2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Sketch2CAD: Generative Adversarial Network for Automated Conversion of Hand-Drawn Sketches to Parametric CAD Models

Xiaogang Wang, Yuncong Liu and Yu Zhang*

***Abstract*— This paper addresses the labor-intensive process of converting imprecise hand-drawn sketches into precise, parametric CAD sketches. We present Sketch2CAD, a novel deep learning framework that leverages generative adversarial networks (GANs) to automate this conversion. Our approach consists of two main stages: first, a sketch correction module transforms freehand sketches into clean, standardized CAD-like sketches; second, a semantic segmentation module parses the generated sketches to identify and classify geometric primitives (lines, circles, arcs, points). We further introduce an optimized post-processing algorithm that extracts parametric primitives and infers geometric constraints from the segmentation results, enabling direct integration with commercial CAD software. Extensive experiments demonstrate that our method significantly outperforms state-of-the-art approaches in both primitive accuracy (94.56%) and constraint recognition. This work provides a robust solution that reduces manual effort in CAD drafting while maintaining engineering precision, particularly suitable for robotics applications requiring rapid prototyping and design iteration.**

I. Introduction

The transition from conceptual hand-drawn sketches to precise Computer-Aided Design (CAD) models represents a critical yet time-consuming phase in industrial design and development. In robotics applications ranging from manipulator design to autonomous vehicle components, the ability to rapidly convert conceptual sketches into precise, parametric models is essential for iterative design and optimization. Traditional CAD workflows require engineers to manually redraw freehand sketches using specialized software, employing appropriate CAD operations to construct three-dimensional models. This process not only introduces significant overhead in time and labor but also often suffers from inconsistencies between original design intent and final CAD implementation due to human error and interpretation differences.

The automation of sketch-to-CAD conversion presents substantial challenges that existing methods struggle to address comprehensively. First, hand-drawn sketches typically exhibit poor geometric quality, containing imprecise lines, irregular curves, and inconsistent strokes that complicate automated analysis. Second, CAD sketches comprise variable numbers of geometric primitives (lines, circles, arcs, points) with specific parametric relationships, requiring methods that can handle this structural variability while maintaining precision. Third, engineering sketches contain critical geometric constraints (parallelism, perpendicularity, connectivity) that are implicitly represented in hand-drawn versions but must be explicitly extracted for CAD model reconstruction. These challenges are particularly acute in robotics applications, where mechanical components often require precise geometric relationships for proper kinematic and dynamic performance.

Recent learning-based approaches have attempted to address aspects of this problem. Vitruvion [1] employed autoregressive networks for primitive extraction but suffered from error accumulation issues that are particularly problematic for complex robotic mechanisms. PPI-Net [2] improved upon this using detection-based approaches, yet still faced limitations in handling diverse primitive types and complex constraints commonly found in industrial designs. Instance segmentation methods, while useful in other domains, often produce incomplete results for CAD sketches due to the variable primitive counts and precise fitting requirements essential for functional robotic components.

The integration of generative adversarial networks (GANs) with geometric reasoning presents a promising direction for addressing these challenges. GANs have demonstrated remarkable capabilities in image-to-image translation tasks, particularly in domains requiring structural preservation while enhancing visual quality. However, their application to engineering sketches requires careful adaptation to maintain geometric precision and support subsequent parametric modeling—a crucial requirement for robotics applications where dimensional accuracy directly impacts system performance.

In this paper, we propose Sketch2CAD, a novel framework that addresses these limitations through an integrated generative and analytical approach specifically tailored for industrial applications. Our method leverages a modified CycleGAN architecture to transform noisy hand-drawn sketches into clean, standardized CAD-like representations while preserving the original design intent. We then perform precise semantic segmentation to identify and classify geometric primitives at the pixel level, incorporating domain-specific knowledge about robotic component design. Finally, we introduce an efficient post-processing algorithm that extracts parametric primitives and infers geometric constraints, enabling direct integration with commercial CAD systems commonly used in robotics development.

The proposed method offers several advantages for industrial design and development: (1) it significantly reduces the time required for initial CAD model generation, enabling

Xiaogang Wang and Yuncong Liu are with the College of Computer and Information Science, Southwest University, Chongqing, China.

Yu Zhang is with Beijing Freedo Technology Co., Ltd., Beijing, China, and also with the College of Computer and Information Science, Southwest University, Chongqing, China.

*Corresponding author: Yu Zhang (zhangyuhetiannan@gmail.com).

faster design iteration; (2) it maintains the designer's original intent while ensuring engineering precision; (3) it supports the creation of fully parametric models that can be easily modified for optimization studies; and (4) it provides a natural interface for conceptual design that bridges the gap between informal sketching and formal CAD modeling.

Our contributions are specifically valuable for robotics applications where rapid prototyping, design iteration, and precise geometric modeling are essential. The technical advancements include:

- A GAN-based architecture that simultaneously performs sketch correction and semantic segmentation, overcoming quality issues in hand-drawn inputs
- A robust post-processing algorithm that converts segmentation results into parameterized CAD primitives with constraint relationships
- Practical integration capabilities that bridge hand-drawn concepts with CAD modeling environments

II. Related Work

A. Sketch Processing and Analysis

The analysis of hand-drawn sketches using deep learning has seen significant advancements in recent years, with applications spanning both 2D and 3D domains [3]. Early approaches focused primarily on sketch classification and retrieval tasks. More recently, research has expanded to include sketch generation [4], segmentation [5], and reconstruction [6].

