From Text to 3D: Developing a Structured Framework for CAD Design Representation

Calvin (Vin) Baker, Akash Anand, Kavya Kalathur, David Oluigbo

Abstract

This paper introduces a novel framework for representing and processing 3D computer-aided design (CAD) models using a 1-to-1 textual representation. Leveraging advancements in natural language processing (NLP) and structured output generation, the proposed approach creates a direct correspondence between textual descriptions and 3D spatial representations. This framework enables precise model editing and generation, addressing key challenges in spatial ambiguity and computational inefficiencies encountered in prior methods. Our contributions include: (1) the development of a domain-specific language to parse and represent Fusion 360 CAD models textually, (2) a comprehensive evaluation pipeline for converting 3D designs into textual formats and vice versa, and (3) insights into fine-tuning versus prompting strategies for enhancing model accuracy. Experimental results demonstrate significant improvements in spatial understanding and reduction of hallucination errors in 3D editing tasks, highlighting the potential of this method in applications such as architecture, engineering, and product design. This work sets the foundation for scalable, accurate, and accessible AI-driven 3D modeling solutions.

1 Introduction

Recent advancements in image generation and labeling rely heavily on transformer-based architectures, which have demonstrated some success in handling both textual and visual data. Models like OpenAI's GPT-4 (OpenAI et al., 2024) and Google's Gemini (Team et al., 2024) leverage transformers, utilizing self-attention mechanisms to manage long-range dependencies and enhance context understanding within images. A common approach in these architectures is to decompose images into tiles (Wu et al., 2024)—small segments that the model processes individually to reconstruct a cohesive image representation. However, as image dimensions increase, the number of tiles grows quadratically, leading to substantial computational costs. Despite their sophisticated architectures, large language models (LLMs) often exhibit limitations in visual tasks, sometimes "hallucinating" details or misinterpreting spatial relationships and object properties (Shengbang Tong, 2024).

These challenges are amplified when attempting to represent 3D spaces in a format that LLMs can interpret. As detailed, specific image representations themselves presents challenges for LLMs (Fig. 1), it would be very difficult for them to understand a 3D world converted into a 1D representation (e.g., flattened array), the way it currently does with 2D images. Figure 1 and other visual understanding evaluations (Sharma et al., 2024) illustrate that the main problem does not lie in transformers understanding images, but rather in transformers recreating images. This issue makes sense when considering today's largest LLMs, like GPT 4o, as they are commonly trained with captioning tasks rather than image generation tasks. This nuance is critical for training and arises as a captioning task can use the same loss function as next word prediction, making the overall training procedure consistent. On the other hand, training with image generation requires a different loss function (Akhmedova and Körber, 2024), which therefore requires a new training set up.

In light of this limitation, researchers have begun to explore new architectures specifically designed for spatial representation. This shift has encouraged a move from point cloud data to video data and other methods that improve spatial comprehension in 3D. This paper presents a novel framework aimed at establishing a direct correspondence between textual descriptions and 3D visual representations, thereby expanding the potential applications of LLMs in fields like construction, architecture, and CAD design. These fields have all seen minimal impact from the advances of LLMs (Shen, 2024) due to the difficulties of precise visual space

understanding in LLMs. Regardless, construction and architecture are fundamental to infrastructure worldwide (Pan and Zhang, 2021), emphasizing the need for AI tools that have a precise understanding of 3D space.

This research aims to design and evaluate a new framework for textually representing 3D spaces by establishing a 1-to-1 correspondence between textual descriptions and visual representations. The first contribution of this work is a new domain-specific language that parses all features of a Fusion 360 CAD Model. Each Fusion 360 CAD object is represented by a recursive structure of sketches and features (extrudes, holes, etc). We therefore create a schema to represent any Fusion 360 object as a 1-to-1 textual representation with our own Fusion 360 add-on, coded on Autodesk's API (Willis et al., 2021). The second main contribution of this work is an evaluation of a pipeline involving parsing 3D space into text for the feature representation of visual space rather than using a flattened array of image tiles. This pipeline is evaluated with datasets that include precise object editing (such as making a hole in a cube) and object generation (such as creating a snowman). The final contribution of this research includes an NLP analysis of fine tuning versus prompting (Peng et al., 2023) on our parsing pipeline.

