

# Freehand Sketch Generation from Mechanical Components Supplementary Materials

Anonymous Authors

## 1 INITIALIZATION ANALYSIS

In this section, we will meticulously contrast and analyze our Edge-constant initialization with the original CLIPasso [11] initialization method.

As described in CLIPasso [11], it utilizes the ViT-32/B CLIP [9] to obtain the salient regions of a target image. This is achieved by averaging the attention outputs from all attention heads across each self-attention layer, generating a total of 12 attention maps. These maps are further averaged to derive the relevancy map, obtained by examining the attention between the final class embedding and all 49 patches. Subsequently, this relevancy map is combined with the edge map obtained through XDoG [12] extraction. The resulting attention map is then utilized to determine the locations for the initial strokes. In the process of determining the initial positions of strokes, CLIPasso[11] utilizes random seeds on the saliency map to sample positions for the first control point of each curve. Following this, it randomly selects the subsequent three control points of each Bezier curve within a small radius (0.05) of the initial point.

Such random initialization methods often result in the initial points of strokes being inadequately distributed around the critical features of mechanical components during sketch generation, leading to the loss of substantial modeling information. Moreover, this approach frequently results in an excessive placement of initial stroke points in certain prominent feature areas, causing confusion in generating sketch strokes and preventing accurate representation of modeling features. To address this issue, we propose the edge-constant initialization to deterministically sample. We utilize SAM [4] to perform feature segmentation on the input contour sketch. Based on the segmentation results, we predefine four stroke initialization points evenly spaced along the edge of each segmented feature. Subsequently, we dynamically change the initialization points based on the comparison with the manually required number of generated strokes. If the requested number of strokes is less than the total predefined initialization points, we evenly discard points contained within each segmented feature. Conversely, if the requested number of strokes exceeds the total predefined initialization points, we employ a greedy algorithm on the saliency map of the target image to determine additional stroke initialization points in the most salient regions [5]. This initialization method not only ensures the precise generation of mechanical component features but also optimizes the distribution of generated strokes, resulting in clearer generated sketches.

As shown in Figure 1, we conduct experiments using three sampling strategies of random seeds provided by CLIPasso[11]. It is evident that in the first instance, no stroke initialization points are placed at the three through-holes of the input flange contour sketch, resulting in the loss of this important feature in the result. In the second instance, the placement of three stroke initialization points on the upper right through-hole is unnecessary, as it is a simple feature that does not require three strokes to depict. In the third instance,

three stroke initialization points are clustered around the edge contour of the flange, while only two initialization points are placed on the structurally complex cylindrical section. These illustrate the irrational distribution of stroke initialization points caused by the random seed sampling method, ultimately leading to unsatisfactory sketch generation results. In contrast, our proposed edge-constant initialization optimizes the placement of stroke initialization points, ensuring their rational distribution on modeling feature edges. It can be observed that sketches generated through our improved method adequately preserve crucial modeling features, with a clear and rational distribution of strokes.

Method	Input	(a)	(b)	(c)	Output
CLIPasso					
Ours					

**Figure 1: Strokes Initialization.** All sketches are produced with 20 strokes. Left to right: input contour sketches, (a) the saliency maps generated from CLIP ViT activations, (b) and (c) are initial stroke locations (in red) in final distribution maps and inputs, output freehand sketches.

## 2 STABILITY ANALYSIS

In this section, we will evaluate the stability of our transformer-based [5, 6, 10] stroke generator.

In Stage-Two, our improved initialization method has enhanced the guidance sketch generator to produce informative freehand sketches. However, the guidance sketch generator employs an optimizer to create sketches through thousands of optimization iterations during sketch generation, leading to uncertainty in the outcomes. Each step of this optimization-based process is guided by CLIP [9] in terms of both semantic and geometric similarities to create strokes. This optimization process is uncontrollable and the optimized result from each step exhibits variability. It results in unstable and uncontrollable quality performance of the generated sketches. In order to consistently generate high-quality sketches,

we adopt a transformer-based [5, 6, 10] generative framework. We extract intermediate sketches from the optimization process of the guidance sketch generator as ideal guides for process sketches from each intermediate layer in the stroke generator. we utilize guidance loss during training to ensure that the stroke generator learns features from corresponding intermediate process guidance sketches. We employ CLIP-based [9] perceptual loss to ensure the similarity between the generated freehand sketches and contour sketches in both geometry and semantic information. Through training, all learned features are fixed into determined weights. During the inference phase, our model can rapidly infer freehand sketches based on the trained weights. This generation approach ensures output consistency and achieves satisfactory generation quality.