In the context of CAD sketch analysis, several notable contributions have emerged. Alaniz et al. [7] proposed a method for matching primitives from user-drawn strokes to construct coherent sketches. Cloud2Curve [8] introduced an innovative approach that takes point cloud data as input and models sketches as a series of low-dimensional parameter Bézier curves. However, these methods typically lack supervision using standard CAD ground truth data, limiting their precision for engineering applications.

Recent work also explores complementary directions such as sketch-driven mesh reconstruction, parametric sketch understanding, and perceptual feature transfer for more faithful stroke interpretation. These studies provide useful insights into robustness and representation learning for sketch-based CAD tasks, and motivate the need for stable primitive fitting and constraint extraction in our setting [9]–[13].

The SketchGraph dataset [14] marked a significant milestone by providing a large-scale collection of sketches extracted from real CAD platforms, establishing a foundation for data-driven CAD sketch analysis. Building upon this resource, Willis et al. [15] developed a two-module system that independently generates realistic sketches, though their approach struggles with complex constraint relationships.

B. CAD Reconstruction from Sketches

The problem of reconstructing parametric CAD models from freehand sketches has gained increasing attention in both computer vision and robotics communities. Vitruvion [1] employed an autoregressive network to extract primitive information from hand-drawn sketches and recover constraints between primitives. However, its sequential nature leads to error accumulation that is particularly problematic for complex robotic mechanisms.

PPI-Net [2] addressed Vitruvion's limitations by adopting a target detection-based approach to reconstruct the primitive network, significantly improving both efficiency and accuracy. Despite these improvements, PPI-Net still faces challenges in handling the diverse geometric relationships found in robotic assembly designs.

SketchConcept [16] proposed an alternative formulation that abstracts sketches into programmatic representations, where sketch concepts are treated as primitives composed into modular functions. While innovative, this approach requires significant domain knowledge and struggles with the imprecision inherent in hand-drawn sketches.

C. Generative Adversarial Networks in Geometric Processing

Generative Adversarial Networks (GANs) [17] have revolutionized image generation tasks and have been increasingly applied to geometric processing problems. Recent advancements include motion generation [18], 3D reconstruction [19], and image super-resolution [20].

In the domain of sketch processing, several GAN-based approaches have shown promising results. CircleGAN [21] introduced cycle-consistency for unpaired image-to-image translation, providing inspiration for cross-domain sketch transformation. Wang et al. [22] developed a Pooling-based Decomposition method to enhance network performance and accelerate convergence in line drawing transformations. LinkGAN [23] employed GAN training to modify latent encodings, enabling local modifications to input images for more realistic results.

More recently, UVCGAN [24] optimized the CycleGAN architecture using Transformer networks, demonstrating improved performance in various image translation tasks. However, these general-purpose GAN approaches often fail to maintain the geometric precision required for engineering applications, particularly in robotics where dimensional accuracy is critical.

III. Method

A. Overview

We aim to convert noisy hand-drawn sketches into precise, parametric CAD sketches for robotics design. As shown in Figure 1, Sketch2CAD contains three tightly coupled modules: (1) a dual-domain GAN that corrects and regularizes sketch geometry, (2) a semantic segmentation head that assigns primitive labels to pixels, and (3) a geometric post-processing pipeline that fits primitives and infers constraints.

Given a binary sketch in Domain A, the network produces two outputs: a cleaned binary CAD sketch (Domain B) and a per-pixel label map for lines, circles, arcs, and points. This joint formulation improves visual quality while preserving the parametric structure needed by CAD systems.