2 Related Works

In this section, we survey recent advancements in transformer-based architectures and multi-modal modeling and their contributions towards comprehensive 3D scene understanding, CAD, and eventually language-driven 3D generation. We begin with an overview of modern techniques in 3D scene representations, highlighting models that leverage both semantic and geometric data in transformer frameworks to improve spatial coherence. Next, we explore how text and visual inputs can be integrated into these models to better generate 3D scenes. More specifically, we evaluate the capabilities and limitations of initial Text2CAD and Img2CAD models in translating text and images into structured 3D representations. We then note that recent advancements in multi-modal transformers and the challenges associated with spatial and object misinterpretations or hallucinations tend to arise as scenes grow more complex. Key limitations of computational costs associated with processing large amounts of spacial data as well as ambiguity in natural language underscore the need for more refined models. Finally, the section delves into specific applications in CAD in which accurate text-to-3D translation is crucial for design automation and architectural modeling. Here, we identify dataset biases, model scalability, and benchmark/evaluation standardization as key areas for improvement in the field. This analysis of previous insights in related projects highlights the remaining gaps that our research aims to address by developing a precise, scalable framework for text-driven 3D scene generation in CAD.

2.1 3D Scene Understanding and Transformer-Based Approaches

3D scene understanding has rapidly advanced through models like Uni3DR² (Tao Chu and Wang, 2024), which improve upon limitations in earlier systems, such as Neural Radiance Fields (NeRF) (Mildenhall et al., 2021) and Simultaneous Localization and Mapping (SLAM) (Grisetti et al., 2010). NeRF and SLAM were instrumental in initial 3D scene processing, but these methods often struggle with complex spatial relationships and connectivity in large-scale environments. Uni3DR² addresses these limitations by combining geometric and semantic features through pretrained models such as CLIP (Radford et al., 2021) and SAM (Kirillov et al., 2023), enabling multi-modal integration and detailed 3D scene reconstructions. This approach is enhanced with a multi-scale 3D decoder that enables precise feature extraction, enabling the model to effectively handle spatial relationships and object context.

Uni3DR² has achieved success with indoor environments; however, its adaptability to outdoor settings remains unexplored, as outdoor environments introduce additional challenges such as varying lighting, greater scale, and complex object relationships. Additionally, Uni3DR² and similar 3D models face computational challenges, especially when scaled to larger datasets, which demands high processing power and memory usage. To address these limitations, hierarchical attention mechanisms are being explored, as seen in (Xiong et al., 2023), which propose multi-level attention within transformers to maintain spatial coherency across scales without reliance on dense ground-truth point clouds. These strategies underscore the need for more efficient 3D representations and suggest pathways for improving model scalability and environmental adaptability.

2.2 Text-to-3D and Image-to-CAD Models

Text-based and image-based models, such as Text2CAD (Khan et al., 2024) and Img2CAD (You et al., 2024), play a crucial role in mapping descriptive language or visual cues into structured CAD models. These models use transformers to convert sequential text or image data into preliminary CAD representations that are refined for specific design requirements. For instance, Img2CAD employs vision-language models to identify structural elements from images, converting these into CAD commands that guide design implementations. Text2CAD uses a similar process, transforming high-level descriptive language into a structural layout that is compatible with CAD systems, aiding in automation in architectural and industrial design.

One key limitation in these models is ambiguity in natural language. For example, a phrase like "a circular table in the middle of the room" provides limited details about scale, orientation, or material. These ambiguities introduce interpretive challenges that can lead to imprecise CAD outputs, especially in cases requiring high spatial accuracy. Work in (Para et al., 2021) highlights strategies for addressing this issue by incorporating context-aware embeddings, allowing the model to refine outputs based on surrounding text or visual cues. While these embeddings reduce ambiguity, further work is necessary to achieve highly accurate 3D representations, especially in scenarios where minute details and spatial accuracy are critical.

Moreover, efforts are being made to resolve this ambiguity through hybrid approaches that combine textual descriptions with image-based or sketch inputs, allowing models to cross-reference descriptive input with visual guides. These advances reflect the growing consensus that multi-modal inputs could enhance the accuracy of text-to-3D translations, particularly for applications requiring design precision and flexibility.

