Diagram	Free text	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q14-c	Diagram	Free text	✓	✓	✓	✓
Q15-a	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q15-b	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q16-a	Diagram	Draw	✓	✓	✓	✓
Q16-b	Diagram	Free text	✓	✓	✓	✓
Q16-c	Graph	Free text	✓	×/ Inference	×/ Inference	×
Q16-d	Diagram	Numerical	×/ Reasoning	✓	✓	✓
Q16-e	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q17-a	Diagram	Numerical	✓	✓	✓	✓
Q17-b	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q17-c	Diagram	Numerical	✓	✓	✓	✓
Q18-a	Diagram	Numerical	×/ Reasoning	✓	×/ Reasoning	×
Q18-b	Diagram	Numerical	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q19-a	Diagram	Free text	×/ Reasoning	×/ Reasoning	×/ Reasoning	×
Q20-a	Diagram	Free text	✓	✓	✓	✓
Q20-b	Diagram	Free text	×/ Inference	×/ Inference	×/ Inference	×
Q21-a	3D-Model	Numerical	✓	✓	✓	✓
Q21-b	3D-Model	Numerical	✓	✓	✓	✓
						16 (36%)

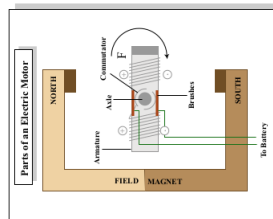
Selected Questions and Answers In the following, we reproduce selected questions and answers to illustrate the type of questions, as well as the type of errors in the answers.

First, we look at an example of imprecise answers. Context 5.1.1 shows the question and answer to Q1 (a–c). Although GPT-4V can describe relationships between stall torque, no-load speed, and maximum power, it fails to provide the exact proportions by which the relationships increase or decrease. For Q1, the expected solution is that by doubling the number of windings, the stall torque doubles, the no-load speed is cut in half and the maximum power stays constant. Noteworthy, the provided answer also contains additional explanations that were not asked for.

Context 5.1.1: Q1 - Repeat 1 - Relating proportional changes

Prompt 1:

A designer proposes to change the electric motor by increasing the number of windings of wire around the armature significantly (e.g., by doubling it). All other parameters of the motor are preserved as they were (radius of the armature, strength of the magnetic field, and so on). Describe the influence (if any) on: a) stall torque, b) no load speed, and c) maximum power.



GPT-4V Answer:

The diagram you provided shows a schematic representation of the main parts of a simple electric motor. When a designer proposes to increase the number of windings around the armature of an electric motor, while keeping all other parameters constant, the following effects can be generally anticipated:

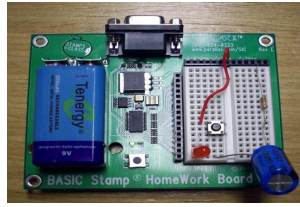
- Stall torque:** Increasing the number of windings around the armature will increase the magnetic field strength generated by the armature when current flows through it. By increasing the interaction between the magnetic fields of the armature and the field magnet, the stall torque should increase. Stall torque is proportional to the strength of the magnetic interaction, and doubling the number of windings will increase the force that the motor can exert when it is not rotating.
- No load speed:** The no load speed, the speed at which the motor runs when no load is applied, might decrease as a result of the increased windings. This is because more windings create more resistance and inductance in the coil, which can limit the current and reduce the motor's efficiency at high speeds. The back-EMF (electromotive force) generated by the motor (which opposes the applied voltage) increases with speed. With more windings, more back-EMF is generated for a given speed, thereby potentially reducing the no load speed.
- Maximum power:** The maximum power is the product of torque and speed at which the power is maximum. Since there may be an increase in torque due to the increased number of windings, and a potential decrease in the maximum speed, the overall effect on maximum power is not straightforward and would depend on the exact parameters of the motor and the specifics of how the speed-torque curve is affected. However, typically, with more windings and thus higher torque at lower speeds, the motor may operate more efficiently at lower speeds and could potentially have higher maximum power.

It's important to note that in practice, these predictions must be tested experimentally since the actual performance can be influenced by various secondary factors not taken into account in this simple analysis, such as thermal effects, winding distribution, core saturation, and brush contact losses, among others. Additionally, the physical space available in the armature and the increased resistance due to more windings (leading to energy loss and heat) must also be considered.

Next, we look at Q2, which is the only multiple-choice question, see Context 5.1.2. The correct answer to this question is (d) since the capacitor is already charged up to the supply voltage and cannot unload through the LED. GPT-4V seems unable to understand the circuit based on the photograph and thus, bases its responses solely on the provided text. While it understood that the LED and the capacitor were in series, it got some basic physics concepts incorrect.

Context 5.1.2: Q2 - Repeat 1 - Reasoning on an electric circuit based on a photograph

Prompt 1:



The components shown here (resistor, capacitor, LED, and normally-open-button) are connected in series. The button is pressed and the LED lights up and then slowly dims over the course of 5 seconds. The button is released and 5 more seconds pass. The button is pressed again. Which statement best describes what happens:

- (a) The LED lights up and then slowly dims over the course of 5 seconds
- (b) The LED starts dim then slowly brightens over the course of 5 seconds
- (c) The LED lights up steadily
- (d) LED does not light

GPT-4V Answer:

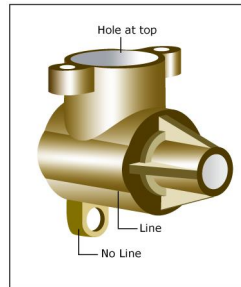
The image you provided shows an electronic setup on a breadboard with a 9V battery connected to a BASIC Stamp Homework Board. The board is labeled and has a microcontroller and other components, but the specific components you mentioned (resistor, capacitor, LED, and normally-open-button) are not directly visible in the picture you provided. They could be on the breadboard area or connected to the microcontroller pins, but they are not clearly identifiable.

Regardless of the image, the scenario you described suggests a basic electronics circuit where the LED's behavior is influenced by a capacitor in series with it. When the button is pressed, the capacitor starts charging through the resistor, and the LED lights up. As the capacitor charges, the voltage across the LED drops, causing it to dim over time.

Based on the information provided in your description, assuming that the capacitor had enough time to discharge when the button was released, when the button is pressed again, the capacitor will start charging again from a discharged or partially discharged state. Thus, the LED will likely light up and then slowly dim again as the capacitor charges up. Hence, the correct answer to your question would be:

- a) The LED lights up and then slowly dims over the course of 5 seconds.

GPT-4V performs well on tasks that require high-level reasoning and explanation. In Context 5.1.3, it can match the natural language description (circular feature), locate the feature, and identify the name of the feature. Furthermore, when asked to explain whether a line should or should not exist, it can identify the relevant regions and combine domain knowledge to solve the problem.

Context 5.1.3: Q3 - Repeat 1 - Identification and explanation*Prompt 1:*

- a) What term would you use to describe the circular feature extending from the left side of the part?
- b) Explain briefly why it is appropriate to show a line on this drawing at the location indicated above by “Line” and not to show a line at the location indicated above by “No Line”.

GPT-4V Answer:

- a) The term to describe the circular feature extending from the left side of the part is called a “boss.” A boss is a protruding feature in a casting or a part designed to serve as a mounting point or to add strength.
- b) The line shown on the drawing where it is labeled “Line” indicates the presence of an edge or a boundary that is visible from the perspective of the viewer, suggesting a change in the contour that is apparent at that angle. However, there is no line shown at the position marked “No Line” because from the viewer’s perspective, there is no visible edge or change in contour at that location; it represents a smooth transition on the surface of the object. Showing a line at that spot could incorrectly imply an edge or feature that doesn’t exist in the actual part, which could lead to confusion or misinterpretation of the part design.

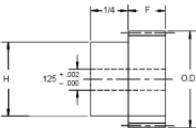
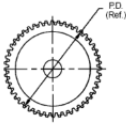
Finally, we look at a question that requires extracting values from a table and performing calculations using them. While arguments can be made to input tabular data as text, tables are often inconvenient to input and are more convenient for users to input as images. In this particular answer (repeat 3), GPT-4V is able to correctly extract the values and calculate the center distance, the torque, and the reaction force. It is worth keeping in mind, see Table 16 that the two other repeats were not as successful, showing how challenging such a task is.