We design comparative experiments to validate the numerical stability of our generative framework. Using the same inputs, we conduct five rounds of sketch generation experiments separately with only the guidance sketch generator (GSG) and the complete generative framework (trained on the collected mechanical component dataset). As shown in Figure 2, the outcomes produced by the guidance sketch generator (GSG) for mechanical freehand sketches vary each time, and some of them exhibit suboptimal performance. For instance, in the case of the first instance, the distribution of the gear teeth slots varies significantly in each generated result, and due to the instability of the optimization-based generation method, issues arise such as chaotic stroke composition in the second round's results and erroneous connections between teeth slot strokes and through-hole strokes in the fifth round's generated sketches. Similar situations are also evident in the second and third instances. For

example, in the second instance, the distribution of continuous sections of the flat-head screws differs in each round of the experiment. And results occasionally are accompanied by contour loss such as the loss of the bottom circle of the screw in the second round of experiments, and the loss of connection at the head of the screw in the fourth round. In the third instance, involving a complex motor model, the strokes creating the main body of the motor within the area marked by the red rectangle exhibit significant variations in distribution across each experimental round. Additionally, some results accurately depict small through-holes on the motor surface, while others fail to capture this information. The reason for these issues arises from the uncontrollable nature of the optimization-based generation process. Despite our efforts to accurately position stroke initialization points on features during preprocessing, deviations in geometric and semantic guidance during optimization may result in inadequate representations of certain details in the generated sketches. In contrast, our comprehensive generation framework, after being trained on a large and diverse dataset of mechanical components, fixes learned features into weights. This ensures consistent outputs in each round of testing, and stable representations of modeling features for the components.

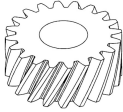










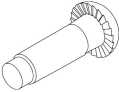




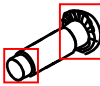





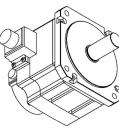










Input	Method	Round 1	Round 2	Round 3	Round 4	Round 5
	GSG					
	Ours					
	GSG					
	Ours					
	GSG					
	Ours					

Figure 2: Stability analysis for our generative framework. All sketches are produced with 20 strokes. "GSG" refers to the method of directly generating sketches through the guidance sketch generator, while "ours" represents the method of inferring sketches by our trained complete framework. The region marked by the red rectangle represents the area of significant variation in the sketches generated by the GSG.

**Table 1: Quantitative comparison results by metrics using real human-drawn sketches as standard data.**

Method	Simple				Moderate				Complex			
	FID↓	GS↓	Prec↑	Rec↑	FID↓	GS↓	Prec↑	Rec↑	FID↓	GS↓	Prec↑	Rec↑
Han et al. [3]	12.66	9.54	0.48	0.75	13.83	11.44	0.39	0.69	14.68	17.22	0.35	0.68
Manda et al. [7]	14.51	9.71	0.47	0.74	15.17	12.73	0.41	0.70	15.43	18.80	0.33	0.66
CLIPasso [11]	12.75	7.59	0.42	0.71	13.67	10.61	0.32	0.67	14.51	13.94	0.29	0.65
LBS [5]	12.40	7.53	0.43	0.73	13.19	9.24	0.30	0.65	14.03	12.60	0.28	0.63
Ours	<b>8.35</b>	<b>4.77</b>	<b>0.51</b>	<b>0.83</b>	<b>8.83</b>	<b>5.43</b>	<b>0.46</b>	<b>0.81</b>	<b>9.26</b>	<b>6.57</b>	<b>0.40</b>	<b>0.78</b>

### 3 IMPLEMENTATION DETAILS

In order to tailor our method specifically for freehand sketch generation in engineering freehand sketch modeling, we build a CAD dataset exclusively comprising mechanical components in the STEP format. We invite numerous mechanical modeling researchers to collect mechanical components from the TraceParts [1]. They are asked to encompass a diverse array of categories to enhance the inference generalization of our generative model. In the end, we obtain a Raw dataset including nearly 2,000 mechanical components. For the collected raw dataset, we employ hashing techniques for deduplication, ensuring the uniqueness of models in the dataset. Subsequently, we remove models with poor quality, which are excessively simplistic or intricate, as well as exceptionally rare instances. Following this, we classify these models based on the International Classification for Standards (ICS) [2] into 24 main categories, comprising 180 corresponding subcategories. Ultimately, we obtained a clean dataset consisting of 926 models.