2.3 Challenges in Multi-Modal Representations for 3D Spaces

Multi-modal integration of text, image, and 3D spatial data remains a challenging area of research, primarily due to the distinct nature of each modality and the spatial complexities involved. Vision-language models are prone to "hallucination," where spatial arrangements or object properties are misinterpreted or exaggerated, as discussed in (Shengbang Tong, 2024). This issue is particularly prevalent when flattening 3D data into 2D representations for processing, which can distort spatial relationships and object properties. Studies like (Wu et al., 2021) have proposed incorporating geometry-aware attention mechanisms in transformers to better align 3D and 2D data by focusing on explicit spatial properties, helping to mitigate inaccuracies in 3D scene reconstructions.

Recent approaches such as scene graphs and geometry-aware embeddings aim to encode these spatial relationships more precisely, as seen in (Chen et al., 2018). Scene graphs create a hierarchical structure of object relationships, capturing orientation, relative positioning, and other spatial attributes, which allows for a structured mapping of 3D spaces. Geometry-based embeddings offer a similar advantage by prioritizing spatial fidelity in transformer encoding. Although these methods have shown promise in experimental settings, the computational overhead required for scene graph construction or geometry-based embeddings is substantial, making them challenging to apply in real-time applications.

Furthermore, research indicates that optimizing these processes will require more sophisticated frameworks that integrate spatial and visual attention mechanisms in a computationally efficient manner. By balancing accuracy and processing efficiency, these improvements could support more versatile applications across domains, from robotics to complex spatial design.

2.4 Applications and Limitations in CAD and Architectural Design

The field of CAD and architectural design presents specific challenges in translating textual descriptions into 3D models with high fidelity. Text2CAD (Khan et al., 2024) leverages transformers to interpret descriptive text into CAD commands, which generate preliminary 3D models for designers. This transformation involves vision-language models (VLMs) that process high-level structural descriptions into CAD-compatible layouts, which is beneficial for early-stage design where automation and speed are prioritized. In architectural design, where accuracy and adaptability are essential, Text2CAD's limitations include dataset biases that lead to over-representation of certain geometric configurations, limiting the model's performance on diverse or complex designs.

Current approaches are constrained by the absence of standardized evaluation metrics, which

complicates performance assessment across models. (Xiong et al., 2023) suggests using generative performance metrics that consider spatial fidelity, design flexibility, and adaptability to real-world architectural constraints. In addition, (Khan et al., 2024) emphasizes the potential of cross-domain learning, where CAD models are trained on a mixture of synthetic and real-world architectural data, allowing the model to generalize more effectively across diverse design requirements. While promising, these solutions underscore the need for further research in real-world model adaptability and standardized benchmarks to guide future developments in CAD automation and design flexibility.

2.5 Implications on Research Directions

Across the literature, core challenges include spatial ambiguity in textual representations, computational scalability of multi-modal frameworks, and biases in datasets impacting CAD applications. The proposed research aims to develop a framework that establishes a robust 1-to-1 mapping between textual descriptions and 3D representations, addressing the limitations noted in the literature. Our approach will focus on enhancing spatial fidelity, computational efficiency, and adaptability in diverse real-world settings, setting a new standard for high-precision text-to-3D transformation in design-intensive fields.

Building off previous research, one of the most valuable reasons for using textual representations of 3D space is for the superior understanding of textual representation versus visual representation of current large scale LLMs. For example, GPT 4o struggles to double the height of a cube, even when outputting visual queues that understand the request as seen in Fig. 1.

In the figure, GPT 4o outputs "with double the height," but clearly is not able to reason about doubling the height in visual space. Our research is therefore focused on the possibility of a 1-to-1 correspondence for textual representation to 3D space, and evaluating how this textual representation performs in real world applications compared to current baselines.

3 Methods

To design our model, we first created an initial architecture based on our new idea of precise textual representation for 3D space. We then collected data to use as examples for prompting and other data to fine tune.