Context 5.1.4: Q9 - Repeat 3 - Calculate with information from a table

Prompt 1: (Tabular)

Spur Gears

24, 32, 48, and 64 Pitch 1/8" Bore AGMA Quality 4
Cold Rolled Steel and Brass 20° Pressure Angle



COLD ROLLED STEEL C12.14 OR C12.15 WITH SELENIUM	BRASS ALLOY 360	STOCK NUMBER	NO OF TEETH	PITCH DIA.	OUTSIDE DIA.	H	F
24 PITCH (.1309)							
PX24S-8	PX24B-8	8	333	.416	.208	1/4	
PX24S-9	PX24B-9	9	375	.458	.250		
PX24S-10	PX24B-10	10	417	.500	.251		
PX24S-12	PX24B-12	12	500	.583	.375		
PX24S-16	PX24B-16	16	666	.750	.542		
PX24S-18	PX24B-18	18	750	.833	.625	1/4	
—	PX24B-22	22	916	1.000	.792		
32 PITCH (.0981)							
PX32S-10	PX32B-10	10	312	.375	.218	1/4	
PX32S-11	PX32B-11	11	344	.409	.250		
PX32S-12	PX32B-12	12	375	.437	.281		
PX32S-14	PX32B-14	14	438	.500	.343		
PX32S-15	PX32B-15	15	469	.531	.375		
PX32S-16	PX32B-16	16	500	.562	.406	1/8	
PX32S-18	PX32B-18	18	563	.625	.468		
PX32S-20	PX32B-20	20	625	.688	.532		
—	PX32B-24	24	750	.813	.656		
—	—	—	—	—	—		
48 PITCH (.0654)							
PX48S-14	PX48B-14	14	292	.333	.229	1/8	
PX48S-15	PX48B-15	15	312	.353	.250		
PX48S-16	PX48B-16	16	333	.375	.271		
PX48S-18	PX48B-18	18	375	.417	.312		
PX48S-24	PX48B-24	24	500	.542	.437		
PX48S-30	PX48B-30	30	666	.708	.654	1/8	
—	PX48B-36	36	750	.792	.687		
—	—	—	—	—	—		
—	—	—	—	—	—		
—	—	—	—	—	—		
64 PITCH (.0490)							
PX64S-15	PX64B-15	15	234	.285	.187	1/8	
PX64S-16	PX64B-16	16	250	.281	.203		
PX64S-18	PX64B-18	18	281	.312	.234		
—	PX64B-24	24	375	.406	.328		
—	PX64B-40	40	625	.656	.578		
—	PX64B-45	45	750	.781	.703		

Berg Manufacturing "The Mark of Quality"

1-800-232-BERG

Berg Manufacturing "The Mark of Quality"

1-800-232-BERG

If gear PX32B-10 and PX32B-20 are mated together in a gear train, how far apart should the centers of their shafts be placed? If gear PX32B-10 and PX32B-20 are mated together in a gear train and a torque of 2 ft lbs is applied to PX32B-10, what is the torque on PX32B-20?

If gear PX32B-10 and PX32B-20 are mated together in a gear train and a torque of 2 ft lbs is applied to PX32B-10, what is the direction and magnitude of the reaction force at the shaft where PX32B-20 is mounted? State any assumptions needed to arrive at your answer.

GPT-4V Answer:

To determine the correct center distance for mating two gears, you generally sum the pitch radii of the two gears. According to the image provided, we have two gears—PX32B-10 and PX32B-20—both with a 32 pitch (which is the number of teeth per inch of the diameter).

From the chart, the pitch diameter (PD) for PX32B-10 (which has 10 teeth) is 0.312 inches, and the PD for PX32B-20 (which has 20 teeth) is 0.625 inches. To find the center distance, we add the two pitch radii:

$$\begin{aligned}
 \text{Center distance} &= (\text{Pitch Radius of PX32B-10}) + (\text{Pitch Radius of PX32B-20}) \\
 &= (\text{PD of PX32B-10}/2) + (\text{PD of PX32B-20}/2) \\
 &= (0.312/2) + (0.625/2) \\
 &= 0.156 + 0.3125 \\
 &= 0.4685 \text{ inches}
 \end{aligned}$$

Regarding the torque, it is conserved because power must be conserved, and assuming there is no loss due to friction or other factors, the power transmitted by one gear will be equal to the power transmitted to the other, following the equation:

$$\text{Power} = \text{Torque} \times \text{Angular Velocity}$$

Cont.—Q9 - Repeat 3 - Calculate with information from a table**Cont. GPT-4V Answer:**

In a gear train, if one gear is spinning twice as fast as the other, it will have half the torque, since the product of torque and speed (angular velocity) is constant (if we disregard efficiency losses). With PX32B-10 having 10 teeth and PX32B-20 having 20 teeth, PX32B-20 will rotate at half the speed of PX32B-10, due to the gear ratio of 2:1. So if a torque of 2 ft lbs is applied to PX32B-10, the torque on PX32B-20 will be double, given by:

$$\begin{aligned}\text{Torque on PX32B-10} \times \text{Gear Ratio} &= \text{Torque on PX32B-20} \\ 2 \text{ ft lbs} \times (20 \text{ teeth}/10 \text{ teeth}) &= \text{Torque on PX32B-20} \\ \text{Torque on PX32B-20} &= 4 \text{ ft lbs}\end{aligned}$$

As for the direction and magnitude of the reaction force at the shaft where PX32B-20 is mounted, the force can be calculated using the relationship between torque (T), force (F), and radius (r):

$$\text{Torque (T)} = \text{Force (F)} \times \text{Radius (r)}$$

To find the force, we would rearrange this equation:

$$F = T/r$$

Assuming a pure torque application with no other forces at play (like axial or radial loads) and using the pitch radius for PX32B-20, which is half of the pitch diameter ($0.625 / 2 = 0.3125$ inches):

$$F = 4 \text{ ft lbs} / 0.3125 \text{ inches}$$

We need to convert the radius to feet to match the units of torque:

$$0.3125 \text{ inches} \times (1 \text{ foot}/12 \text{ inches}) = 0.02604 \text{ feet}$$

Now let's calculate the force:

$$F = 4 \text{ ft lbs} / 0.02604 \text{ feet} \approx 153.61 \text{ lbs}$$

This is the magnitude of the reaction force at the shaft of PX32B-20. The direction of this force will be radially inward toward the center of the gear, opposite to the force that PX32B-10 is applying to PX32B-20, because for every action there is an equal and opposite reaction, according to Newton's third law of motion.

Discussion

As previously mentioned, GPT-4V makes three types of mistakes: reasoning, image misinterpretation, and imprecision. We go into detail below.

Reasoning GPT-4V can sometimes provide incorrect answers as a result of reasoning errors. This was especially apparent with multi-step reasoning tasks. When asked what happens to a system after a series of actions are performed, such as the one given in Context 5.1.2, GPT-4V often makes a mistake in reasoning, such as hallucinating a fact, and derails its chain of thought. Furthermore, When provided a question to compute a numerical answer, GPT-4V can sometimes have the correct methodology for arriving at the correct answer but provides an incorrect answer due to an incorrect numerical approximation. For example, when asked to compute $e^{0.3\pi}$, GPT-4V instead used e^π as an approximation and arrived at an incorrect answer even though its previous steps were correct. This issue is not difficult to alleviate, however, as previous papers have shown that leveraging tools such as calculators can enable GPT-4V to perform numerical reasoning tasks better.

Image misinterpretation GPT-4V can have trouble understanding and inferring information in images. In Context 5.1.2, GPT-4V fails to interpret the circuit shown as a photograph. As a result of this misinterpretation, GPT-4V cannot leverage the information present in it and only answers based on the provided text.

Imprecision While GPT-4V has reasonable success in providing qualitative answers, it sometimes fails to do so as its answers are too vague or do not capture the main idea of a question. In a question that asked about potential errors in an injection molding experiment, GPT-4V provided a long list of potential issues, but not issues that were specific to the

question. GPT-4V also had issues with relating exact numerical relationships between different variables as seen from Context 5.1.1.