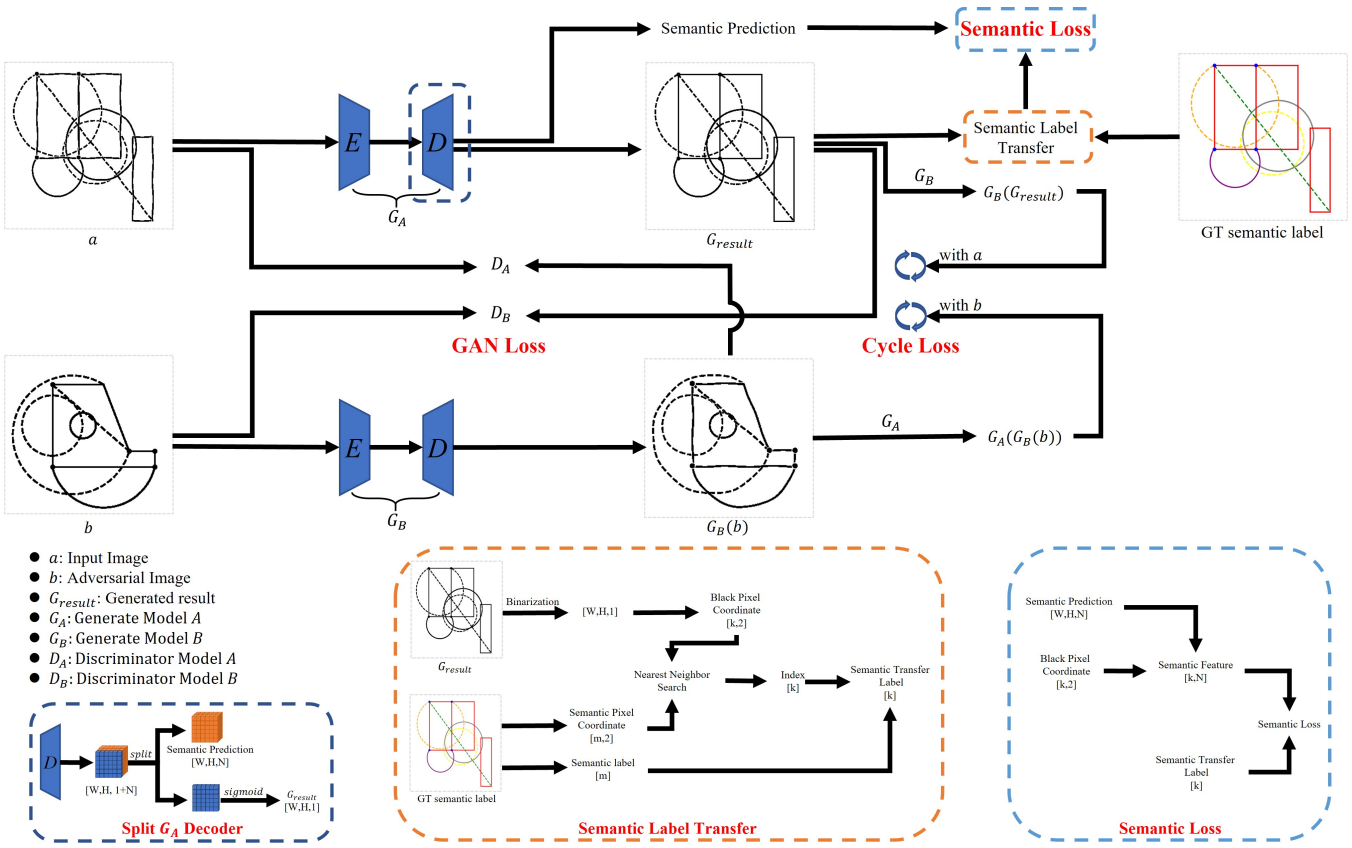

Fig. 1. Overview of the proposed Sketch2CAD framework for hand-drawn sketch correction and semantic segmentation. The network takes a hand-drawn sketch image a as input and generates a corrected CAD-quality sketch G_{result} through the generative adversarial module. Subsequently, the semantic segmentation module parses the generated sketch to identify and classify geometric primitives (lines, circles, arcs, points), enabling the post-processing algorithm to extract parametric primitives and constraints for standard CAD sketch generation.

In practice, the processing flow is: (i) normalize and binarize the input sketch, (ii) generate G_{result} and semantic logits in a single forward pass, (iii) transfer labels to the corrected sketch for cleaner boundaries, and (iv) fit parametric primitives followed by constraint inference. This end-to-end pipeline avoids separate post-hoc correction steps and reduces error propagation between stages.

B. Network Architecture

1) Generator Architecture: The generator is a modified U-Net with residual blocks. G_A maps Domain A (hand-drawn) to Domain B (CAD) using:

- **Encoder**: 8 conv layers with instance norm and ReLU, downsampling from 512×512 to 64×64
- **Bottleneck**: 6 residual blocks with dilated convs for multi-scale context
- **Decoder**: 8 transposed conv layers with skip connections
- **Split Output**: a shared decoder followed by two 1×1 conv heads:

$$F_{CAD} = \text{Conv}_{1\times1}(\text{ReLU}(\text{BN}(F_{decoder})))$$
$$F_{SEM} = \text{Conv}_{1\times1}(\text{ReLU}(\text{BN}(F_{decoder})))$$

The output generation is defined as:

$$F_{CAD}, F_{SEM} = \text{split}(G_A^D(a)), \quad G_{result} = \sigma(F_{CAD}), \quad (1)$$

where σ denotes the sigmoid activation function.

2) Discriminator Design: We use PatchGAN discriminators [25]:

- **Architecture**: 5-layer convolutional network with leaky ReLU (0.2 slope) and instance normalization
- **Receptive Field**: 70×70 patches for local feature discrimination
- **Output**: Probability map indicating patch authenticity rather than single scalar value

C. Loss Functions

1) Adversarial Loss: We adopt least squares GAN loss [26] for stable training:

$$\mathcal{L}_{G_A} = \mathbb{E}_{a\sim p_{data}(a)}[(D_B(G_A(a)) - 1)^2] \quad (2)$$
$$\mathcal{L}_{G_B} = \mathbb{E}_{b\sim p_{data}(b)}[(D_A(G_B(b)) - 1)^2] \quad (3)$$
$$\mathcal{L}_{D_A} = \mathbb{E}_{a\sim p_{data}(a)}[(D_A(a) - 1)^2] + \mathbb{E}_{b\sim p_{data}(b)}[D_A(G_B(b))^2] \quad (4)$$
$$\mathcal{L}_{D_B} = \mathbb{E}_{b\sim p_{data}(b)}[(D_B(b) - 1)^2] + \mathbb{E}_{a\sim p_{data}(a)}[D_B(G_A(a))^2] \quad (5)$$