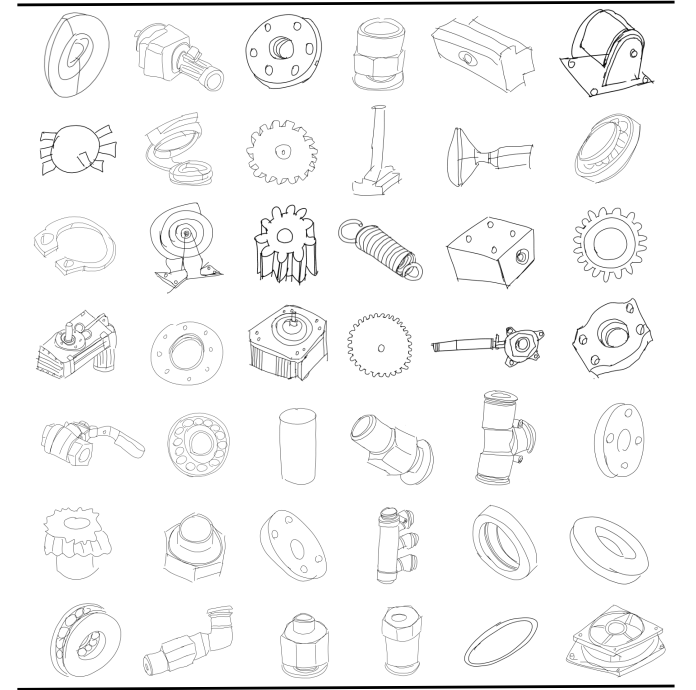
We implement the methods of Stage One using Python3 with PythonOCC and PyTorch, where PyTorch supports the viewpoint selector. For Stage Two, PyTorch and DiffVG are used to implement the model, where DiffVG is used for the differentiable rasterizer.

### 4 ADDITIONAL QUANTITATIVE EVALUATION

**Metrics Evaluation** In section 4.3 of this paper, we employ evaluation metrics for image generation to assess the quality of generated sketches. Given the absence of benchmark datasets specifically for mechanical component freehand sketches within the sketch community, we utilize component outlines processed through PythonOCC [8], which encapsulate the most comprehensive engineering information, as the standard data. The experiment results demonstrate the superiority of our method over existing freehand sketch generation methods in preserving the modeling features of mechanical components. In this section, we will conduct quantitative metric evaluations on our method and other competitors using real human-drawn sketches of mechanical components collected by ourselves.

We firstly introduce the construction process of the real human-drawn sketch dataset of mechanical component. From our collection of 926 three-dimensional mechanical component dataset, we randomly select 500 components. We invite 58 researchers with sketching expertise in the mechanical modeling domain, requesting them to draw a sketch for each component from a given perspective. We then obtain a test dataset comprising 500 mechanical sketches

drawn by human engineers. As shown in the Figure 3, we showcase the collection of real human-drawn sketches. Sketches of components drawn by researchers in the mechanical modeling domain preserve crucial modeling features which are essential for freehand sketch modeling. Correspondingly, certain minor details for modeling may be simplified, or overlooked by the researchers and not drawn. Moreover, it is evident that sketches crafted by humans exhibit a distinctive freehand style.