5.2 Spatial Reasoning

Spatial reasoning is the ability of humans to perform mental spatial operations: rotations, translation, projection, and orientation. Spatial reasoning is at play when humans read maps, navigate their homes at night without light, or solve pretty much any problem in the fields of science, technology, engineering, and mathematics (STEM) [Maeda and Yoon, 2013]. Spatial reasoning skills are considered essential skills for understanding graphs, diagrams, plots, 3D objects, and representations. Indeed, multiple studies have found that spatial abilities are a good predictor of academic success [Shea et al., 2001, Berkowitz and Stern, 2018]. Consequently, spatial reasoning skills have been well studied in humans, and many standardized tests exist, e.g., The Revised Purdue Spatial Visualization Test: Visualization of Rotations (PSVT:R) [Yoon, 2011], the Mental Cutting Test “Schnitte” [Quaiser-Pohl, 2003], or the Novel Spatial Ability Tests [Berkowitz et al., 2021].

Following some of our observations on the apparent struggles of GPT-4V regarding spatial understanding, we specifically tested its spatial abilities in order to provide additional insights. Spatial reasoning tests are also good candidates to evaluate vision language models since they focus on inherently visual tasks and are often not publicly available, to maintain their validity. They are thus unlikely to be part of the training data.

Methods We assessed GPT-4V’s spatial reasoning skills using the openly accessible *packing test* (part of the *Novel Spatial Ability Tests* [Berkowitz et al., 2021]), and the *MechE Rotation Test* [Picard, 2023]. While the first one is openly accessible, the latter is released publicly for the first time in parallel to this work.

The *MechE Rotation Test* follows the general principles of the PSVT:R, but uses objects with features typically seen on mechanical parts. It measures the ability of participants to visualize one or two rotations applied to a reference object and apply them to another object. For each question, five possible configurations of the object are shown and the participants select the correct one. The test is composed of an example—for which the correct answer is given to the participant—followed by ten questions of increasing difficulty. The *packing test* requires participants to evaluate if shapes can be composed of or decomposed into smaller sub-shapes. The *packing test* is split into two parts: packing and unpacking. In the first part, participants have to choose among four options which set of sub-shapes can be packed together to form a larger shape. In the second part, participants do the opposite and select among four large shapes, which can be decomposed into the provided smaller shapes. The example questions for these tests are shown in Figure 11.

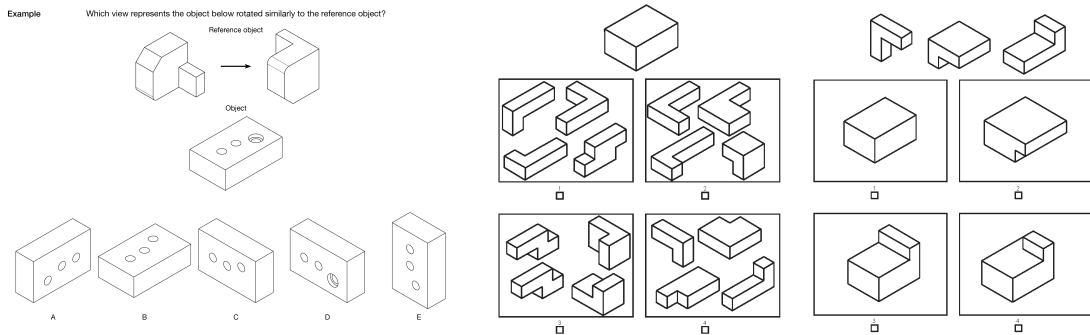


Figure 11: Example questions from the considered spatial reasoning tests.

In this work, we have GPT-4V take the tests as they are given to human participants. Each questionnaire is passed in a single context, sequentially going through the examples and the questions, and providing the instructions and images. To account for some stochasticity, each questionnaire is repeated five times. In addition, and inspired by Yang et al. [2023a], we evaluate if adding visual marks—reference coordinates and coloring faces—improved the performance of the model on the *MechE rotation test*. The runs using the original test are referred to as *Run H*, while the ones with the marks are called *Run P*.

The full set of prompts is made available as a benchmark for any future vision language model.

Scores on The Spatial Reasoning Tests The answers of GPT-4V for the *packing test* and the *MechE rotation test* are provided in Table 17 and 17, respectively. Starting with the *packing test*, GPT-4V obtains an average score across five runs of 36%, slightly higher than the expected average score if answering at random (25%). Interestingly, all five questions that have been answered correctly have been done so by at least two runs, further suggesting that GPT-4V is

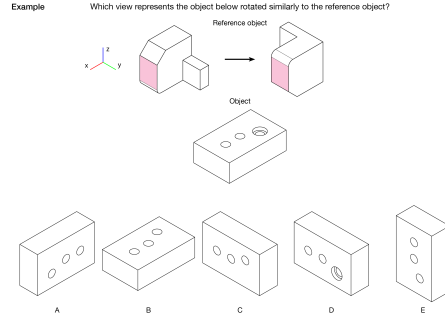


Figure 12: Example of the MechE rotation test with the additional visual prompts to support the model.

Table 17: Answers and scores for the Packing Test [Berkowitz et al., 2021]. Correct answers are in bold. Each run was conducted within the same context.

(Correct)	Run 1	Run 2	Run 3	Run 4	Run 5
Part 1 Q1 (4)	3	2	3	3	3
Part 1 Q2 (3)	4	1	4	4	4
Part 1 Q3 (3)	3	3	3	3	3
Part 1 Q4 (2)	2	1	2	2	2
Part 1 Q5 (1)	3	4	4	4	3
Part 2 Q1 (3)	3	3	2	3	3
Part 2 Q2 (3)	4	1	1	4	1
Part 2 Q3 (2)	2	4	3	2	2
Part 2 Q4 (2)	1	3	1	3	1
Part 2 Q5 (3)	3	1	4	4	3
Scores	50%	20%	20%	40%	50%
Average					36%

not answering at random. In comparison to humans, however, it remains significantly lower than the average score of undergraduate (66%) and graduate (73%) students reported by Berkowitz et al. [2021]. For the *MechE rotation test*, the average scores (16% and 20%) are lower and closer to the expected score for random answering (20%). While slightly higher, it is unclear if the visual prompting supports GPT-4V. While no human results have been published for this test, average scores between 60% and 70% are expected based on internal tests and by comparison to the revised PSVT:R test.

Table 18: Answers and scores to the MechE Rotation Test. Correct answers are in bold. Each run was conducted within the same context.

(Correct)	Run H1	Run H2	Run H3	Run H4	Run H5	Run P1	Run P2	Run P3	Run P4	Run P5
Q1 (D)	C	C	C	D	D	D	D	C	C	B
Q2 (A)	D	B	A	D	C	D	B	D	B	D
Q3 (C)	C	C	C	E	B	C	D	B	C	C
Q4 (E)	C	D	A	A	C	C	C	D	B	D
Q5 (C)	C	B	A	B	B	D	B	B	C	C
Q6 (B)	E	D	C	D	C	D	E	E	D	E
Q7 (C)	E	E	D	C	D	E	C	C	B	D
Q8 (A)	B	C	C	B	E	C	D	B	C	C
Q9 (E)	D	B	A	C	B	C	A	C	B	E
Q10 (A)	C	C	C	B	D	B	E	B	C	B
Scores	20%	10%	20%	20%	10%	20%	20%	10%	20%	30%
Average					16%					20%

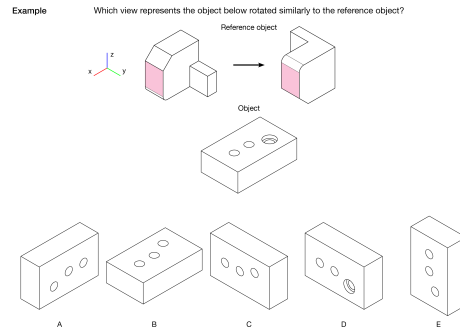
To gain more insights, run P1 is reproduced in Context 5.2.1. GPT-4V’s answers seem to indicate that the nature of the test and the task is well understood. However, while the answer is correct, the reasoning is incorrect. Based on the provided coordinate system, the reference object is rotated by 90° around the X-axis, and not the Z-axis as stated by GPT-4V. This type of behavior has already been reported regarding numerical reasoning [Stechly et al., 2023]. As such, it seems like additional visual or textual instructions are needed to properly root the model within a spatial system.

Context 5.2.1: Run P1

Prompt 1:

You will be shown a set of questions evaluating your spatial reasoning abilities, specifically your ability to mentally rotate 3D objects. Each question will demonstrate a set of one or two rotations applied to a reference object (at the top of the image, original position on the left, rotated configuration on the right). Study how the reference object has been rotated and then apply the same rotations to the object in the middle and select from among the five options given at the bottom which one represents the rotated configuration.