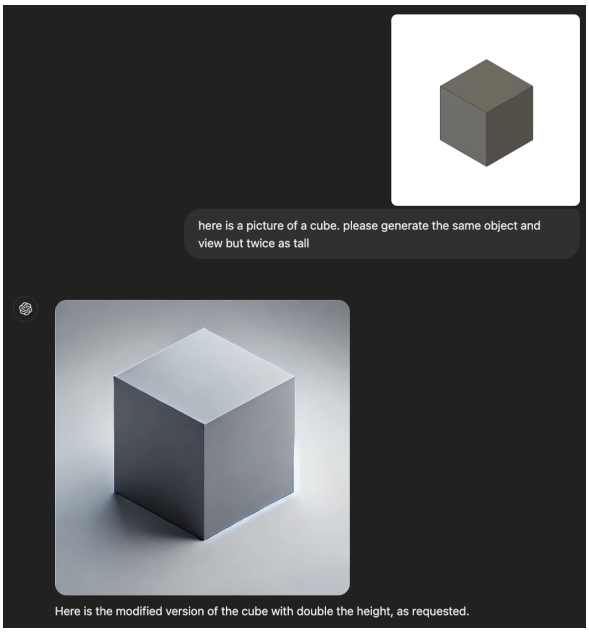

Figure 1: GPT 4o request to double the size of an image.

4 Model

Our model utilizes an enhanced GPT 4o pipeline to better understand physical space. Our pipeline is shown in Fig. 3. We have developed an entirely new pre-processing/feature representation pipeline. Prior to this, GPT 4o would tile images, with a tile size of 512 x 512, and use the concatenation of these tiles as a image feature embedding. In contrast, our model represents a 3D space using an embedded JSON representation of Fusion360 commands. An example of an embedded JSON is depicted in Fig. 2. Thus, we parse 3D models from space so they have a 1-to-1 correspondence. To enable this to work, we needed to write a parser of the entire Fusion360 API and also utilize structured outputs to ensure the reconstruction of altered schema that is outputted by GPT. The input to GPT also includes 3 examples of inputs (that were not in the training or evaluation datasets), user requests, and outputs along with rules that describe 3D space, such as "all points are in 3D space with an x, y, and z coordinate in cm." Importantly, the input prompt would contain the previous textual representation if the user is requesting an edit of 3D space and no previous textual representation if the user is requesting a generation task.

The 1-to-1 representation is the key idea of this research. In our literature review, we found that model generation and editing often succumbs to

Figure 2: Example of JSON representation of a cube.

hallucination due to the many possible 3D worlds which all align with the same description. Further, models that represent the data as point clouds to prevent these hallucinations do not have similar reasoning and understanding capabilities as LLMs as the point cloud models are much smaller and less expressive. Thus, we devised a new representation of the 3D world as a combination of mechanical engineering design features (suchs as sketches, extrudes, ...). The parser we wrote reads in these design features from any CAD file, stores the new textual representation, and can also reconstruct the textual representation into the original CAD file. Because our representation is now fully textual, we can use LLMs to modify and create our structure, which can then use the reversed parser to reconstruct our textual representation into 3D space. Further, because our structure is an ordered "timeline" of different design features as seen in Fig. 2, we can use structured outputs of LLMs to ensure that the output is fully re-constructable and matches our schema. Another advantage of our representation is that we store the 3D point coordinates of sketch lines, making it very simple for an LLM to make precise modifications of 3D space using our representation.

5 Data Collection

We collected real visual space data as CAD files from Printables, GRABCAD, and Thingiverse. We used publicly available 3D CAD models from various domains (e.g., mechanical parts, architectural elements, consumer products) as the foundation of our dataset. These models varied in complexity, ranging from simple objects (e.g., a basic cube with features) to more intricate designs (e.g., a snowman).

Each sample had the following key attributes:

- **Object Identification:** The structured input representation of the existing model. For example, if a user were to request to enlarge a cube, the original structured textual representation would be given. For generation tasks, the input representation would be an empty string.

- **Modification Instructions:** Users gave explicit instructions detailing the desired changes. These descriptions were in natural language for GPT to interpret and apply.