**Figure 3: Our collection of real human-drawn sketches for mechanical components.**

In this experiment, we utilize real human-drawn sketches of mechanical components as benchmark data, which balances maintaining crucial modeling features with exceptional freehand style well. We employ the same 500 components which are randomly selected during the construction of our real human-drawn sketches

dataset as a test dataset for this experiment. Consistent with our previous approach, we generate sketches for components using different strokes based on their complexity, and categorize the generated sketches into three levels according to the number of strokes ( $NoS$ ): Simple ( $16 \leq NoS < 24$  strokes), Moderate ( $24 \leq NoS < 32$  strokes), and Complex ( $32 \leq NoS < 40$  strokes). We continue to evaluate the generated sketches using metrics such as FID, GS, and so on. As shown in Table 1, we compare our method with methods designed for generating engineering sketches as well as methods for producing freehand sketches. It is evident that our generation method achieved the most favorable metric scores across three different levels of complexity, demonstrating the superiority of our approach in generating freehand sketches for mechanical components. In the experimental results, our outcomes obtain lower FID and GS scores and higher Prec and Rec scores. It indicates that our sketches more closely resemble real human-drawn sketches, exhibiting a higher level of consistency in preserving key modeling features and maintaining the freehand style between our results and real ones.

**Details for User Study.** In the user study conducted in this paper, we invited 47 mechanical modeling researchers to rate the generated mechanical component sketches based on two dimensions: "information" and "style." In this part, we provide detailed explanations of the specific criteria represented by these two dimensions. In the "information" dimension, we ask the researchers to evaluate the completeness of modeling features contained in the sketches. This means that the higher the number of accurate modeling features retained in the generated sketches, the higher the score obtained. In the "style" dimension, we ask the researchers to assess the overall hand-drawn style of the sketches. Specifically, they were required to consider whether the generated sketches exhibit a hand-drawn style, whether the distribution of strokes in the generated sketches is reasonable, and whether it is more similar to the distribution structure of strokes drawn by humans.

From the results of the user study, it can be observed that Han et al. [3] and Manda et al. [7] perform better in the "information" dimension. This is because their sketches are generated by contour extraction from components, nearly retaining all modeling features. However, it is worth emphasizing that, to meet the requirements of improving modeling efficiency and lowering the modeling threshold, sketches used for freehand sketch modeling should mimic human-drawn characteristics as closely as possible that preserving key modeling features while simplifying or disregarding minor ones. Therefore, although Han et al. [3] and Manda et al. [7] retain relatively comprehensive features, they fail to meet the data requirements for freehand sketch modeling and their results lack a hand-drawn style, which fundamentally does not align with the demands of the task. Meanwhile, It can be observed that our generation results outperform in preserving key features of modeling among methods for generating freehand sketches. It demonstrates the effectiveness of the modules designed in our framework to retain crucial features. In terms of "style" dimension, our sketches perform best because they exhibit a hand-drawn style while maintaining a more reasonable stroke distribution, resembling the stroke distribution habits of human drawings. Considering both dimensions, our method achieved the highest overall scores, indicating that our approach performs better than existing methods in balancing

the retention of key component modeling features and mimicking human hand-drawn style.

## 5 ADDITIONAL QUALITATIVE RESULTS

Figure 4, Figure 5, and Figure 6 show a large number of excellent freehand sketches of mechanical components generated by our method.