Here is an example question. What is the correct configuration for this example?



In the example question, the reference object is rotated 180° . Answers A, C, D, and E are wrong since they cannot be obtained by a 180° rotation. Answer B is the correct answer.

You will now see 10 questions and will be asked to identify which configuration is the correct one. Always start by stating how the reference object is rotated and then answer which configuration among the choices A, B, C, D, or E represents the rotated configuration.

GPT-4V Answer:

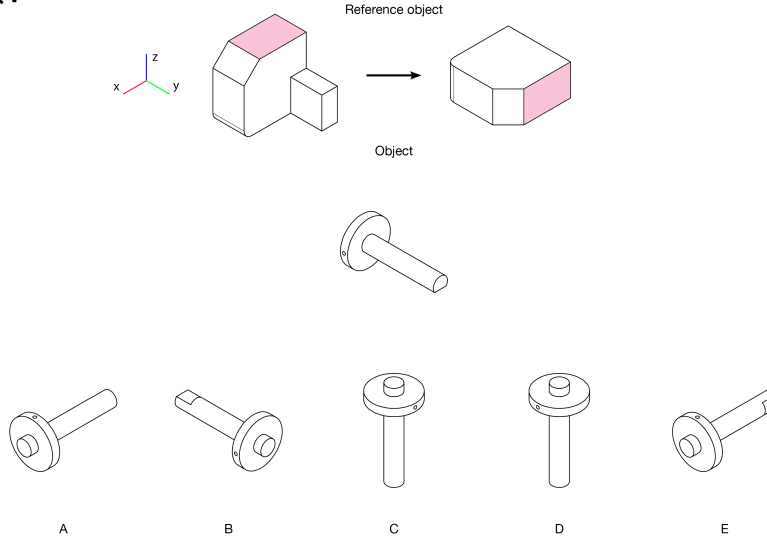
Alright, please provide the first question, and I'll help you identify the correct configuration based on the rotation of the reference object.

Cont.—Run P1

Prompt 2:

Q1

Which view represents the object below rotated similarly to the reference object?

**GPT-4V Answer:**

For Q1: The reference object has been rotated 90° around the Z-axis.
Considering the options:

- A and E are simply showing the original position, so they are incorrect.
- B has been rotated 90° around the Y-axis, so it's incorrect.
- C is also incorrect as it has been rotated 90° around the X-axis.
- D is the correct representation of the object when rotated 90° around the Z-axis.

Answer: D.

Discussion Overall, our evaluation of the spatial abilities of GPT-4V using standardized (human) tests tends to suggest that, compared to humans, GPT-4V has some, although limited, spatial reasoning capabilities. Indeed, while these visualization tasks are hard and constructed to be somewhat deceptive, most untrained undergraduate students in science and technical fields answer at least half of the questions correctly [Yoon, 2011, Berkowitz et al., 2021]. Unfortunately, this seeming lack could, in part, explain GPT-4V’s limitations in performing engineering design tasks, such as CAD generation, see Section 3.2. These results also corroborate findings recently reported in the literature Wen et al. [2023].

6 Discussion

In this paper, we aimed to evaluate GPT-4V’s capabilities in several engineering design tasks ranging from conceptual design to manufacturing.

Conceptual Design We looked at design similarity analysis, sketch descriptions, and concept selection. We found that GPT-4V was able to assess design similarity with high self-consistency and low transitive violations. It was also consistent with human-generated idea maps in identifying unique sketches and groups of similar sketches. Additionally, it effectively matched design sketches to their descriptions when provided with the entire sketch (an average score of 10/10), including a handwritten description, but without the description, it often chose “None of the above,” and therefore performed worse (average score of 5.33/10). When “None of the above” was not an option, GPT-4V performed better (an average score of 7/10). This suggests a level of “caution,” so when GPT-4V has the chance to not be incorrect, it takes it. GPT-4V could generate useful and accurate text descriptions of designs even for sketches with very low drawing scores. Lastly, the model generated appropriate selection criteria but did not generate Pugh charts when only provided with design sketches. Overall, GPT-4V shows the great potential of VLMs for design sketch analysis and supporting the conceptual design stage beyond what has been identified in previous work [Siddharth et al., 2022, Stella et al., 2023].

System-Level And Detailed Design We investigated GPT-4V’s ability to use several Ashby diagrams to suggest appropriate materials, to analyze engineering drawings, to generate CAD scripts, to understand and analyze structures resulting from topology optimization (TO), and to analyze results from CFD simulations. We found that GPT-4V could correctly respond where to look for materials in Ashby diagrams, but made errors when asked to be specific. The model faced difficulty in understanding the nuances of a block-with-blind-hole engineering drawing, but it was able to extract most dimensions and assign them appropriate labels. In terms of CAD generation ability, GPT-4V had limited success in generating correct CAD scripts on the first attempt, and our iterations to fix the scripts did not improve the results. It showed a good general understanding of both TO and CFD, and could relate to specific features on the shown images. However, it struggled to identify floating material for TO and incorrectly matched theoretical information with visual inputs for CFD. For example, GPT-4V interpreted one image as showing shock waves, despite the Mach number being well below the threshold for this. Overall, the addition of vision enables GPT-4V to provide relevant explanations for common detailed design tasks, but lacks precision.

Manufacturing and Inspection In the manufacturing stage, we tested GPT-4V’s understanding of design for manufacturing (DfM) for subtractive and additive manufacturing operations. GPT-4V was, as we interpret it, cautious, and suggested that none of the additive manufacturing parts were printable, even when the parts were well within the provided guidelines. On the feature identification task for subtractive manufacturing, GPT-4V was able to identify at least one feature 12 out of 20 times, but never all of them. The provided explanations were most of the time inconsistent and confused different technical terms. Furthermore, we assessed GPT-4V’s ability to inspect images to find and identify defects. For the evaluated cases, GPT-4V tended to overly predict the presence of defects and was inconsistent in the identification of the type of defect. Even more than in the detailed design stage, design for manufacturing and inspection is all about precision, and GPT-4V overall fails to deliver reliable and consistent performance on the evaluated task.

Educational Tasks Lastly, we evaluated GPT-4V’s ability at educational tasks in engineering, namely solving textbook problems and performing a spatial reasoning assessment. Overall, GPT-4V achieves rather low scores for both types of tasks, 36%, 36%, and 18% accuracy over the textbook problems, the packing test, and the MechE rotation test respectively. It performed the best for textbook problems asking for explanations (free text answers) but struggled for numerical questions, both in the reasoning and in the numerical value extraction from the provided images. Specifically on the spatial reasoning tests, GPT-4V’s scores are indistinguishable from random answer picking and the provided explanations did not match the visual representations. Given the low scores, the textbook problems and spatial reasoning tests could become competitive benchmark to evaluate future multimodal LLMs.

Overarching Themes Across, all our experiments we noted a few overarching themes:

- Providing additional textual context usually increases the model’s understanding of the provided images—e.g., when told that the image contains a blind hole it was able to generate better CAD and its sketch description improved when given the prompt of the design task provided to the students. These observations suggest that providing detailed, context-rich descriptions can enhance AI’s performance in many engineering tasks. This can lead to more accurate and efficient design processes, especially in early stages where conceptual understanding is crucial. Practitioners should focus on developing clear, comprehensive problem descriptions to improve AI utility. Furthermore, future research can investigate how different types and levels of contextual information impact AI performance in various engineering tasks.