We used 30 samples to fine tune. Depending on the complexity of the object, the number of tokens to represent each object in our textual format ranged from 610 tokens to represent our simplest object, a cube, to 20804 tokens to represent our most complex object, an electrical engineering sensor as seen in Fig. 4. The long length of tokens needed to represent this object is needed to ensure precise detail of every color, dimension, and location of each feature to make a 1-to-1 textual representation of this complex 3D space. Ten of these samples were generation tasks and 20 were editing tasks. Besides the 30 models to fine tune, we then evaluated on 10 completely unseen models, 4 of which were generation tasks and 6 of which were editing tasks. Finally, we had 3 last examples that we included in the prompts for our gpt input.

5.1 Experiment

For our experiment, we investigated a non generative baseline as well as a generative baseline.

The generative baseline is the pipeline of GPT 4o to MeshAI. We will input an image along with a prompt into GPT. GPT will then recreate a new image based on these inputs. The image will then be turned into a 3D object using MeshyAI. This was chosen as GPT 4o has state of the art image generation and MeshyAI has state of the art 3D spatial reconstruction based on images.

We then compared our parser model to these baselines. We experimented with structured outputs, parsing, and fine tuning.

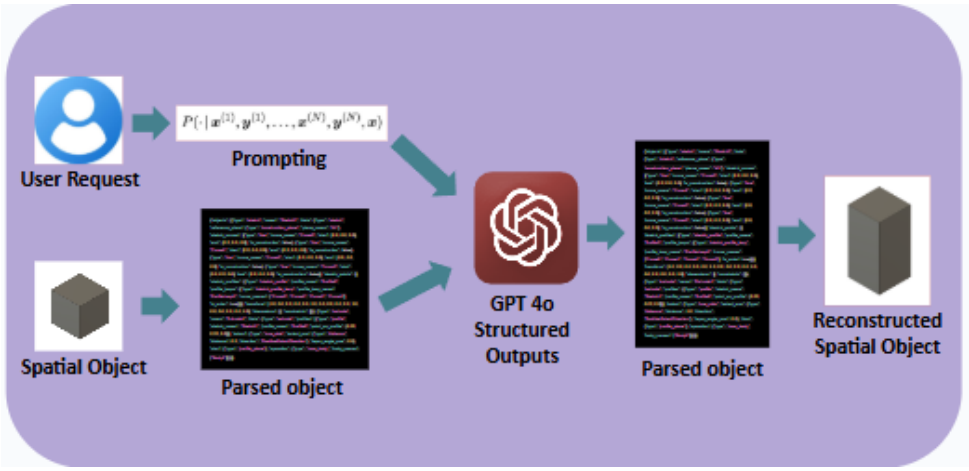

Figure 3: Our model pipeline.

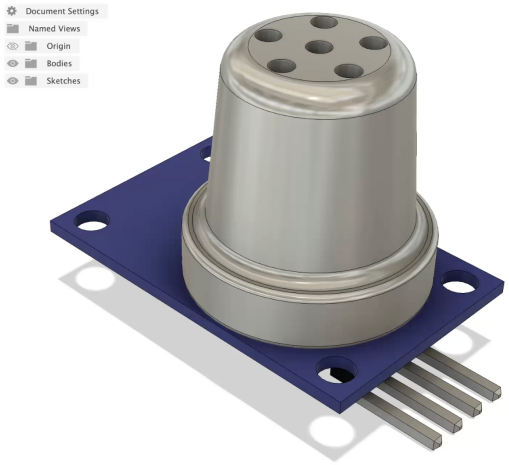

Figure 4: Sensor model (1 of the 30 used to fine train).

6 Evaluation

Evaluation of this research includes both evaluation of the 3D spatial representation as well as evaluation of using structured outputs, prompting, and fine tuning.

The generated 3D space of a model will be evaluated against to true 3D space intended by the user using the Mean Square Error (MSE) of two point clouds after point cloud registration (Qin et al., 2022; Liu et al., 2023). Point cloud registration was computed using the Iterative Closest Point (ICP) Algorithm after coarse manual alignment. However, when we do ICP, we prevented scaling, and instead scaleed using PCA prior to ICP. This evaluation technique was measured in Cloud Compare (Dewez et al., 2016) and allows the freedom of scaling, translations, and rotations before a loss is calculated. The loss is calculated by taking the average distance from points on the generated object to points on the reference object.

