Sketch-to-CAD: A Deep Learning Approach for Predicting CAD Steps from Isometric Sketches

Ghadi Nehme [1] Benjamin Randoing [1]

[1]Department of Mechanical Engineering, Stanford University

Introduction

Computer Aided Design (CAD) is the industry standard 3D development that features a customizeable timeline with history of each step, so edits upstream will propagate through to appropriately modify the design. However, CAD design is a time intensive process that warrants sketching a preliminary 2D model. Here we show Sketch-to-CAD, an encoder-decoder model to expedite CAD development by transforming a 2D sketch into a 3D CAD workflow. Four encoder-decoder model architectures were trained using a subset of CAD 2D input sketches and output cad vector instructions from the ABC-dataset. A CNN encoder paired with a pretrained transformer decoder from an autoencoder demonstrated the best performance with command and parameter test accuracies of 93.50% and 68.30%.

Problem Statement

Revamping 3D CAD workflows by minimizing time and effort is crucial. Existing 2D to 3D generative models produce static file formats, hindering efficiency. This research introduces a novel solution: converting 2D sketches into dynamic CAD timelines, streamlining engineering workflows and saving time. To evaluate our models we look at the **number of successful CAD parts**. We also look at the **accuracy** of the predicted sequences compared to the ground truth as well as the **Mean-Squared Error (MSE)** of the CAD commands and parameters:

$$\mathcal{L} = \sum_{i=1}^{N_c} \ell(\hat{t}_i, t_i) + \beta \sum_{i=1}^{N_c} \sum_{j=1}^{N_P} \ell(\hat{\mathbf{p}}_i, \mathbf{p}_i)$$

Dataset

- We utilize a subset of the **DeepCAD Dataset**, which consists of **166,225 CAD models** [1].
- Each CAD model in the DeepCAD Dataset is represented as a sequence of steps for sequentially building the sketches and extrusions of the model.
- We collect the isometric sketches by loading the STEP files in Fusion 360.

Input

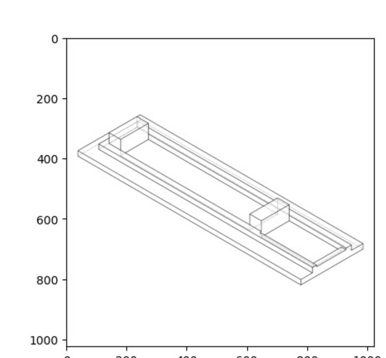


Figure 1. Example of a CAD isometric sketch

Output A sequence of 60 CAD steps where each is a 17 × 1 vector: $[t_i, x, y, \alpha, f, r, \theta, \phi, \gamma, p_x, p_y, p_z, s, e_1, e_2, b, u]$.

Commands	Parameters
⟨SOL⟩	$\emptyset$
L (Line)	x, y : line end-point
A (Arc)	x, y : arc end-point α : sweep angle f : counter-clockwise flag
R (Circle)	x, y : center r : radius
E (Extrude)	θ, ϕ, γ : sketch plane orientation p_x, p_y, p_z : sketch plane origin s : scale of associated sketch profile e_1, e_2 : extrude distances toward both sides b : boolean type, u : extrude type
⟨EOS⟩	$\emptyset$

Figure 2. CAD commands and associated parameters [1]

- We discretize the space of the parameters into 256 bins, making the problem a classification task.