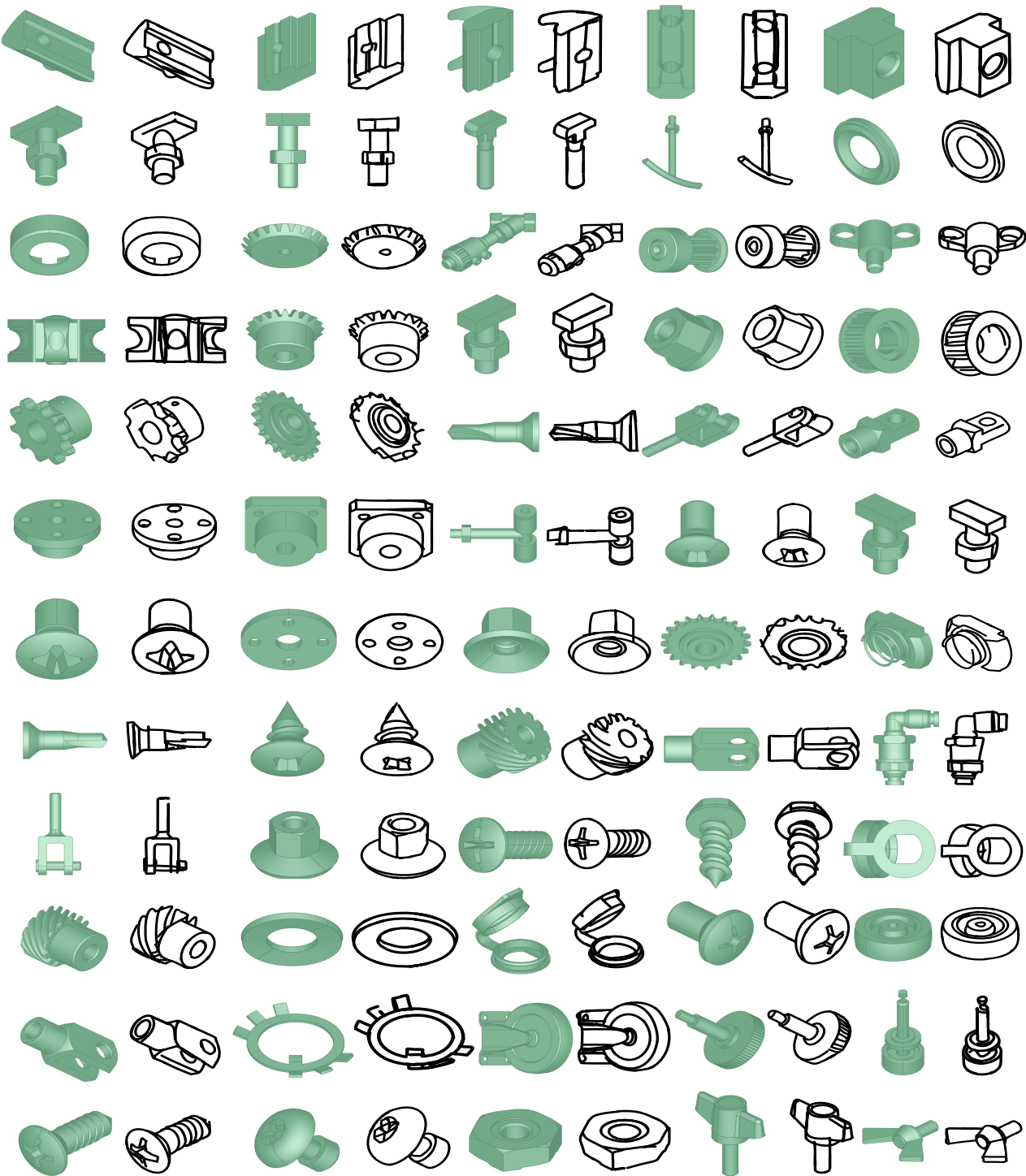


Figure 4: Robust performance across abundant categories.



Figure 5: Robust performance across abundant categories.

