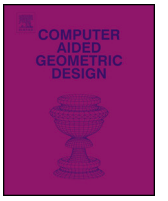
Comput. Aided Geom. Des. 111 (2024) 102327

Contents lists available at ScienceDirect

Computer Aided Geometric Design

journal homepage: www.elsevier.com/locate/cagd



VQ-CAD: Computer-Aided Design model generation with vector quantized diffusion

Hanxiao Wang [a,b], Mingyang Zhao [c,*], Yiqun Wang [d], Weize Quan [a,b,**], Dong-Ming Yan [a,b]

[a] MAIS, Institute of Automation, Chinese Academy of Sciences, China
[b] University of Chinese Academy of Sciences, China
[c] Hong Kong Institute of Science & Innovation, Chinese Academy of Sciences, China
[d] College of Computer Science, Chongqing University, China

ARTICLE INFO

Keywords:
Diffusion model
Computer-aided design
CLIP
Vector quantization
Representation learning

ABSTRACT

Computer-Aided Design (CAD) software remains a pivotal tool in modern engineering and manufacturing, driving the design of a diverse range of products. In this work, we introduce VQ-CAD, the first CAD generation model based on *Denoising Diffusion Probabilistic Models*. This model utilizes a vector quantized diffusion model, employing multiple hierarchical codebooks generated through VQ-VAE. This integration not only offers a novel perspective on CAD model generation but also achieves state-of-the-art performance in 3D CAD model creation in a fully automatic fashion. Our model is able to recognize and incorporate implicit design constraints by simply forgoing traditional data augmentation. Furthermore, by melding our approach with CLIP, we significantly simplify the existing design process, directly generate CAD command sequences from initial design concepts represented by *text* or *sketches*, capture design intentions, and ensure designs adhere to implicit constraints.

1. Introduction

Computer-Aided Design (CAD) techniques continue to be a fundamental element in the engineering and manufacturing sectors today. Underpinning the design of everything from the most rudimentary household items to sophisticated aircraft (Ikubanni et al., 2022; Shahin, 2008; Camba et al., 2016). The power of CAD lies in its ability to offer an intricate design platform that melds efficiency with precision. Typically, designers employ a "sketch-and-extrude" technique (Oh et al., 2006; Ren et al., 2022; Li et al., 2023) for CAD model creation, starting with 2D curves to define contours and then transforming them into 3D shapes. The culmination of these stages results in detailed CAD models.

While "sketch-and-extrude" is intuitive and efficient, it inevitably has various limitations. In intricate CAD designs, discrepancies often manifest when parameters fail to align with the set of requirements. These deep-rooted issues are not only elusive but also require significant time and resources for rectification. Despite the progress that has been made in CAD model generation through deep learning techniques (Wu et al., 2021; Willis et al., 2021; Xu et al., 2022; Lambourne et al., 2022; Xu et al., 2023), there remains a conspicuous oversight in addressing the automation of error correction, which is a challenge central to our research.

* Corresponding author at: Hong Kong Institute of Science & Innovation, Chinese Academy of Sciences, China.
** Corresponding author at: MAIS, Institute of Automation, Chinese Academy of Sciences, China.
E-mail addresses: migyangz@gmail.com (M. Zhao), qweizework@gmail.com (W. Quan).

https://doi.org/10.1016/j.cagd.2024.102327

Available online 30 April 2024

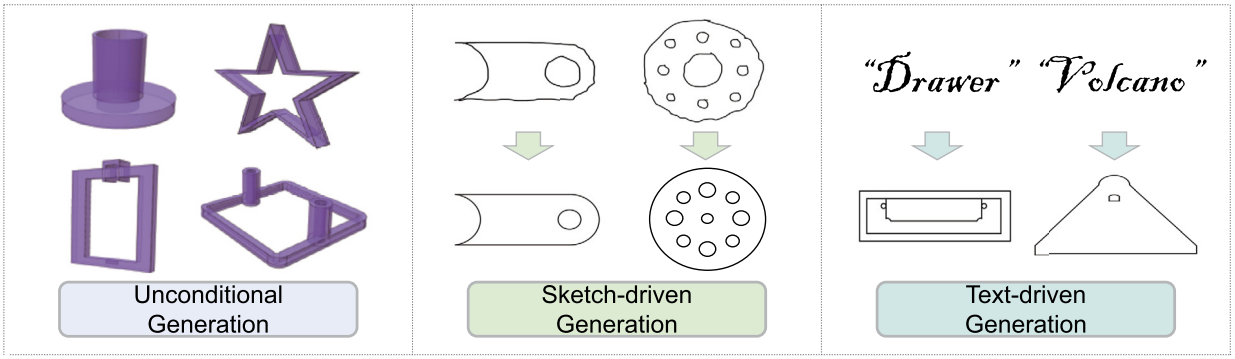


Fig. 1. Our VQ-CAD model is capable of accomplishing various types of generation tasks. The illustration on the left demonstrates the generation of unconditional CAD sequences. In the middle and right are examples of sketch-driven and text-driven generation of CAD sequences, respectively, with the top row showing inputs and the bottom row showing the corresponding outputs.