- GPT-4V can request information when uncertain—e.g., in Context 3.4.1 the model asks about Ma or Re, and when asked to make a Pugh chart, the model requests more information. These observations indicate that practitioners should be prepared to engage in an iterative information exchange with AI tools. For researchers, there’s a need to understand the dynamics of AI-human information exchange and how this can be optimized for better and more efficient problem-solving. Furthermore, future research can focus on the development of new AI models that proactively seek out relevant information, perhaps by integrating with databases or using external sources. Research can also focus on improving AI’s understanding of when and what additional information is needed.
- GPT-4V tends to side on the more cautious side—e.g., when asked to describe design concepts without text descriptions, the model chooses “None of the above” more frequently, similarly, GPT-4V says a part will not print or that there is a crack when none. This could reflect the alignment process followed by OpenAI to make the model “safe” [OpenAI, 2023]. Such takeaways may not apply broadly and could be different for base models such as LLaVA 1.5. These observations indicate that practitioners should be aware of AI’s conservative tendencies and incorporate human oversight to balance this. Researchers should also investigate how to fine-tune AI’s risk assessment capabilities to align with different engineering domains’ safety and reliability standards. Research can also explore the integration of AI with decision-making frameworks to balance caution with practicality. Future models could understand the risk and return trade-offs, and be more or less conservative based on potential implications.
- More general and descriptive tasks are usually better solved than specific tasks—e.g., identifying the presence of a concrete defect compared to identifying the nature of the concrete defect, and similarly, GPT-4V can explain Ashby plots well, but struggles when it comes to quantitative selection of a material. These observations indicate that practitioners should leverage AI for conceptual and general analysis phases and potentially rely on human expertise or other tools for detailed, technical work. This also highlights that there is a need to explore how AI’s capabilities in specific, detailed tasks can be enhanced. Research could also involve developing hybrid systems where AI handles broad tasks while humans manage detailed aspects.
- GPT-4V has limited spatial reasoning capabilities both on tests created to evaluate it specifically and in our experiments across domains—e.g., when generating CAD and analyzing floating material in TO. Given the importance of these skills in engineering, their lack presents a serious limitation to perform certain tasks with precision. These observations indicate that practitioners need to recognize the current limits of AI in spatially intensive tasks and be cautious when they are used. Furthermore, researchers could investigate ways to enhance AI’s spatial reasoning capabilities, possibly through advanced training techniques, larger specialized datasets, or by integrating AI with spatial reasoning tools.

Applicability to Other Vision-Language Models While this work investigates GPT-4V specifically, we proposed a set of benchmark cases that could be used to assess other current and future models and that could inform about their performance for engineering design. In particular, we provide our datasets, prompts, and specific questions, where applicable. It is worth noting that several second-generation open-source vision language models have been released recently, such as LLaVA 1.5 [Liu et al., 2023b] or Fuyu-8B [Bavishi et al., 2023]. Unfortunately, most of these models have not yet been fine-tuned and aligned to the level of GPT-4V, making them less practical out-of-the-box. As we have shown in Section 2.2.1, while LLaVA 1.5 obtains similar quantitative results to GPT-4V, the incorrectly select answers are less consistent. These models, however, can be particularly useful as base models to develop custom fine-tuned models, for example, to specifically target the needs of engineers.

Limitations of This Study The parts of this work related to the performance of GPT-4V face the same challenges and concerns raised in previous studies of LLMs [Mao et al., 2023]. Below, we highlight a few of the limitations.

Specificity of Engineering Problems: While we attempted to cover a wide range of engineering tasks, we recognize that the study still focuses on a limited range of engineering design problems, which might not represent the broad spectrum of challenges encountered in the field. This could limit the applicability of the findings to other types of engineering tasks.

Dependency on Prompt Engineering: The results might be highly sensitive to how prompts are engineered. Subtle variations in prompt structure or wording could lead to markedly different responses from the model, affecting the reliability of the evaluations.

Dataset Representativeness: The results also depend on the selection of benchmark datasets. The quality, diversity, and representativeness of the datasets used can significantly impact the model’s performance. While we created a large set of evaluation problems for vision-language models, we recognize that these datasets might not fully capture the diversity and complexity of real-world engineering scenarios. This could affect the generalizability of the results to practical engineering applications.

Black-box and evolving models: Model changes, including data leaks, and the lack of control when using the chat interface, mean that we cannot fully define the experimental environment and some of the results may be different if reevaluated. However, for the assessments, we strived to create larger benchmarks within the limitation of the chat interface and repeated the experiments to obtain a better sample of the model’s performance.

Impact of Model Updates: Since AI models are frequently updated, the results of the study might quickly become outdated, limiting their long-term relevance. While the release of new vision-language models will enable new capabilities, we believe that this study provides a lot of value by demonstrating the tasks that future models should be evaluated on as well as providing these tasks in our datasets. We release all our quantitative datasets to measure how much future models improve for different engineering problems.

Human-AI Interaction: An important part of the engineering design process involves how humans interact with design. In this study, we did not test the capabilities of how human designers may interact with Vision-Language Models and how this interaction influences the problem-solving process is crucial, as human biases, trust, and interpretations can affect the outcomes [Zhang et al., 2023].

In conclusion, while this study offers valuable insights into the capabilities of GPT-4V in addressing engineering design problems, it is essential to recognize these limitations as an integral part of our findings. They highlight the areas needing further exploration and remind us of the cautious approach required when generalizing AI capabilities to broader real-world applications. Our research is a step in an ongoing journey, contributing to the evolving dialogue on the role and effectiveness of AI in complex, multifaceted fields like engineering design.

7 Conclusion and Future Work

The first avenue for future research following this study involves expanding the scope and depth of the engineering problems evaluated. This can be achieved by incorporating a wider variety of engineering challenges, particularly from domains that were less represented in our initial study. We believe that industry can play a crucial role in this step, by providing representative problems for different types of engineering design tasks faced by them. Such diversity in problem selection will provide a more comprehensive understanding of VLM’s capabilities across the engineering design process. Alongside this, there is a need for enhanced dataset curation. Developing more robust datasets that closely mirror complex, real-world engineering scenarios can significantly improve the model’s evaluation. These datasets should capture the multifaceted and multidimensional nature of engineering tasks, allowing for a more nuanced assessment of multimodal LLMs’ applicability and effectiveness. An effort should also be made to avoid publicly available datasets, to limit challenges with evaluation data leaking into future model training.

Another critical area of future work lies in the realm of human-AI collaboration. It is imperative to study how engineers interact with LLMs in real-world design scenarios. Such studies can shed light on practical utility, user trust, and the integration of AI into engineering workflows. This includes understanding how engineers’ biases and decision-making processes interact with AI-generated solutions. Additionally, conducting longitudinal studies to monitor the impact of model evolution over time on its performance in engineering tasks will be highly beneficial. Given the rapid developments in AI, understanding how updates and changes affect its applicability and effectiveness is crucial. This will help in keeping the AI applications in engineering up-to-date and relevant, ensuring that they continue to meet the evolving demands of the field.

Acknowledgment

CP acknowledges the Swiss National Science Foundation for its financial support (grant P500PT_206937). KME and ACD acknowledge the National Science Foundation Graduate Research Fellowship for its financial support.

Contributions

CP co-led the project and developed the Engineering Education Tasks section. Prepared the introduction and discussion with KME.

KME co-led the project and developed the whole Conceptual Design section. Prepared the introduction and discussion with CP.

ACD developed the System-level and Detailed Design section, focusing on Material Selection, CAD Generation, and Engineering Drawings. Developed the Manufacturing and Inspection section, focusing on design for additive manufacturing and post-manufacturing inspection.

BM developed the Engineering Education Tasks section, focusing on the Textbook Problems.

GG developed the System-level and Detailed Design section, focusing on Topology Optimization and Fluid Dynamics Simulation.

MFA developed the Manufacturing and Testing section, focusing on the design for subtractive manufacturing.

FA provided supervision and guidance across all project sections. Assisted in manuscript preparation and editing.