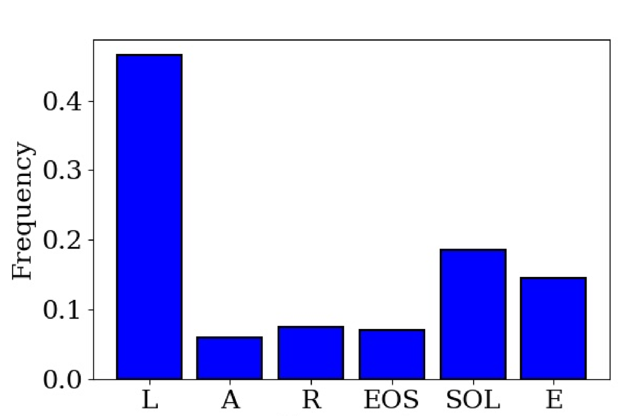


Figure 3. Frequency of the classes of each command in the dataset

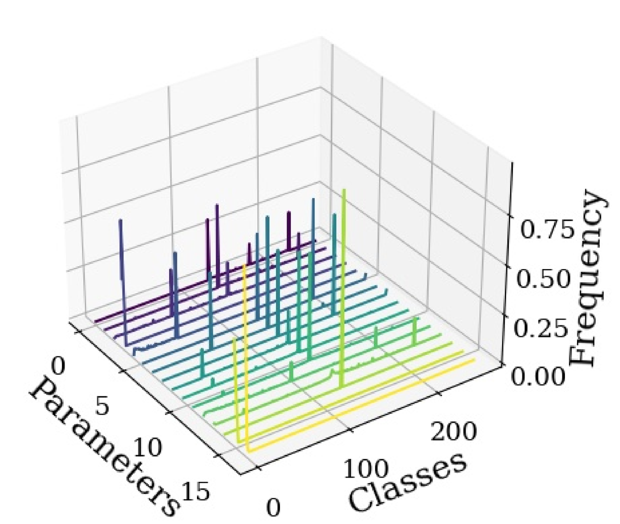


Figure 4. Frequency of the classes of each command in the dataset

Methods

For training our models, we split our data into **90%, 5% and 5%** for the training, validation and test sets, respectively. We train four encoder-decoder models using **Mean-Squared Error (MSE)**:

VGG Encoder and Transformer Decoder ▪ **CNN Encoder and LSTM Decoder** ▪ **Transformer Encoder and Transformer Decoder** ▪ **CNN Encoder and Transformer Decoder**

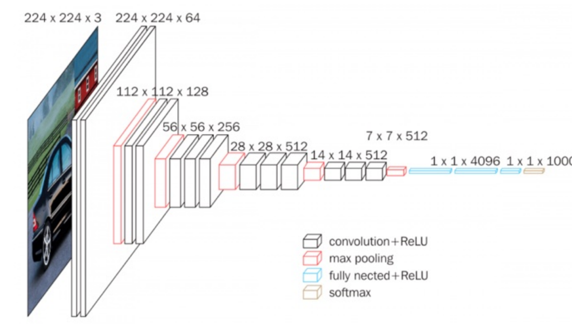


Figure 5. VGG Encoder

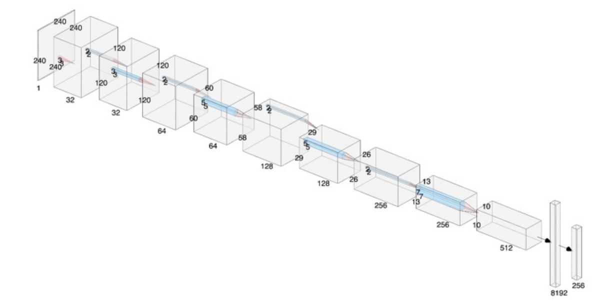


Figure 6. CNN Encoder

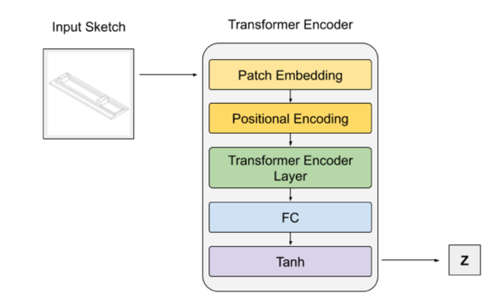


Figure 7. Transformer Encoder

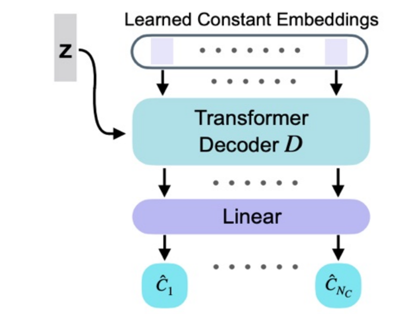


Figure 8. Transformer Decoder

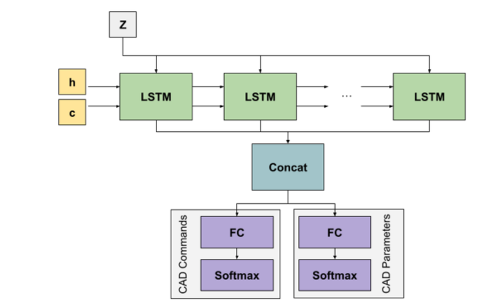


Figure 9. LSTM Decoder

Results

The models were trained for a maximum of 50 epochs, with batch size set to 256. We observe that our best model is the CNN Encoder - Transformer Decoder.