Studies like (Para et al., 2021; Ganin et al., 2021; Seff et al., 2021) probe into the display constraints intrinsic to sketch generation, however, their reliance predominantly leans towards explicit constraints. We delve deeper, analyzing from a theoretical perspective and validating through experiments, the potential reasons why prior works may generate sketches that violate these constraints. Our solution is straightforward yet highly effective, *i.e.*, we give up data augmentation during the transformer training stage. A hallmark of our approach is its adeptness at autonomously generating commands without resorting to any explicit constraints, ensuring natural conformity to core design principles.

Building on this foundation, we aim to use *Denoising Diffusion Probabilistic Models* (DDPM), which has been proven effective in many fields (Ho et al., 2020; Lugmayr et al., 2022), to enhance the quality of the generated results. However, applying diffusion to CAD commands directly presents challenges. Inspired by *Vector Quantized Diffusion* (VQ-Diffusion) (Gu et al., 2022) model, we incorporate vector quantization in conjunction with hierarchical representation (Xu et al., 2023) to facilitate discrete diffusion within the latent space.

Furthermore, we have improved our system by leveraging *Contrastive Language-Image Pre-Training* (CLIP) (Radford et al., 2021). This integration permits a seamless transition from image sketches and text descriptions directly to CAD sequences, ensuring a level of implicit regularity, as shown in Fig. 1. By utilizing CLIP's capabilities, we not only open doors for a more intuitive design process, but also facilitate a robust mechanism for error detection and correction. This fusion of traditional CAD design with CLIP's advanced methodologies offers a glimpse into the future of CAD design, where systems do not just follow commands but actively learn, adapt, and even correct. This represents not just an advancement for design professionals, but also a meaningful progression in CAD research, transitioning from fixed manual constraints to a more adaptive, learning-focused design approach.

The main contributions are summarized as follows:

- We introduce VQ-CAD, a novel and effective approach for CAD model generation by utilizing VQ-Diffusion.
- Our model shows a unique ability to discern implicit design constraints, paving the way for potential automated error rectification mechanisms in CAD design.
- We further enhance the CAD model generation system by incorporating the strengths of CLIP, facilitating a seamless transition directly from image sketches and text descriptions to CAD sequences.

2. Related work

2.1. Constructive solid geometry learning

Constructive Solid Geometry (CSG) is a technique for creating complex 3D models by applying Boolean operations on geometry primitives. Despite its wide applications in like mechanical manufacturing and architectural design, the process can be complex. Recently, deep learning techniques have been employed to study the process of generating 3D models using CSG. The first such method is CSGNet (Sharma et al., 2018), which employs reinforcement learning to identify CSG commands that minimize reconstruction error. UCSGNet (Kania et al., 2020) provided a larger solution space by performing multiple Boolean operations on the generated primitives. CSG-Stump (Ren et al., 2021) optimized the solution space, outputting CSG programs composed of unions of intersections of primitives or their complements. Similarly, CAPRI-Net and D^2CSG (Yu et al., 2022, 2023) transformed original quadratic surfaces into convex bodies through intersection and defined two types of shapes by uniting these bodies, with the final output being the difference between these shapes. These pioneering works have opened up new avenues for reconstructing generative commands for 3D shapes through deep learning methods.

2.2. Sketch and extrude CAD generation

In terms of 3D model reconstruction, Point2Cyl (Uy et al., 2022) is an innovative supervised network that effectively transformed 3D point clouds into a set of extrusion cylinders. ExtrudeNet (Ren et al., 2022) and SECAD-Net (Li et al., 2023) extended the CAPRI-Net (Yu et al., 2022) architecture by replacing convex bodies created by intersecting quadratic surfaces with extruded 2D sketches while maintaining the end-to-end differentiability. Sketch2CAD, CAD2Sketch, and Free2CAD (Li et al., 2020, 2022; Hähnlein et al., 2022) learned to convert between hand drawings and CAD commands. DeepCAD (Wu et al., 2021) is a forerunner in 3D model generation by translating shapes into sequences of CAD operations with a transformer-based generative network, which shows its efficacy in both shape auto encoding and random shape generation. SkexGen (Xu et al., 2022) deployed auto-regressive generative models and distinct transformer architectures for CAD construction sequences, allowing efficient design space exploration. It improved the user control and the production of diverse CAD models. Additionally, HNC-CAD (Xu et al., 2023) employed a hierarchical tree of neural codes for high-level design concepts, excelling in tasks like unconditional generation and enhancing conditional generation through code tree manipulations. Compared to these methods, we can generate 3D models that better satisfy implicit design constraints and our model has the ability to generate CAD construction sequences from prompts.