References

- Sameer Agarwal, Josh Wills, Lawrence Cayton, Gert Lanckriet, David Kriegman, and Serge Belongie. Generalized non-metric multidimensional scaling. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 2 of *Proceedings of Machine Learning Research*, pages 11–18, San Juan, Puerto Rico, 21–24 Mar 2007. PMLR. URL <https://proceedings.mlr.press/v2/agarwal07a.html>.
- Faez Ahmed, Sharath Kumar Ramachandran, Mark Fuge, Samuel Hunter, and Scarlett Miller. Interpreting Idea Maps: Pairwise Comparisons Reveal What Makes Ideas Novel. *Journal of Mechanical Design*, 141(2):021102, 12 2018. doi:[10.1115/1.4041856](https://doi.org/10.1115/1.4041856).
- Teresa M Amabile. Social psychology of creativity: A consensual assessment technique. *Journal of personality and social psychology*, 43(5):997, 1982. doi:[10.1037/0022-3514.43.5.997](https://doi.org/10.1037/0022-3514.43.5.997).
- Konstantine Arkoudas. GPT-4 can’t reason, 2023. URL <https://arxiv.org/abs/2308.03762>.
- Michael F. Ashby. *Materials Selection in Mechanical Design*. Elsevier Science & Technology Books, 2016. ISBN 9780081005996.
- Mohsen Attaran. The rise of 3-D printing: The advantages of additive manufacturing over traditional manufacturing. *Business horizons*, 60(5):677–688, 2017. doi:[10.1016/j.bushor.2017.05.011](https://doi.org/10.1016/j.bushor.2017.05.011).
- John Baer and James C Kaufman. Assessing creativity with the consensual assessment technique. In *The Palgrave Handbook of Social Creativity Research*, pages 27–37. Springer, 2019. doi:[10.1007/978-3-319-95498-1_3](https://doi.org/10.1007/978-3-319-95498-1_3).
- Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşlılar. Introducing our multimodal models, 2023. URL <https://www.adept.ai/blog/fuyu-8b>.
- Martin P Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193–202, December 1989. doi:[10.1007/BF01650949](https://doi.org/10.1007/BF01650949).
- Martin Philip Bendsøe and Noboru Kikuchi. Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988. ISSN 0045-7825. doi:[10.1016/0045-7825\(88\)90086-2](https://doi.org/10.1016/0045-7825(88)90086-2).
- Michal Berkowitz and Elsbeth Stern. Which Cognitive Abilities Make the Difference? Predicting Academic Achievements in Advanced STEM Studies. *Journal of Intelligence*, 6(4):48, December 2018. doi:[10.3390/jintelligence6040048](https://doi.org/10.3390/jintelligence6040048).
- Michal Berkowitz, Andri Gerber, Christian M. Thurn, Beatrix Emo, Christoph Hoelscher, and Elsbeth Stern. Spatial Abilities for Architecture: Cross Sectional and Longitudinal Assessment With Novel and Existing Spatial Ability Tests. *Frontiers in Psychology*, 11:4096, 2021. doi:[10.3389/fpsyg.2020.609363](https://doi.org/10.3389/fpsyg.2020.609363).
- Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.
- Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual review of fluid mechanics*, 52(1):477–508, 2020. doi:[10.1146/annurev-fluid-010719-060214](https://doi.org/10.1146/annurev-fluid-010719-060214).
- Cari Bryant, Robert Stone, Daniel Mcadams, Tolga Kurtoglu, and Matthew Campbell. Concept generation from the functional basis of design. *Proceedings ICED 05, the 15th International Conference on Engineering Design*, 01 2005.
- Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of Artificial General Intelligence: Early experiments with GPT-4, April 2023. URL <https://arxiv.org/abs/2303.12712>.
- Hannah D Budinoff and Sara McMains. Will it print: a manufacturability toolbox for 3D printing. *International Journal on Interactive Design and Manufacturing*, 15(4):613–630, 2021. doi:[10.1007/s12008-021-00786-w](https://doi.org/10.1007/s12008-021-00786-w).
- Hannah Dawes Budinoff. *Geometric manufacturability analysis for additive manufacturing*. PhD thesis, University of California, Berkeley, 2019.

- Markus J Buehler. MeLM, a generative pretrained language modeling framework that solves forward and inverse mechanics problems. *Journal of the Mechanics and Physics of Solids*, 181:105454, 2023. doi:[10.1016/j.jmps.2023.105454](https://doi.org/10.1016/j.jmps.2023.105454).
- Weijuan Cao, Trevor Robinson, Yang Hua, Flavien Boussuge, Andrew R. Colligan, and Wanbin Pan. Graph representation of 3D CAD models for machining feature recognition with deep learning. volume Volume 11A: 46th Design Automation Conference (DAC) of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, August 2020. doi:[10.1115/DETC2020-22355](https://doi.org/10.1115/DETC2020-22355).
- J. Corbett and J.R. Crookall. Design for economic manufacture. *CIRP Annals*, 35(1):93–97, 1986. doi:[10.1016/S0007-8506\(07\)61846-0](https://doi.org/10.1016/S0007-8506(07)61846-0).
- Steve Cranford. Getting DEEP with machine learning. *Matter*, 6(10):3113–3116, 2023. doi:[10.1016/j.matt.2023.07.021](https://doi.org/10.1016/j.matt.2023.07.021).
- Genevieve M Cseh and Karl K Jeffries. A scattered CAT: A critical evaluation of the consensual assessment technique for creativity research. *Psychology of Aesthetics, Creativity, and the Arts*, 13(2):159, 2019. doi:[10.1037/aca0000220](https://doi.org/10.1037/aca0000220).
- Daniel Frey and David Gossard. 2.007 Design And Manufacturing I. <https://ocw.mit.edu/courses/2-007-design-and-manufacturing-i-spring-2009>, spring 2009. Massachusetts Institute of Technology: MIT OpenCourseWare, Accessed: 2023-10-20.
- Madhurima Das and Maria C. Yang. Assessing Early Stage Design Sketches and Reflections on Prototyping. *Journal of Mechanical Design*, 144(4):041403, 02 2022. doi:[10.1115/1.4053463](https://doi.org/10.1115/1.4053463).
- Mark Drela, Steven Hall, Paul A. Lagace, Ingrid Kristina Lundqvist, Gustaf Naeser, Heidi Perry, Raúl Radovitzky, and Ian A. Waitz. Unified engineering I, II, III, & IV (supplementary notes for lectures m17-m20). https://ocw.mit.edu/courses/16-01-unified-engineering-i-ii-iii-iv-fall-2005-spring-2006/resources/zm17_20/, 2005a. Massachusetts Institute of Technology: MIT OpenCourseWare, Accessed: 2023-11-18.
- Mark Drela, Steven Hall, Paul A. Lagace, Ingrid Kristina Lundqvist, Gustaf Naeser, Heidi Perry, Raúl Radovitzky, Ian A. Waitz, Peter Young, and Jennifer L. Craig. Unified engineering I, II, III, & IV (lecture notes). <https://ocw.mit.edu/courses/16-01-unified-engineering-i-ii-iii-iv-fall-2005-spring-2006/resources/zm21/>, 2005b. Massachusetts Institute of Technology: MIT OpenCourseWare, Accessed: 2023-11-18.
- Thomas D Economon, Francisco Palacios, Sean R Copeland, Trent W Lukaczyk, and Juan J Alonso. SU2: An open-source suite for multiphysics simulation and design. *AIAA Journal*, 54(3):828–846, 2016. doi:[10.2514/1.J053813](https://doi.org/10.2514/1.J053813).
- Kristen M. Edwards, Aoran Peng, Scarlett R. Miller, and Faez Ahmed. If a Picture is Worth 1000 Words, Is a Word Worth 1000 Features for Design Metric Estimation? *Journal of Mechanical Design*, 144(4):041402, 12 2021. doi:[10.1115/1.4053130](https://doi.org/10.1115/1.4053130).
- Yan Fu, Liming Wang, and Long Li. Conceptual design scheme automatic generation and decision-making considering green demand. *Procedia Manufacturing*, 43:407–414, 2020. doi:[10.1016/j.promfg.2020.02.194](https://doi.org/10.1016/j.promfg.2020.02.194).
- Shuming Gao and Jami J Shah. Automatic recognition of interacting machining features based on minimal condition subgraph. *Computer-Aided Design*, 30(9):727–739, 1998. doi:[10.1016/S0010-4485\(98\)00033-5](https://doi.org/10.1016/S0010-4485(98)00033-5).
- Giorgio Giannone, Akash Srivastava, Ole Winther, and Faez Ahmed. Aligning optimization trajectories with diffusion models for constrained design generation, 2023. URL <https://arxiv.org/abs/2305.18470>.
- Kathryn Henderson. *On Line and on Paper: Visual Representations, Visual Culture, and Computer Graphics in Design Engineering*. Inside Technology. MIT Press, Cambridge, MA, 1999. ISBN 978-0-262-08269-3.
- Amin Heyrani Nobari, Wei Chen, and Faez Ahmed. Pcdgan: A continuous conditional diverse generative adversarial network for inverse design. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD ’21, page 606–616, 2021. doi:[10.1145/3447548.3467414](https://doi.org/10.1145/3447548.3467414).
- Hubs. What are the key design elements for 3d printing? <https://www.hubs.com/knowledge-base/key-design-considerations-3d-printing/>, 2023. Accessed: 2023-11-18.
- William Hunter et al. ToPy - Topology optimization with python. <https://github.com/williamhunter/topy>, 2017.
- Cole Jetton, Matthew I Campbell, and Christopher Hoyle. Constraining the feasible design space in Bayesian optimization with user feedback. *Journal of Mechanical Design*, pages 1–35, 2023. doi:[10.1115/1.4063906](https://doi.org/10.1115/1.4063906).
- Baer John and S. McKool Sharon. Assessing creativity using the consensual assessment technique. In *Handbook of Research on Assessment Technologies, Methods, and Applications in Higher Education*, pages 65–77. IGI Global, 2009. doi:[10.4018/978-1-60566-667-9.ch004](https://doi.org/10.4018/978-1-60566-667-9.ch004).
- Jung-Hoon Chun and Sang-Gook Kim. 2.008 Design And Manufacturing II. <https://ocw.mit.edu/courses/2-008-design-and-manufacturing-ii-spring-2004>, spring 2004. Massachusetts Institute of Technology: MIT OpenCourseWare, Accessed: 2023-10-20.

