CAD-LLM: Large Language Model for CAD Generation

Sifan Wu [1,2] Amir Khasahmadi[2] Mor Katz[2]

Pradeep Kumar Jayaraman[2] Yewen Pu[2] Karl Willis[2] Bang Liu [1]

[1]University of Montreal [2]Autodesk AI Lab

Abstract

Parametric Computer-Aided Design (CAD) is the dominant paradigm for modern mechanical design. Training generative models to reason and generate parametric CAD can dramatically speed up design workflows. Pre-trained foundation models have shown great success in natural language processing and computer vision. The cross-domain knowledge embedded in these models holds significant potential for understanding geometry and performing complex reasoning about design. In this work, we develop generative models for CAD by leveraging pre-trained language models and apply them to manipulate engineering sketches. Our results demonstrate that models pre-trained on natural language can be fine-tuned on engineering sketches and achieve remarkable performance in various CAD generation scenarios.

1 Introduction

Parametric Computer-Aided Design (CAD) stands as the prevailing paradigm in the field of mechanical engineering for crafting modern physical objects. The creation of engineering sketches necessitates a profound understanding of geometry and often relies heavily on a substantial repository of reference CAD designs. A recent body of work has explored the generative modeling of engineering sketches, with a particular emphasis on employing transformer-based architectures[4, 5, 6]. Despite their potential, these models have struggled to capture the intricate geometric reasoning inherent in engineering sketches, rendering their application to real engineering design a challenging endeavor. Large language models, renowned for their versatile abilities in various domains and their capability to process a plethora of data forms, present a promising avenue. Given that engineering sketches can be converted into a string format representing points and their connectivity, we scrutinize the potential of these models in engieering sketch design. Our contributions are as follows:

1. We establish a comprehensive pipeline to model CAD sketches by finetuning a pre-trained foundational language model.
2. We introduce three novel evaluation metrics for CAD generative models: Entity Accuracy, Sketch Accuracy, and CAD F1 score.

2 Method

Engineering Sketch Definition. Sketches, varying from millimeters to meters, are centered at the origin within a 1-meter width bounding box to reduce ambiguity. Continuous parameters are converted to discrete variables through 6-bit uniform quantization for flexible modeling of diverse shapes. Normalization and quantization is similar to [5]. Each sketch is formatted as $\mathcal{S} = (e_1, e_2, ..., e_N)$,

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

where N is the number of entities in the sketch. Each entity e_i is represented by position parameters $(p_1, ..., p_k)$, where p_k represents the normalized coordinate parameters of the points forming the entity. We have three types of entities: line, arc, and circle. We represent them with two, three and four points respectively as in [6] allowing each sketch to be viewed as a sequence of tokens.

CAD AutoCompletion Task. We introduce a generative model to address the CAD AutoCompletion task, for automating routine CAD design procedures. The objective of this task is to complete a given partial sketch, thereby generating a fully realized design. The task can be expressed as:

$$\mathcal{L}(\Phi) = -\sum \log\Phi(\mathcal{S}|\mathcal{S}_p) = -\sum \log\Phi(p_{i:N}|p_{<i}), \quad (1)$$

where $\mathcal{S}_p = (e_1, ..., e_j)$ represents a randomly sampled sequence consisting of 20% to 80% of the entities in the complete sketch $\mathcal{S}$.

We utilize open-source natural language pretrained transformer models for CAD sketches, with a focus on the T5 models[3]. Additionally, we compare with finetuning GPT 3.5 [1]. Our experimental results indicate the presence of cross-domain knowledge transfer across each pretrained model.

Evaluation Metrics To quantitatively evaluate the CAD generative models, we propose three CAD-specific evaluation metrics: Entity Accuracy, Sketch Accuracy, and CAD F1. Sketch and Entity Accuracy show the probability of generating the correct full remaining sketch or at least one entity in the ground truth, respectively. CAD F1 is a balance between Entity Accuracy and Sketch Accuracy, calculated as:

$$\text{CAD F1} = \frac{2 \times \text{precision} \times \text{recall}}{(\text{precision} + \text{recall})}, \text{precision} = \frac{N_c}{N_p}, \text{recall} = \frac{N_c}{N},$$

where N_c represents the number of correct entities in the completed sketches, N_p is the total number of entities in the completed sketches, and N is the number of entities in the labeled sketch.

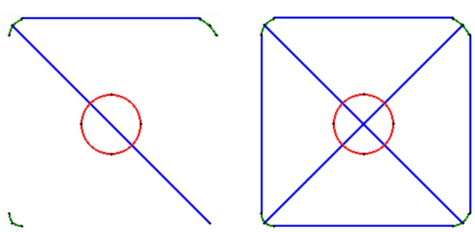

Figure 1: Example for CAD sketch. The left one is the uncompleted prefix sketch, right one is the full sketch.

Model	Entity acc	Sketch acc	CAD F1
Vitruvion	0.407	0.030	0.113
ChatGPT	0.413	0.068	0.212
CAD-LLM	**0.689**	**0.225**	**0.440**

Table 1: Performance Comparison of different models. The input prefix for all three models is randomly selected between 20% to 80% of the full sketch. Here ChatGPT is finetuned on 15% portion of the data. For all metrics, the bigger the better. The best results are shown in **bold**. More results are in the supplmentary.

3 Experiments

We use SketchGraphs[4] dataset for all our experiments and select Vitruvion as the baseline method, which is the SOTA for CAD generation tasks. The results are shown in Table 1. Limited by the high cost of ChatGPT finetuning, we only finetuned a small portion of the data (1%, 7% and 15%). And we found that performance didn't change dramatically with the increase of data or training epochs. We assume the reason is that GPT3.5 is being finetuned using PEFT (e.g. LORA)[2] methods. The T5 model, with 770m parameters, as our CAD-LLM base model, achieves the best results across all prefix ratio settings. Remarkable performance on all three metrics is achieved with training for only 20 epochs. These results demonstrate that foundational pre-trained models can indeed aid in CAD representation and generation tasks.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[3] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

[4] Ari Seff, Yaniv Ovadia, Wenda Zhou, and Ryan P Adams. Sketchgraphs: A large-scale dataset for modeling relational geometry in computer-aided design. *arXiv preprint arXiv:2007.08506*, 2020.

[5] Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P Adams. Vitruvion: A generative model of parametric cad sketches. In *International Conference on Learning Representations*, 2021.

[6] Karl DD Willis, Pradeep Kumar Jayaraman, Joseph G Lambourne, Hang Chu, and Yewen Pu. Engineering sketch generation for computer-aided design. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2105–2114, 2021.

4 Supplementary Material

Here we provide more experimental results. Covering different masking ratios and multiple fine-tuned versions of GPT3.5(i.e. ChatGPT)

Model	Prefix ratio	Entity acc	Sketch acc	CAD F1
Vitruvion	20%	0.341	0.00059	0.085
	40%	0.408	0.0071	0.110
	60%	0.414	0.035	0.119
	80%	**0.424**	**0.146**	**0.125**
	random	0.407	0.030	0.113
ChatGPT-15%	random	0.413	0.068	0.212
ChatGPT-7%		**0.438**	**0.076**	0.222
ChatGPT-1%		0.431	0.065	**0.225**
Ours	20%	0.468	0.013	0.179
	40%	0.662	0.108	0.362
	60%	0.728	0.308	0.528
	80%	**0.741**	**0.518**	**0.638**
	random	0.689	0.225	0.440

Table 2: Performance Comparison of different models and prefix ratio. For all metrics the bigger the better. Prefix ratio shows the partial input sketch to full sketch ratio