The combined adversarial loss is:

$$\mathcal{L}_{GAN} = \frac{\mathcal{L}_{G_A} + \mathcal{L}_{G_B} + \mathcal{L}_{D_A} + \mathcal{L}_{D_B}}{4} \quad (6)$$

2) Cycle Consistency Loss: We enforce bidirectional L1 reconstruction:

$$\begin{aligned}\mathscr{L}_{Cycle} = &\mathbb{E}_{a\sim p_{data}(a)}[\|G_B(G_A(a)) - a\|_1] \\ &+ \mathbb{E}_{b\sim p_{data}(b)}[\|G_A(G_B(b)) - b\|_1]\end{aligned} \tag{7}$$

3) Semantic Segmentation Loss: The semantic segmentation pipeline includes:

1. **Binarization**:

$$B_{G_{result}}(i,j) = \begin{cases} 1 & \text{if } G_{result}(i,j) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

2. **Label Transfer**:

$$B_{xy} = \{(x,y) \mid B_{G_{result}}(x,y) = 0\} \tag{9}$$

$$GT_{xy} = \{(x,y) \mid GT(x,y) = 0\} \tag{10}$$

$$B_{SEM} = \mathrm{NN}(B_{xy}, GT_{xy}, GT_{SEM}) \tag{11}$$

where NN denotes nearest-neighbor search with KD-tree acceleration.

3. **Segmentation Loss**:

$$\mathscr{L}_{SEM} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} B^i_{SEM}(c)\log(\mathrm{softmax}(\hat{F}^i_{SEM})_c) \tag{12}$$

where $C = 4$ represents the primitive classes.

4) Geometric Regularization Loss: We add a geometric regularizer:

$$\mathscr{L}_{Geo} = \lambda_{smooth}\mathscr{L}_{smooth} + \lambda_{length}\mathscr{L}_{length} \tag{13}$$

where $\mathscr{L}_{smooth}$ encourages primitive continuity and $\mathscr{L}_{length}$ prevents degenerate primitives.

5) Full Optimization Objective: The full objective is:

$$\begin{aligned}\mathscr{L}_{total} = &\lambda_{GAN}\mathscr{L}_{GAN} + \lambda_{Cycle}\mathscr{L}_{Cycle} \\ &+ \lambda_{SEM}\mathscr{L}_{SEM} + \lambda_{Geo}\mathscr{L}_{Geo}\end{aligned} \tag{14}$$

with $\lambda_{GAN} = 1$, $\lambda_{Cycle} = 10$, $\lambda_{SEM} = 5$, $\lambda_{Geo} = 2$.

D. Parametric Primitive Extraction Pipeline

1) Primitive Instance Segmentation:

1) **Type-based Separation**: Generate binary masks for each primitive type using semantic predictions
2) **Connected Component Analysis**: Apply morphological operations to isolate individual primitive instances
3) **Farthest Point Sampling (FPS)**: Select representative points while preserving geometric features:

$$S_{fps} = \mathrm{FPS}(P,k), \quad k = \min(50, 0.1\times|P|) \tag{15}$$

4) **Graph Construction**: Build k-NN graph ($k = 8$) for spatial connectivity analysis
5) **Component Labeling**: Use Union-Find algorithm with distance thresholding

2) Geometric Fitting Algorithms: For each primitive type, we use standard fitting routines:

Lines: Weighted least squares with RANSAC:

$$\min_{m,c}\sum_i w_i(y_i - (mx_i + c))^2, \quad w_i = \exp(-d_i^2/\sigma^2) \tag{16}$$

Circles: Taubin fit with algebraic minimization:

$$\min_{x_c,y_c,r}\sum_i \left(\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r\right)^2 \tag{17}$$

Arcs: Segment-wise circle fitting with angle constraints:

$$\begin{aligned}\theta_{\text{start}} &= \arctan 2(y_1 - y_c, x_1 - x_c), \\ \theta_{\text{end}} &= \arctan 2(y_n - y_c, x_n - x_c)\end{aligned} \tag{18}$$

After fitting, we snap nearby endpoints to enforce topological consistency and remove short, isolated segments that fall below a minimum length threshold. This filtering stabilizes downstream constraint reasoning without altering the major geometry.

3) Constraint Reasoning Engine: We detect constraints in four stages:

1. **Proximity Analysis**:

$$\text{Connected} \iff \|p_i - p_j\|_2 < \tau_c \quad (\tau_c = 3\text{px}) \tag{19}$$

2. **Angular Relationships**:

$$\text{Parallel} \iff |\theta_i - \theta_j| < \tau_p \quad (\tau_p = 5^\circ) \tag{20}$$

$$\text{Perpendicular} \iff ||\theta_i - \theta_j| - 90^\circ| < \tau_v \quad (\tau_v = 5^\circ) \tag{21}$$

3. **Geometric Properties**:

$$\text{Concentric} \iff \|c_i - c_j\|_2 < \tau_{cc} \quad (\tau_{cc} = 5\text{px}) \tag{22}$$

4. **Symmetry Detection**: Hough transform-based symmetry axis identification

IV. Experiment

A. Implementation Details

The model was implemented using PyTorch 1.12 with CUDA 11.6 acceleration and trained on NVIDIA RTX 3090 GPUs with 24GB memory. We employed the Adam optimizer with $\beta_1 = 0.5$, $\beta_2 = 0.999$ and a learning rate of 2×10^{-4}, using a batch size of 4 due to the high-resolution (512×512) processing requirements. The training process required approximately 48 hours for 100 epochs, incorporating data augmentation strategy. Constraint solving was handled by a specialized geometric constraint solver Onshape based on numerical optimization techniques.