Table 1. Command and Parameter Model Accuracies

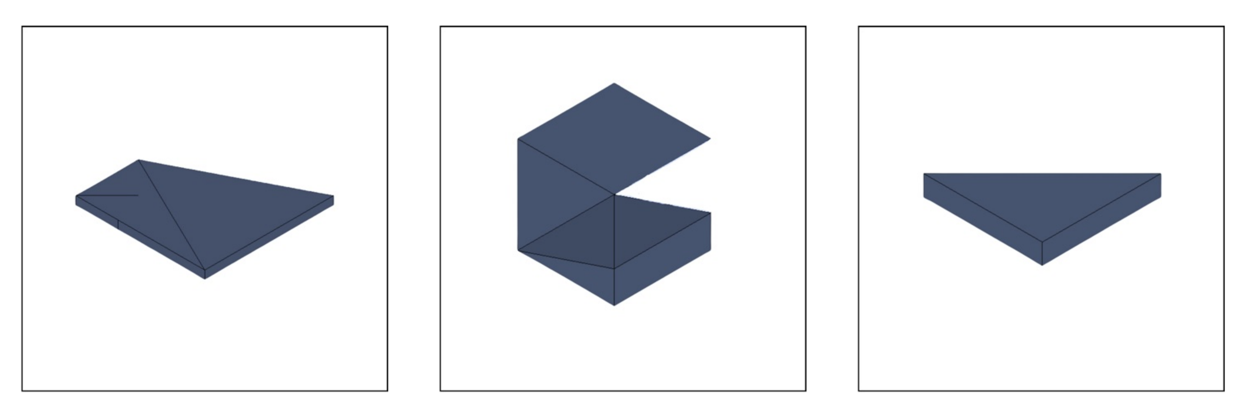
Model	Train C	Train P	Val C	Val P	Test C	Test P
VGG — Transformer	76.44	50.39	75.59	50.41	75.81	50.58
CNN — Transformer	97.04	80.72	92.30	71.33	**93.50**	**68.30**
Transformer — Transformer	91.61	71.40	91.09	69.59	90.53	64.26
CNN — LSTM	78.28	59.95	78.74	59.86	78.13	55.88

- A fully accurate CAD part exactly matches the original part used to obtain the sketch image

Table 2. Number of fully accurate CAD parts for the different models in the test dataset

Model	Fully Accurate CAD parts
VGG — Transformer	0
CNN — Transformer	109
Transformer — Transformer	19
CNN — LSTM	0

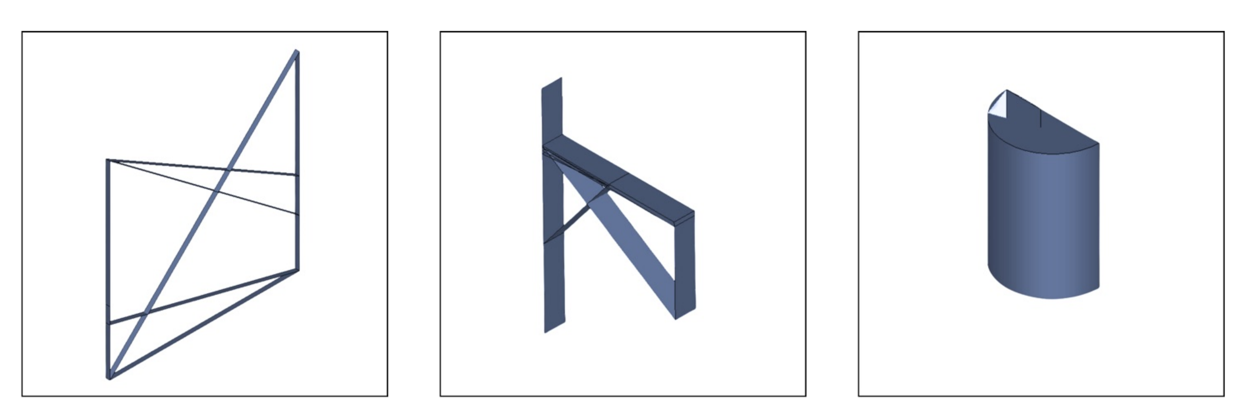
- The CNN model is only capable of generating CAD parts made of elementary elements (cylinders, cubes, triangles ...)



(a) CNN - Transformer (b) Transformer - Transformer (c) VGG - Transformer

Figure 10. Successful CAD Predictions

- Unsuccessful CAD demonstrated unaddressed open loops in sketches preventing a proper extrude



(a) CNN - Transformer (b) Transformer - Transformer (c) VGG - Transformer

Figure 11. Unsuccessful CAD Predictions

Discussion

Saliency Plot Analysis

- The saliency visualizations in Figure 12 highlight the ability of the CNN-Transformer model to recognize the attributes of the input image that correspond to a sketch.
- The CNN-LSTM and VGG-Transformer models do not demonstrate an ability to recognize the sketch portion of the input image.