- Daniel Martin Katz, Michael James Bommarito, Shang Gao, and Pablo Arredondo. GPT-4 passes the bar exam. Available at SSRN 4389233, 2023. doi:[10.2139/ssrn.4389233](https://doi.org/10.2139/ssrn.4389233).
- Andrei Nikolaevich Kolmogorov. Dissipation of energy in the locally isotropic turbulence. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 434(1890):15–17, July 1991a. doi:[10.1098/rspa.1991.0076](https://doi.org/10.1098/rspa.1991.0076).
- Andrei Nikolaevich Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large Reynolds numbers. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 434(1890):9–13, July 1991b. doi:[10.1098/rspa.1991.0075](https://doi.org/10.1098/rspa.1991.0075).
- Andrey Nikolaevich Kolmogorov. A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high reynolds number. *Journal of Fluid Mechanics*, 13(1):82–85, 1962. doi:[10.1017/S0022112062000518](https://doi.org/10.1017/S0022112062000518).
- Fuxiao Liu, Tianrui Guan, Zongxia Li, Lichang Chen, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusion-Bench: You see what you think? or you think what you see? an image-context reasoning benchmark challenging for GPT-4V(ision), LLaVA-1.5, and other multi-modality models, 2023a. URL <https://arxiv.org/abs/2310.14566>.
- Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved Baselines with Visual Instruction Tuning, 10 2023b. URL <https://arxiv.org/abs/2310.03744>.
- Kai Liu and Andrés Tovar. An efficient 3D topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 50:1175–1196, 2014. doi:[10.1007/s00158-014-1107-x](https://doi.org/10.1007/s00158-014-1107-x).
- Harvard Lomax, Thomas H Pulliam, David W Zingg, Thomas H Pulliam, and David W Zingg. *Fundamentals of computational fluid dynamics*, volume 246. Springer, 2001. doi:[10.1007/978-3-662-04654-8](https://doi.org/10.1007/978-3-662-04654-8).
- Yukiko Maeda and So Yoon Yoon. A Meta-Analysis on Gender Differences in Mental Rotation Ability Measured by the Purdue Spatial Visualization Tests: Visualization of Rotations (PSVT:R). *Educational Psychology Review*, 25(1): 69–94, March 2013. doi:[10.1007/s10648-012-9215-x](https://doi.org/10.1007/s10648-012-9215-x).
- Liane Makatura, Michael Foshey, Bohan Wang, Felix Hähnlein, Pingchuan Ma, Bolei Deng, Megan Tjandrasuwita, Andrew Spielberg, Crystal Elaine Owens, Peter Yichen Chen, Allan Zhao, Amy Zhu, Wil J. Norton, Edward Gu, Joshua Jacob, Yifei Li, Adriana Schulz, and Wojciech Matusik. How Can Large Language Models Help Humans in Design and Manufacturing?, July 2023. URL <https://arxiv.org/abs/2307.14377>.
- James Manyika and Sissie Hsiao. An overview of Bard: An early experiment with generative AI. <https://ai.google/static/documents/google-about-bard.pdf>, 2023.
- Rui Mao, Guanyi Chen, Xulang Zhang, Frank Guerin, and Erik Cambria. GPTEval: A survey on assessments of ChatGPT and GPT-4, 2023. URL <https://arxiv.org/abs/2308.12488>.
- François Mazé and Faez Ahmed. Diffusion models beat GANs on topology optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, Washington, DC, 2023. doi:[10.1609/aaai.v37i8.26093](https://doi.org/10.1609/aaai.v37i8.26093).
- Scarlett R. Miller, Samuel T. Hunter, Elizabeth Starkey, Sharath Ramachandran, Faez Ahmed, and Mark Fuge. How Should We Measure Creativity in Engineering Design? A Comparison Between Social Science and Engineering Approaches. *Journal of Mechanical Design*, 143(3):031404, 01 2021. doi:[10.1115/1.4049061](https://doi.org/10.1115/1.4049061).
- Martin Mundt, Sagnik Majumder, Sreenivas Murali, Panagiotis Panetsos, and Visvanathan Ramesh. Meta-learning convolutional neural architectures for multi-target concrete defect classification with the concrete defect bridge image dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11196–11205, 2019.
- Ananya Nandy and Kosa Goucher-Lambert. Do Human and Computational Evaluations of Similarity Align? An Empirical Study of Product Function. *Journal of Mechanical Design*, 144(4):041404, 03 2022. doi:[10.1115/1.4053858](https://doi.org/10.1115/1.4053858).
- Matt D Nelson, Brady L Goenner, and Bruce K Gale. Utilizing ChatGPT to assist CAD design for microfluidic devices. *Lab on a Chip*, 23(17):3778–3784, 2023. doi:[10.1039/D3LC00518F](https://doi.org/10.1039/D3LC00518F).
- Zhenguo Nie, Tong Lin, Haoliang Jiang, and Levent Burak Kara. Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain. *Journal of Mechanical Design*, 143(3), February 2021. doi:[10.1115/1.4049533](https://doi.org/10.1115/1.4049533).
- Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of GPT-4 on medical challenge problems, 2023. URL <https://arxiv.org/abs/2303.13375>.
- Wilson R Nyemba. *Computer Aided Design: Engineering Design and Modeling Using AutoCAD*. CRC Press, 2022. doi:[10.1201/9781003288626](https://doi.org/10.1201/9781003288626).