B. Dataset

We used Vitruvion [1] to filter the sketchgraph [14] dataset, including four primitives: points, lines, circles, and arcs. In Figure 2, if a sketch is missing other types of primitives, we supplemented them based on the physical composition of CAD sketches. We use the *xkcd* library in Matplotlib to generate images with a hand-drawn look and resolution of 512x512. We rendered a total of 5000 images, 95% as the training set and 5% as the test set.

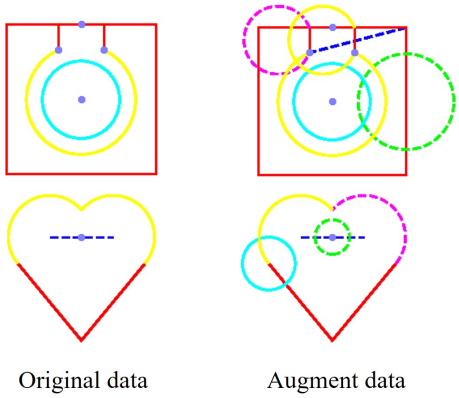

Fig. 2. Comparison of original and augmented sketch data. The left displays the original sketch from the dataset, while the right demonstrates the augmented result after primitive type completion. This augmentation strategy enriches the diversity of geometric primitives (lines, circles, arcs, points) to enhance network robustness and generalization capability for handling various CAD sketch configurations.

TABLE I

COMPARED WITH TWO STATE-OF-THE-ART CAD PRIMITIVE ANALYSIS METHODS VITRUVION [1] AND PPI-NET [2]. THE UPWARD ARROW↑ INDICATES THAT LARGER VALUES ARE PREFERABLE, WHILE A DOWNWARD ARROW↓ SUGGESTS THE OPPOSITE.

	Metrics			
	Type acc ↑	CD↓	Precision↑	Recall↑
Vitruvion [1]	86.01%	0.0436	65.33%	63.27%
PPI-Net [2]	90.01%	0.0379	68.89%	70.11%
Ours	**94.56%**	**0.0251**	**77.26%**	**81.35%**

C. Data Augmentation Strategy

To address the limited geometric diversity in the Vitruvion dataset [1], which is predominantly composed of straight-line sketches, we developed a targeted augmentation strategy to enhance primitive variety. Sketches with fewer than four lines are excluded, while eligible samples are enriched by adding missing primitives using geometrically consistent rules: circles are generated by selecting line midpoints or endpoints as centers with diameters scaled by a random factor; arcs are constructed using line endpoints as start/end points with midpoints placed along the symmetrical axis at proportional offsets; points are directly adopted from existing line endpoints.

This approach ensures geometric coherence between original and synthetic elements, significantly improving the model's ability to recognize and reconstruct complex primitive configurations and enhancing its robustness and generalization capability for real-world CAD applications.

D. Comparison with State-of-the-art

As demonstrated in Figure 3(**A**), input hand-drawn sketches are processed by generator G_A to produce standardized sketches G_{result} alongside their semantic segmentation results. To ensure a fair comparison with Vitruvion [1] and PPI-Net [2], both trained on 128×128 sketches, we resized our input resolution accordingly and fine-tuned our model. Qualitative evaluation of Figure 3(**B**) shows that our post-processing algorithm outperforms these state-of-the-art methods, which directly predict parametric primitives. Quantitative results in Table I, computed following PPI-Net's metric protocol, further confirm our superiority across all metrics (bold values indicate best performance).

Experimental Analysis. The generative capability of our GAN architecture enables the network to produce clean, regularized CAD sketches with consistent lines, circles, and arcs, as visible in Figure 3(**A**). More importantly, as evidenced in Figure 3(**B**), our method leverages semantic segmentation to accurately extract parameterized primitives and infer their geometric constraints, forming fully consistent sketches. In contrast, Vitruvion and PPI-Net require separate constraint reasoning networks, introducing additional complexity and potential error accumulation. Our end-to-end approach avoids this overhead, delivering more accurate and computationally efficient results.

TABLE II

ABLATION STUDY COMPARE THE EFFECTS OF DIFFERENT LOSS COMPOSITIONS.

Loss	Metrics		
	Per-pixel acc	IOU	Semantic acc
Cycle alone	61.36%	56.23%	—
GAN alone	83.58%	78.66%	—
GAN + Cycle	92.42%	86.65%	—
Cycle+Semantic	62.69%	56.78%	70.15%
GAN+Semantic	83.42%	78.87%	89.80%
GAN+Cycle+Semantic	**92.44%**	**86.79%**	**92.60%**

TABLE III

ABLATION STUDY COMPARE THE RESULTS OF METRICS ON ORIGINAL SKETCHES UNDER DIFFERENT TRAINING CONDITIONS.