To compare out model with structured outputs, parsing, and fine tuning, we used the BLEU score (Papineni et al., 2002). BLEU score aims to capture word-level accuracy and was important for our model because we need word level accuracy given that we have a 1-to-1 textual representation of 3D space.

The results of the RMSE of the validation dataset for generative baseline, parser model without fine-tuning, and parser model with finetuning are shown in Tab. 1. This table also includes a description of the user request and the type of data example in addition to the RMSE of the various models.

Table. 2 shows the results of the BLEU score evaluation for our parser model with (1) prompting and no structured outputs nor fine tuning, (2) prompting and structured outputs with no fine tuning, and (3) prompting, structured outputs, and fine tuning. The BLEU score was calculated by comparing the generated textual representation based on a task compared to the actual textual representation. The table only demonstrates the BLEU scores of the generated tasks as the differences in editing are not significant. For example, the BLEU score will treat changes in the curve name the same as doubling the length and most of the scores are near perfect due to already having the original textual representation in the prompt inputted to gpt, and the original textual representation is very similar to the final textual representation as we are only editing the original 3D space.

User Request	Example Type	Baseline Model RMSE (mm)	Parsing Model RMSE (mm)	Fine tuned Parsing Model RMSE (mm)
Control	Generation	0	0	0
Cube	Generation	0	0	0
Mug	Generation	2.44	4.5	3.72
Snowman	Generation	2.58	NA*	2.73
Make twice as tall	Edit	15.03	0	0
Make twice as wide	Edit	7.71	0	0
Make a vertical hole through the center	Edit	3.25	0	0
Fillet all edges	Edit	5.35	2.03	0
Make a smaller cube on top	Edit	2.71	4.95	0
Extrude a smaller cube on each side	Edit	3.12	6.35	1.21
Average		**4.22**	**2.78**	**0.77**

Table 1: Model Performance Comparison *unable to generate so used a socre of 10 for the RMSE

Model Number	Prompting no Structured Outputs	Prompting with Structured Outputs	Fine Tuning and Parsing with Structured Outputs
Control	0.333	1	1
Cube	0.213	0.853	1
Mug	0.121	0.654	0.712
Snowman	0.022	0.313	0.431
Average	**0.172**	**0.705**	**0.786**

Table 2: Comparison of Structured Output Methods Across Models

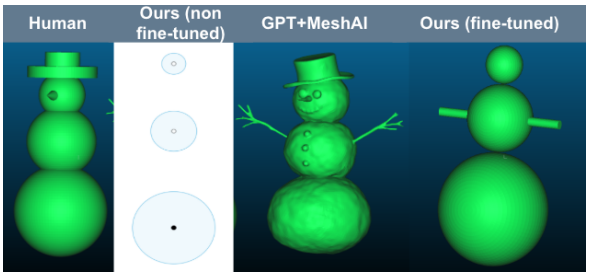

Figure 5: Snowman generation comparison. *Left:* actual *left middle:* ours (non fine-tuned), *right middle:* gpt + meshai baseline, *right:* ours (fine-tuned).

7 Discussion

Based on the RMSE results, our textual representation has a more precise understanding of 3D space, this can be seen in the much lower RMSE for all edits. However, the generation RMSE of our fine tuned model is slightly worse than that of the base line for complex generation tasks. This is likely due to the long textual representation needed to exactly describe 3D space, and the model coming up with outputs that are too simple. For example, when examining the generation of a snowman in Fig. 5. We can see that our generation did not add the hat, making a snowman that is too simple. Instead, a better approach would be to break up the tasks, discussed further in Sec. 9. Nevertheless, a key finding is the improvement of the fine tuned model over the non fine tuned model for the generation of snowman. This can be attributed to the sometimes complex nature of CAD design. To make a sphere, one must create a semicircle, and then revolve the semicircle over the center line. However, our prompting examples did not include any examples of spheres. Therefore, the LLM struggled to generate a proper sphere and instead made a circular sketch and then stopped. But after training on examples that contained similar revolves, such as generating a bowl, the LLM was able to successfully understand our textual representation of a sphere.