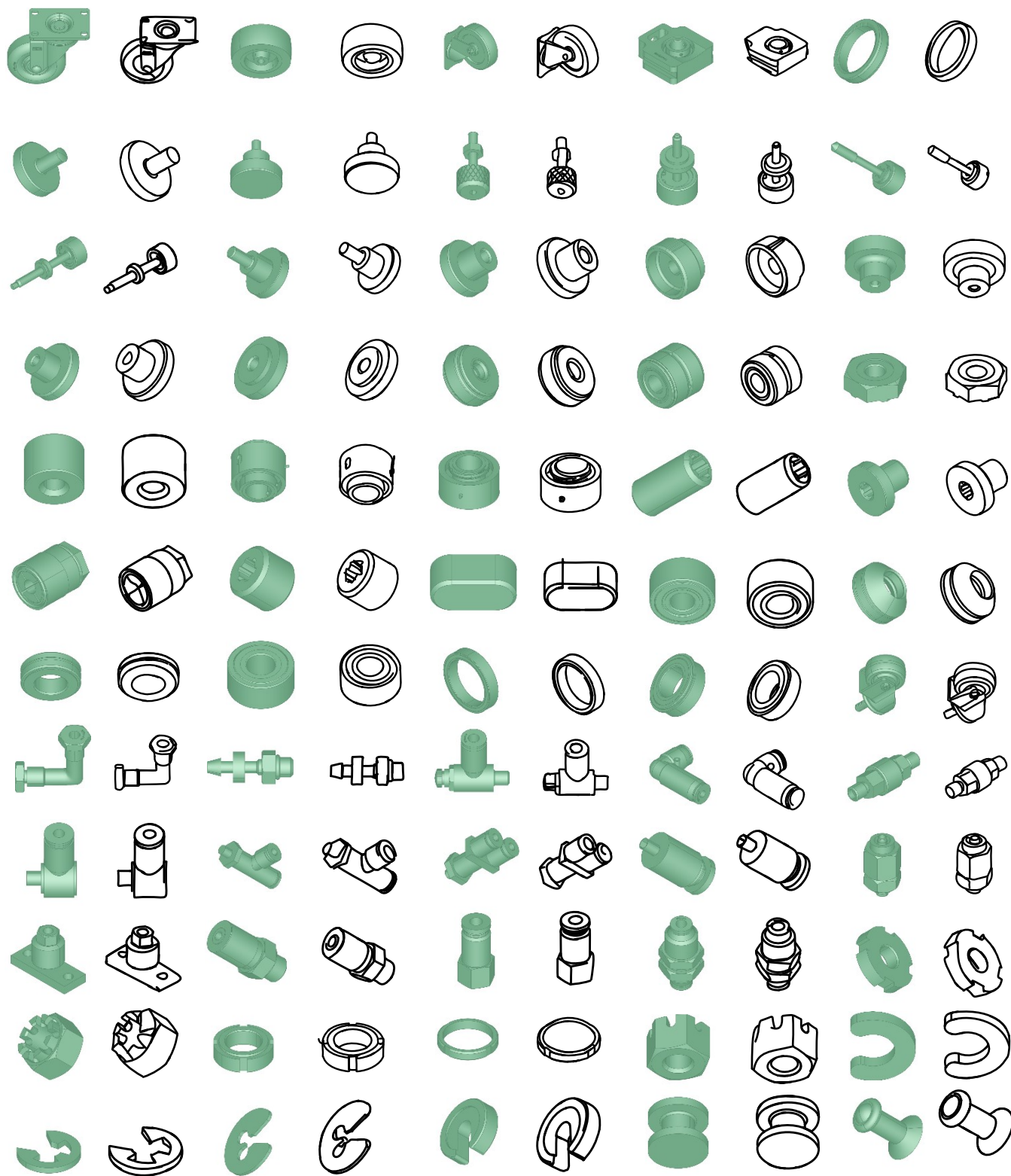


Figure 6: Robust performance across abundant categories.

## REFERENCES

- [1] 2001. TraceParts. <https://www.traceparts.com/en>.
- [2] 2015. International Classification for Standards. <https://www.iso.org/publication/PUB100033.html>.
- [3] Wenyu Han, Siyuan Xiang, Chenhui Liu, Ruoyu Wang, and Chen Feng. 2020. Spare3d: A dataset for spatial reasoning on three-view line drawings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14690–14699.
- [4] 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*. 4015–4026.
- [5] Hyundo Lee, Inwoo Hwang, Hyunsung Go, Won-Seok Choi, Kibeom Kim, and Byoung-Tak Zhang. 2023. Learning Geometry-aware Representations by Sketching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 23315–23326.
- [6] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. 2021. Paint Transformer: Feed Forward Neural Painting With Stroke Prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 6598–6607.
- [7] Bharadwaj Manda, Shubham Dhayarkar, Sai Mitheran, VK Viekash, and Ramanathan Muthuganapathy. 2021. 'CADSketchNet'-An annotated sketch dataset for 3D CAD model retrieval with deep neural networks. *Computers & Graphics* 99 (2021), 100–113.
- [8] T Paviot. 2018. pythonocc, 3d cad/cae/plm development framework for the python programming language.
- [9] 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*. PMLR, 8748–8763.
- [10] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. 2020. Sketchformer: Transformer-Based Representation for Sketched Structure. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [11] Yael Vinker, Ehsan Pajouheshgar, Jessica Y Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. 2022. Clipasso: Semantically-aware object sketching. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–11.
- [12] Holger Winnemöller, Jan Eric Kyprianidis, and Sven C Olsen. 2012. XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization. *Computers & Graphics* 36, 6 (2012), 740–753.