Train Strategy		Metrics		
Augment	Semantic	Per-pixel acc	IOU	Semantic acc
×	×	90.25%	82.80%	—
✓	×	92.42%	86.65%	—
×	✓	90.36%	83.25%	85.01%
✓	✓	**92.44%**	**86.79%**	**92.60%**

E. Ablation Study

We follow CycleGAN [21] metric settings to calculate the sketch metrics generated by our network. We perform ablation experiments on the loss composition of the network in Table II. We also show the metric results of training under different strategies and testing on original hand-drawn sketches in Table III.

Experimental Analysis. Table II illustrates the qualitative variations in network metric results arising from different compositions of loss functions. The combination of loss functions directly influences the sensitivity of the network to data features, thereby affecting its overall performance. In Table III, a further exploration is conducted on the impact of data augmentation on network performance. The introduction of data augmentation leads to increased diversity in the training data, contributing to the enhancement of the network's robustness and generalization capabilities.

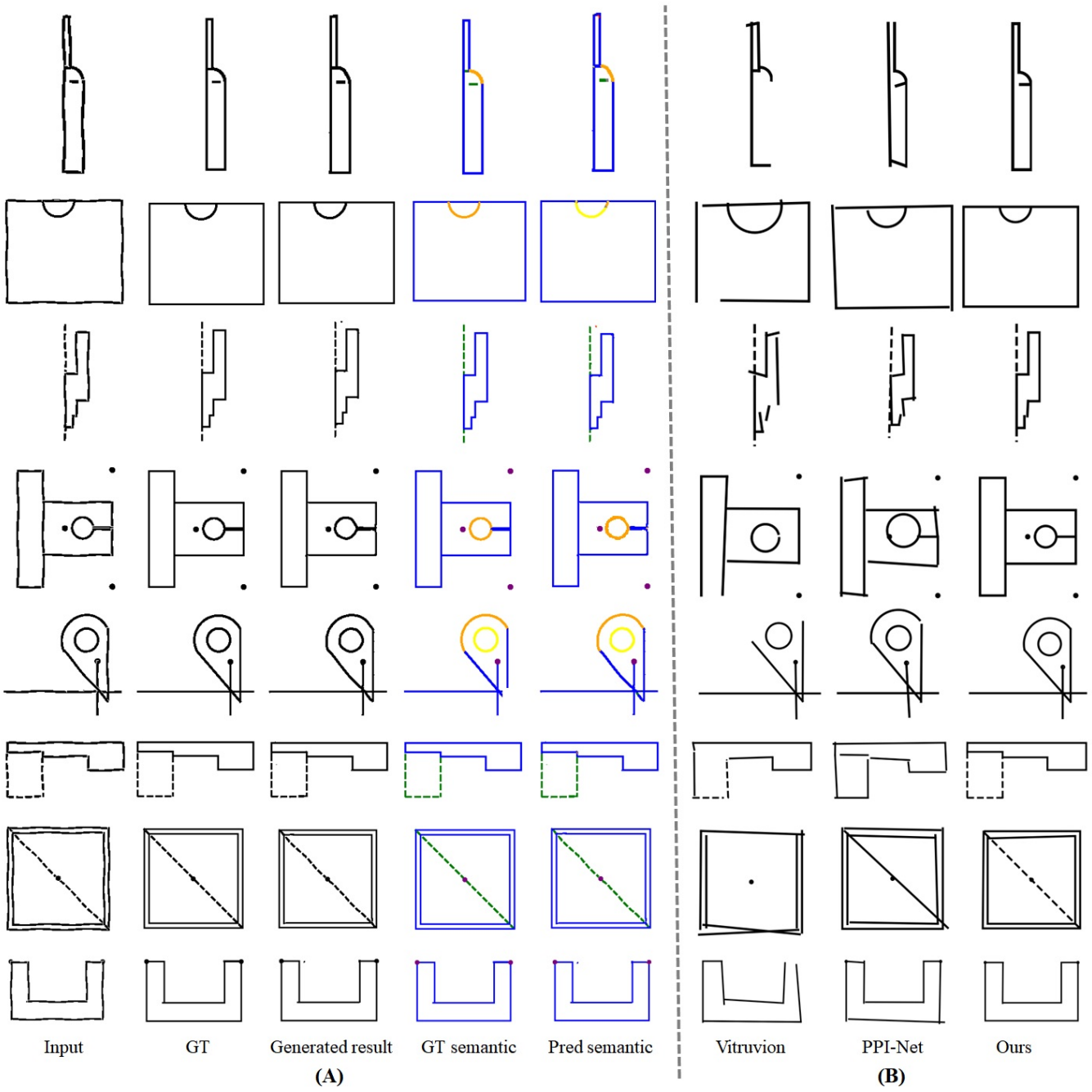

Fig. 3. **(A)** Conversion pipeline from input hand-drawn sketches to generated high-quality CAD sketches and corresponding semantic segmentation results. **(B)** Comparative analysis of parametric primitive extraction: our post-processing algorithm applied to the network's semantic predictions versus state-of-the-art methods Vitruvion [1] and PPI-Net [2], demonstrating superior performance in primitive accuracy and constraint recognition.