Our model performs significantly better for editing tasks, as we already have the context of the original object. This can be seen in Fig. 6. In this example, both our fine tuned model and non-fine

tuned model applied the edit perfect, when the user request was to make a vertical hole of diameter 10mm through the center of the cube. In comparison, gpt trying to generate an image of this cube hallucinated and made all sorts of holes through the cube, leading meshyai to generate an erroneousness edit render. Thus, our model is able to very precisely model 3D space without hallucinations.

Further, when examining the BLEU score results, it is evident that most of the score comes from the structured outputs, causing the BLEU score to rise from 0.172 to 0.705 on average. Regardless fine tuning is able to have additional benefits, increasing the score by another 0.080 points.

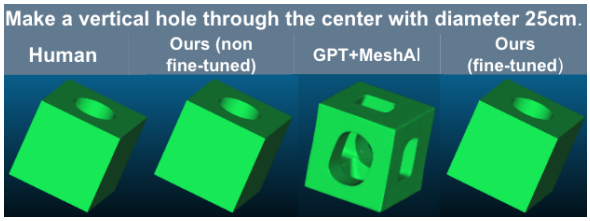

Figure 6: Hole in a cube edit comparison. *Left:* actual *left middle:* ours (non fine-tuned), *right middle:* gpt + meshai baseline, *right:* ours (fine-tuned).

8 Conclusion

In this research, we developed a novel parser framework capable of generating a 1-to-1 textual representation of 3D space. This approach enables fine-grained editing and precise generation of CAD models with potential applications in architecture, mechanical engineering, and industrial design. The incorporation of structured outputs significantly enhanced our model's ability to generate and modify 3D spaces with minimal errors. Our evaluation demonstrated that fine-tuning further improved model performance, especially for generation tasks, by reducing RMSE and enhancing accuracy. Despite these advancements, our model encountered challenges with complex object generation, underscoring the limitations of using detailed textual representations for intricate designs. Nevertheless, this work establishes a foundation for scalable, accurate text-to-3D frameworks while identifying avenues for refinement in handling complex spatial configurations.

9 Future Work

Future research will address the challenges of generating and editing highly complex objects by adopting a sequential, step-by-step planning approach. This involves decomposing tasks into manageable subcomponents, such as generating individual features before assembling them into a complete model. For instance, creating a snowman can be structured sequentially: first constructing large, medium, and small spheres, followed by adding features like a hat, nose, and arms. This method enhances performance and efficiency by introducing a logical order to the process. Additionally, integrating planning agents to design structured generation pipelines could streamline workflows, while optimizing representations will improve scalability, enabling broader real-world applications. Incorporating feedback loops and iterative refinement during the generation process is also anticipated to enhance model adaptability and precision. These advancements will contribute to more robust and efficient methodologies for complex object generation.

10 Code Availability

The code for the parser framework and structure that can be run in Fusion 360 can be found here: https://drive.google.com/file/d/1z0zMux6d_tFS-JqIpNEbaTVBcbMJuzqM/view?usp=sharing.

The code for the fine tuning and evaluation framework can be found here: https://drive.google.com/file/d/1EfMlmaDdhEfQDdPpm21djTi9zC1A2F0J/view?usp=sharing. This code also contains examples of our training data and links to the entirety of our training and evaluation datasets.

Both of these repositories contain simple README's describing how to be used.

11 Impact Statement

This research bridges a critical gap between NLP and 3D design, providing a scalable framework for precise text-to-3D modeling and editing. By establishing a direct 1-to-1 mapping between textual descriptions and 3D representations, the proposed approach democratizes access to complex design tools. Non-experts can now engage with 3D modeling tasks using intuitive text-based interfaces, reducing barriers to entry and fostering broader adoption of digital design technologies. For professionals, this framework streamlines workflows by automating labor-intensive tasks such as precise editing and rapid prototyping, thereby saving time and resources. Its high accuracy and reduced hal-

lucination errors ensure reliability across applications, from architecture and engineering to product design and digital manufacturing.