- Gul E. Okudan and Shafin Tauhid. Concept selection methods – a literature review from 1980 to 2008. *International Journal of Design Engineering*, 2(3):243–277, 2009. doi:[10.1504/IJDE.2008.023764](https://doi.org/10.1504/IJDE.2008.023764).
- OpenAI. GPT-4V(ision) system card. Technical report, OpenAI, 2023.
- Gerhard Pahl, Wolfgang Beitz, Joerg Feldhusen, and Karl-Heinrich Grote. *Engineering Design: A Systematic Approach*. Springer London, 2007. ISBN 978-1-84628-318-5. doi:[10.1007/978-1-84628-319-2](https://doi.org/10.1007/978-1-84628-319-2).
- Cyril Picard. MechE Rotation Test, 11 2023.
- Stephen B Pope. Turbulent flows. *Measurement Science and Technology*, 12(11):2020–2021, 2001. doi:[10.1088/0957-0233/12/11/705](https://doi.org/10.1088/0957-0233/12/11/705).
- Stuart Pugh. *Total Design*. Addison-Wesley, 1991.
- Stuart Pugh. Concept selection – a method that works. In *International Conference of Engineering Design*, pages 497–506, 1995.
- Claudia Quaiser-Pohl. The Mental Cutting Test "Schnitte" and the Picture Rotation Test—Two New Measures to Assess Spatial Ability. *International Journal of Testing*, 3(3):219–231, 2003. doi:[10.1207/S15327574IJT0303_2](https://doi.org/10.1207/S15327574IJT0303_2).
- Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, et al. Skilful precipitation nowcasting using deep generative models of radar. *Nature*, 597(7878):672–677, 2021. doi:[10.1038/s41586-021-03854-z](https://doi.org/10.1038/s41586-021-03854-z).
- Abdullahi Saka, Ridwan Taiwo, Nurudeen Saka, Babatunde Salami, Saheed Ajayi, Kabiru Akande, and Hadi Kazemi. GPT models in construction industry: Opportunities, limitations, and a use case validation, 2023. URL <https://arxiv.org/abs/2305.18997>.
- Mohammad Saravi, Linda Newnes, Antony Roy Mileham, and Yee Mey Goh. Estimating cost at the conceptual design stage to optimize design in terms of performance and cost. In *Collaborative Product and Service Life Cycle Management for a Sustainable World*, pages 123–130, London, 2008. Springer London. doi:[10.1007/978-1-84800-972-1_11](https://doi.org/10.1007/978-1-84800-972-1_11).
- Jami J Shah, Steve M Smith, and Noe Vargas-Hernandez. Metrics for measuring ideation effectiveness. *Design studies*, 24(2):111–134, 2003. doi:[10.1016/S0142-694X\(02\)00034-0](https://doi.org/10.1016/S0142-694X(02)00034-0).
- Daniel L. Shea, David Lubinski, and Camilla P. Benbow. Importance of assessing spatial ability in intellectually talented young adolescents: A 20-year longitudinal study. *Journal of Educational Psychology*, 93(3):604–614, 2001. doi:[10.1037/0022-0663.93.3.604](https://doi.org/10.1037/0022-0663.93.3.604).
- Yongxin Shi, Dezhi Peng, Wenhui Liao, Zening Lin, Xinhong Chen, Chongyu Liu, Yuyi Zhang, and Lianwen Jin. Exploring OCR capabilities of GPT-4V(ision): A quantitative and in-depth evaluation, 2023. URL <https://arxiv.org/abs/2310.16809>.
- L. Siddharth, Lucienne Blessing, and Jianxi Luo. Natural language processing in-and-for design research. *Design Science*, 8:e21, January 2022. doi:[10.1017/dsj.2022.16](https://doi.org/10.1017/dsj.2022.16).
- Ole Sigmund. A 99 line topology optimization code written in matlab. *Structural and multidisciplinary optimization*, 21(2):120–127, April 2001. doi:[10.1007/s001580050176](https://doi.org/10.1007/s001580050176).
- Ole Sigmund and Kurt Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, December 2013. doi:[10.1007/s00158-013-0978-6](https://doi.org/10.1007/s00158-013-0978-6).
- Binyang Song, Scarlett Miller, and Faez Ahmed. Attention-Enhanced Multimodal Learning for Conceptual Design Evaluations. *Journal of Mechanical Design*, 145(4):041410, 02 2023a. doi:[10.1115/1.4056669](https://doi.org/10.1115/1.4056669).
- Binyang Song, Rui Zhou, and Faez Ahmed. Multi-modal machine learning in engineering design: A review and future directions, 2023b. URL <https://arxiv.org/abs/2302.10909>.
- Elizabeth Starkey, Christine A. Toh, and Scarlett R. Miller. Abandoning creativity: The evolution of creative ideas in engineering design course projects. *Design Studies*, 47:47–72, 2016. doi:[10.1016/j.destud.2016.08.003](https://doi.org/10.1016/j.destud.2016.08.003).
- Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. GPT-4 Doesn't Know It's Wrong: An Analysis of Iterative Prompting for Reasoning Problems, 10 2023. URL <https://arxiv.org/abs/2310.12397>.
- Francesco Stella, Cosimo Della Santina, and Josie Hughes. How can LLMs transform the robotic design process? *Nature Machine Intelligence*, 5(6):561–564, June 2023. doi:[10.1038/s42256-023-00669-7](https://doi.org/10.1038/s42256-023-00669-7).
- Hanqi Su, Binyang Song, and Faez Ahmed. Multi-modal machine learning for vehicle rating predictions using image, text, and parametric data. In *Proceedings of the International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Boston, MA, 2023. ASME.

- Roman Taraban. Information fluency growth through engineering curricula: Analysis of students' text-processing skills and beliefs. *Journal of Engineering Education*, 100(2):397–416, 2011. doi:[10.1002/j.2168-9830.2011.tb00019.x](https://doi.org/10.1002/j.2168-9830.2011.tb00019.x).
- Christine Toh and Scarlett R. Miller. Does the Preferences for Creativity Scale Predict Engineering Students' Ability to Generate and Select Creative Design Alternatives? *Journal of Mechanical Design*, 141(6):062001, 04 2019. doi:[10.1115/1.4042154](https://doi.org/10.1115/1.4042154).
- Christine A. Toh and Scarlett R. Miller. Choosing creativity: the role of individual risk and ambiguity aversion on creative concept selection in engineering design. *Research in Engineering Design*, 27:195–219, 2016. doi:[10.1007/s00163-015-0212-1](https://doi.org/10.1007/s00163-015-0212-1).
- Karl T. Ulrich, Steven D. Eppinger, and Maria C. Yang. *Product Design and Development*. McGraw-Hill, New York, NY, seventh edition edition, 2020. ISBN 978-1-260-04365-5.
- Paul-Armand. Verhaegen, Dennis Vandevenne, and Joost R. Duflo. Originality and novelty: A different universe. In *International Design Conference - DESIGN 2012*, pages 1961–1966, Dubrovnik, Croatia, May 21 - 24 2012.
- Ricardo Vinuesa and Steven L Brunton. Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, 2(6):358–366, 2022. doi:[10.1038/s43588-022-00264-7](https://doi.org/10.1038/s43588-022-00264-7).
- Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. SciBench: Evaluating college-level scientific problem-solving abilities of large language models, 2023. URL <https://arxiv.org/abs/2307.10635>.
- Clair Webb. 45nm design for manufacturing. *Intel Technology Journal*, 12(2), 2008.
- Licheng Wen, Xuemeng Yang, Daocheng Fu, Xiaofeng Wang, Pinlong Cai, Xin Li, Tao Ma, Yingxuan Li, Linran Xu, Dengke Shang, Zheng Zhu, Shaoyan Sun, Yeqi Bai, Xinyu Cai, Min Dou, Shuanglu Hu, and Botian Shi. On the road with GPT-4V(ision): Early explorations of visual-language model on autonomous driving, 2023. URL <https://arxiv.org/abs/2311.05332>.
- John F Wendt. *Computational fluid dynamics: an introduction*. Springer Science & Business Media, 2008.
- Rebekka V Woldseth, Niels Aage, J Andreas Barentzen, and Ole Sigmund. On the use of artificial neural networks in topology optimisation. *Structural and Multidisciplinary Optimization*, 65(10):294, 2022. doi:[10.1007/s00158-022-03347-1](https://doi.org/10.1007/s00158-022-03347-1).
- Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-Mark Prompting Unleashes Extraordinary Visual Grounding in GPT-4V, 10 2023a. URL <https://arxiv.org/abs/2310.11441>.
- Sheng Yang and Yaoyao Fiona Zhao. Additive manufacturing-enabled design theory and methodology: a critical review. *The International Journal of Advanced Manufacturing Technology*, 80:327–342, 2015. doi:[10.1007/s00170-015-6994-5](https://doi.org/10.1007/s00170-015-6994-5).
- Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of LLMs: Preliminary explorations with GPT-4V(ision), 2023b. URL <https://arxiv.org/abs/2309.17421>.
- So Yoon Yoon. *Psychometric Properties of the Revised Purdue Spatial Visualization Tests: Visualization of Rotations (the Revised PSVT:R)*. PhD thesis, Purdue University, United States – Indiana, 2011.
- Chenxi Yuan, Tucker Marion, and Mohsen Moghaddam. Leveraging End-User Data for Enhanced Design Concept Evaluation: A Multimodal Deep Regression Model. *Journal of Mechanical Design*, 144(2):021403, 09 2021. doi:[10.1115/1.4052366](https://doi.org/10.1115/1.4052366).
- Guanglu Zhang, Ayush Raina, Ethan Brownell, and Jonathan Cagan. Artificial Intelligence Impersonating a Human: The Impact of Design Facilitator Identity on Human Designers. *Journal of Mechanical Design*, 145(5):051404, 01 2023. doi:[10.1115/1.4056499](https://doi.org/10.1115/1.4056499).
- Zhibo Zhang, Prakhar Jaiswal, and Rahul Rai. FeatureNet: Machining feature recognition based on 3D convolution neural network. *Computer-Aided Design*, 101:12–22, 2018. doi:[10.1016/j.cad.2018.03.006](https://doi.org/10.1016/j.cad.2018.03.006).
- Mingyu Zong and Bhaskar Krishnamachari. Solving math word problems concerning systems of equations with GPT models. *Machine Learning with Applications*, 14:100506, 2023. doi:[10.1016/j.mlwa.2023.100506](https://doi.org/10.1016/j.mlwa.2023.100506).