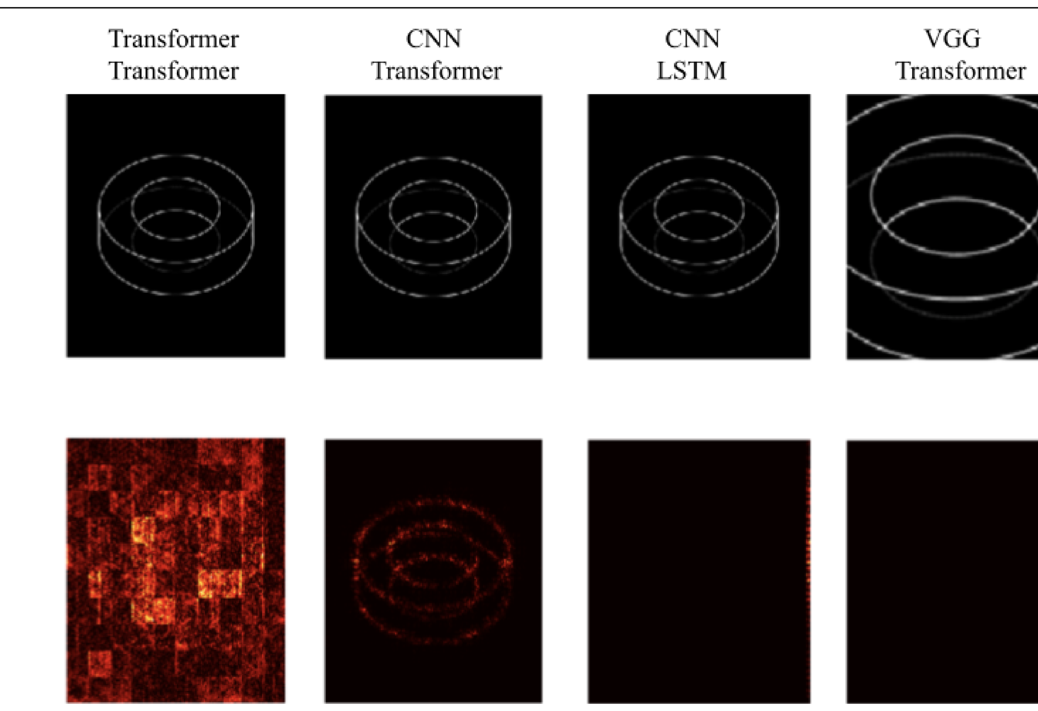


Figure 12. Comparative Saliency Visualizations

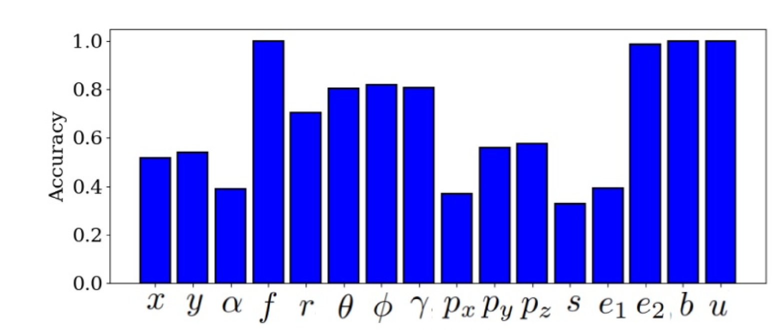


Figure 13. Parameters Accuracies for the CNN encoder - Transformer Decoder Model

Parameter Accuracies

- We see that parameters related to coordinates like the point coordinates (x, y) and the plane origin coordinates (p_x, p_y, p_z), and the parameter related to scaling s are the ones which are associated with the lowest accuracies.
- The failure we observe in our CAD parts is due to this inability of extracting coordinates of key points or the scaling of the different components by simply looking at a 2D sketch.

Future

- **Deep Reinforcement Learning:** Training a Deep RL model which, based on a current state: CAD steps, is able to take an action: CAD step and get a reward based on the new state.
- **Larger and more diverse dataset:** Collecting a larger and more diverse dataset of isometric sketches and corresponding CAD steps with more commands.
- **Fine-grained CAD representations:** Investigating alternative representations for CAD steps, such as continuous or vector-based representations.

References

[1] Rundi Wu, Chang Xiao, and Changxi Zheng. Deepcad: A deep generative network for computer-aided design models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6772–6782, 2021.

ghadi@stanford.edu

bar39@stanford.edu