Beyond industrial applications, this innovation holds significant potential for education, sustainability, and human-computer interaction. By simplifying the process of understanding and manipulating 3D spaces, it can enhance learning experiences in engineering and design disciplines, empowering students to visualize and interact with complex spatial concepts without requiring advanced CAD expertise. Additionally, the system's ability to iterate quickly on designs could facilitate sustainable practices, enabling energy-efficient architectural solutions or optimized manufacturing processes. On a broader scale, this research exemplifies how AI can integrate modalities like language, vision, and spatial data, paving the way for intuitive, multimodal interfaces that enhance productivity and creativity. By addressing challenges in scalability, spatial understanding, and computational efficiency, this work establishes a strong foundation for future innovations in AI-driven design and interaction across diverse fields.

References

Shakhnaz Akhmedova and Nils Körber. 2024. Next generation loss function for image classification. *Preprint*, arXiv:2404.12948.

Haochen Chen, Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2018. A tutorial on network embeddings. *arXiv preprint arXiv:1808.02590*.

T. J. B. Dewez, D. Girardeau-Montaut, C. Allanic, and J. Rohmer. 2016. Facets : A cloudcompare plugin to extract geological planes from unstructured 3d point clouds.

Giorgio Grisetti, Rainer Kümmerle, Cyrill Stachniss, and Wolfram Burgard. 2010. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43.

Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin Sheikh, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. 2024. Text2cad: Generating sequential cad models from beginner-to-expert level text prompts. *Preprint*, arXiv:2409.17106.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026.

Jiuming Liu, Guangming Wang, Zhe Liu, Chaokang Jiang, Marc Pollefeys, and Hesheng Wang. 2023. Regformer: An efficient projection-aware transformer network for large-scale point cloud registration. *Preprint*, arXiv:2303.12384.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, and et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Yue Pan and Limao Zhang. 2021. Roles of artificial intelligence in construction engineering and management: A critical review and future trends. *Automation in Construction*, 122:103517.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation.

Wamiq Reyaz Para, Shariq Farooq Bhat, Paul Guerrero, Tom Kelly, Niloy Mitra, Leonidas Guibas, and Peter Wonka. 2021. Sketchgen: Generating constrained cad sketches. *Preprint*, arXiv:2106.02711.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. 2022. Geometric transformer for fast and robust point cloud registration. *Preprint*, arXiv:2202.06688.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Pratyusha Sharma, Tamar Rott Shaham, Manel Baradad, Stephanie Fu, Adrian Rodriguez-Munoz, Shivam Duggal, Phillip Isola, and Antonio Torralba. 2024. A vision check-up for language models. *Preprint*, arXiv:2401.01862.

Zhuocheng Shen. 2024. Llm with tools: A survey. *Preprint*, arXiv:2409.18807.

Yuexiang Zhai Yi Ma Yann LeCun Saining Xie Shengbang Tong, Zhang Liu. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal llms.

Xiaoyi Dong Yuhang Zang Qiong Liu Tao Chu, Pan Zhang and Jiaqi Wang. 2024. Unified scene representation and reconstruction for 3d large language models. *Preprint*, arXiv:2404.13044.

Gemini Team, Rohan Anil, Sebastian Borgeaud, and et al. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Karl DD Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G Lambourne, Armando Solar-Lezama, and Wojciech Matusik. 2021. Fusion 360 gallery: A dataset and environment for programmatic cad construction from human design sequences. *ACM Transactions on Graphics (TOG)*, 40(4):1–24.

Rundi Wu, Chang Xiao, and Changxi Zheng. 2021. Deepcad: A deep generative network for computer-aided design models. *Preprint*, arXiv:2105.09492.

Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. 2024. Mind's eye of llms: Visualization-of-thought elicits spatial reasoning in large language models. *Preprint*, arXiv:2404.03622.

Conghao Xiong, Hao Chen, Joseph J.Y. Sung, and Irwin King. 2023. Diagnose like a pathologist: Transformer-enabled hierarchical attention-guided multiple instance learning for whole slide image classification. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, Main Track*, pages 1587–1595.

Yang You, Mikaela Angelina Uy, Jiaqi Han, Rahul Thomas, Haotong Zhang, Suya You, and Leonidas Guibas. 2024. Img2cad: Reverse engineering 3d cad models from images through vlm-assisted conditional factorization. *Preprint*, arXiv:2408.01437.