F. Sketch Translate to Model

As shown in Figure 4, we illustrate the process of generating high quality CAD sketches and predicting semantic segmentation from hand-drawn sketches. The predicted semantic segmentation results are processed through our post-processing algo rithm to extract parameterized primitives and infer relation ships between these primitives. Subsequently, these are im ported into CAD software to create a comprehensive CAD sketch. Finally, using CAD operations, the comprehensive sketch is transformed into the definitive 3D CAD model. The image provides multiple examples, highlighting the applica bility of our approach in the CAD domain.

V. Conclusion

In this paper, we have presented Sketch2CAD, a novel deep learning framework that effectively addresses the challenging problem of converting freehand sketches into precise,

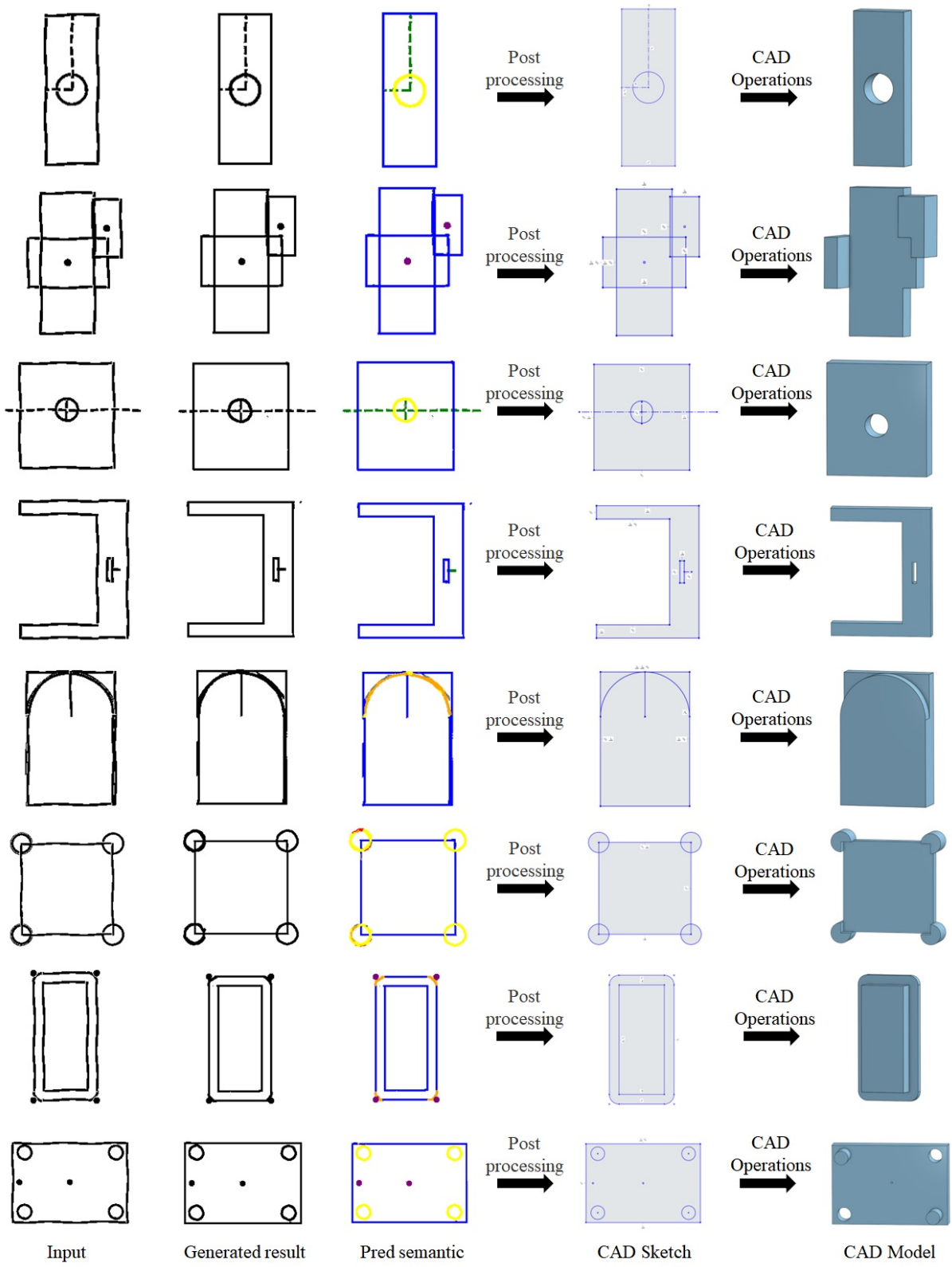

Fig. 4. Qualitative demonstration of the complete Sketch2CAD pipeline. From left to right: input hand-drawn sketches; generated high-quality CAD sketches; predicted semantic segmentation results distinguishing primitive types (lines, circles, arcs, points); parametric primitive extraction and constraint inference through our post-processing algorithm; reconstructed standard CAD sketch with geometric constraints; and finally, the generated 3D CAD model through subsequent CAD operations.

parametric CAD models. Our approach integrates generative adversarial networks for sketch correction, semantic segmentation for primitive identification, and a robust post-processing pipeline for geometric constraint extraction. This comprehensive methodology successfully bridges the gap between conceptual hand-drawn designs and production-ready CAD models.

The key innovations of our work include: (1) a dual-domain GAN architecture that simultaneously enhances sketch quality and performs semantic segmentation; (2) an efficient post-processing algorithm that extracts parametric primitives and infers geometric constraints with high accuracy; and (3) a complete workflow that enables direct integration with commercial CAD systems. Extensive experimental results demonstrate that our method significantly outperforms state-of-the-art approaches in both primitive recognition accuracy (94.56%) and constraint extraction performance.

While our method shows promising results, several directions warrant further investigation. Future work will focus on extending the framework to handle more complex geometric constraints, supporting additional primitive types, and improving robustness to highly noisy input sketches. We also plan to explore real-time applications and mobile implementation for on-site design assistance. In addition, because the current evaluation is limited by available training and test data, we will collect more real-world sketches and conduct broader user studies to validate performance and usability in practical workflows. The proposed approach represents a significant step toward intelligent design automation, with potential applications extending beyond CAD to various domains requiring geometric understanding from visual inputs.

REFERENCES

[1] Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P Adams, "Vitruvion: A generative model of parametric cad sketches," *arXiv preprint arXiv:2109.14124*, 2021.

[2] Liang Wang and Xiaogang Wang, "Ppi-net: End-to-end parametric primitive inference," in *Computer Graphics International Conference*. Springer, 2023, pp. 67–78.

[3] Peng Xu, Timothy M Hospedales, Qiyue Yin, Yi-Zhe Song, Tao Xiang, and Liang Wang, "Deep learning for free-hand sketch: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 285–312, 2022.

[4] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales, "Learning deep sketch abstraction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8014–8023.

[5] Peng Xu, Yongye Huang, Tongtong Yuan, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, Zhanyu Ma, and Jun Guo, "Sketchmate: Deep hashing for million-scale human sketch retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8090–8098.

[6] Lan Yang, Kaiyue Pang, Honggang Zhang, and Yi-Zhe Song, "Sketchaa: Abstract representation for abstract sketches," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10097–10106.

[7] Stephan Alaniz, Massimiliano Mancini, Anjan Dutta, Diego Marcos, and Zeynep Akata, "Abstracting sketches through simple primitives," in *European Conference on Computer Vision*. Springer, 2022, pp. 396–412.

[8] Ayan Das, Yongxin Yang, Timothy M. Hospedales, Tao Xiang, and Yi-Zhe Song, "Cloud2curve: Generation and vectorization of parametric sketches," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 7088–7097.

[9] Benoit Guillard, Edoardo Remelli, Pierre Yvernay, and Pascal Fua, "Sketch2mesh: Reconstructing and editing 3d shapes from sketches," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13023–13032.

[10] Xiaogang Wang, Liang Wang, Hongyu Wu, Guoqiang Xiao, and Kai Xu, "Parametric primitive analysis of cad sketches with vision transformer," *arXiv preprint arXiv:2407.00410*, 2024.

[11] Ahmet Serdar Karadeniz, Dimitrios Mallis, Nesryne Mejri, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada, "Picasso: A feed-forward framework for parametric inference of cad sketches via rendering self-supervision," *arXiv preprint arXiv:2407.13394*, 2024.

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 2016, pp. 694–711.

[13] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] 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.

[15] Karl D.D. 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 (CVPR) Workshops*, June 2021, pp. 2105–2114.

[16] Yuezhi Yang and Hao Pan, "Discovering design concepts for cad sketches," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28803–28814, 2022.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, Eds. 2014, vol. 27, Curran Associates, Inc.

[18] Liang Xu, Ziyang Song, Dongliang Wang, Jing Su, Zhicheng Fang, Chenjing Ding, Weihao Gan, Yichao Yan, Xin Jin, Xiaokang Yang, et al., "Actformer: A gan-based transformer towards general action-conditioned 3d human motion generation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2228–2238.

[19] Fei Yin, Yong Zhang, Xuan Wang, Tengfei Wang, Xiaoyu Li, Yuan Gong, Yanbo Fan, Xiaodong Cun, Ying Shan, Cengiz Oztireli, et al., "3d gan inversion with facial symmetry prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 342–351.

[20] JoonKyu Park, Sanghyun Son, and Kyoung Mu Lee, "Content-aware local gan for photo-realistic super-resolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 10585–10594.

[21] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[22] Yinhuai Wang, Yujie Hu, Jiwen Yu, and Jian Zhang, "Gan prior based null-space learning for consistent super-resolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023, vol. 37, pp. 2724–2732.

[23] Jiapeng Zhu, Ceyuan Yang, Yujun Shen, Zifan Shi, Bo Dai, Deli Zhao, and Qifeng Chen, "Linkgan: Linking gan latents to pixels for controllable image synthesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 7656–7666.

[24] Dmitrii Torbunov, Yi Huang, Haiwang Yu, Jin Huang, Shinjae Yoo, Meifeng Lin, Brett Viren, and Yihui Ren, "Uvcgan: Unet vision transformer cycle-consistent gan for unpaired image-to-image translation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 702–712.

[25] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[26] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2794–2802.