2.3. Discrete diffusion models

DDPM have been recognized for their effectiveness in generative modeling, particularly in image generation and restoration (Ho et al., 2020; Saharia et al., 2022; Lugmayr et al., 2022). The advent of Latent Diffusion Models (LDM) extended DDPM to the latent space, expanded the capabilities of DDPM into the latent space, yielding promising results across various domains, including high-resolution image synthesis and brain imaging (Vahdat et al., 2021; Rombach et al., 2022; Vahdat et al., 2022). When confronted with discrete data like text, DDPM faces more challenges. The foundation of diffusion models for discrete state spaces can be traced back to the work by Sohl-Dickstein et al. (2015), which introduced a diffusion process for binary random variables. This concept was later expanded to categorical random variables with uniform transition probabilities by Hoogeboom et al. (2021). Building on these concepts, D3PM (Austin et al., 2021) devised a comprehensive framework for diffusion processes involving categorical random variables, which have been applied in tasks such as text-to-sound generation (Yang et al., 2023). Inspired by this, VQ-Diffusion Gu et al. (2022), a blend of VQ-VAE (Van Den Oord et al., 2017) and conditional DDPM, has shown potential in text-to-image generation, with its “mask-and-replace” approach effectively reducing prediction errors. In addition to text-to-image generation, Inoue et al. (2023) has validated the effectiveness of VQ-Diffusion in the field of layout generation.

2.4. Text-to-shape generation

The recent surge in technological advancements has brought the generation of text-to-shape to the forefront. However, the lack of sufficient text-to-shape datasets poses a significant challenge, confining the applications of supervised generation techniques to limited categories. CLIP-Draw (Frans et al., 2022) circumvented this limitation by marrying differentiable rendering with CLIP for the synthesis of text-to-drawn. Similarly, numerous studies (Jain et al., 2022; Mohammad Khalid et al., 2022; Poole et al., 2022; Wang et al., 2023) have leveraged the image and text embeddings provided by CLIP (Radford et al., 2021), utilizing rendering techniques to generate 3D models. Deviation from this trajectory, CLIP-Forge and CLIP-Sculptor (Sanghi et al., 2022, 2023a) utilize CLIP for conditional generation in the latent space, advancing text-driven zero-shot 3D generation. Similarly, Sketch-a-Shape (Sanghi et al., 2023b) employs a pre-trained image encoder and a masked transformer to transform sketches into various 3D shape representations.

Our approach distinguishes itself by seamlessly intertwining *CLIP, VQ-Diffusion and hierarchical code tree*. This integration facilitates the conversion of images, sketches, and textual directives into CAD command sequences, satisfied with the inherent regularization constraints. This emerging field holds huge potential, especially in areas such as design, art, and content generation. Moreover, the ability to automatically extract CAD instructions from mere textual descriptions or rudimentary sketches can drastically enhance operational efficiency.

3. Methodology

3.1. CAD sequence representation details

In the sketch-and-extrude model, multiple extruded sketches combine to form a composite design, with extrude parameters detailed in Table 1. The extrusion representation in our model is characterized by six parameters, summing up to ten individual settings. S-Offset refers to the deviation of the sketch plane from the origin, while S-Scaling denotes the scaling factor applied to alter its dimensions. Extrusion signifies the magnitude of extrusion executed on either side of the sketch plane. The Rotation parameter embodies the rotational aspect of the extruded body, and in this work, we employ rotation matrices solely with entries of 0, 1, or -1, aggregating to 26 unique matrices, which can conveniently be represented by a single parameter. Transition indicates the offset of the extruded body relative to the origin. Finally, Boolean Operation specifies the operation type, *i.e.*, intersection, union, or subtraction. The aforementioned parameters fully determine an extrusion. Each sketch is an amalgamation of various faces, with each face being structured by multiple loops. Diving deeper, each loop is defined by a series of curves, which could manifest as a line, an arc, or a circle, comprising 2, 3, or 4 points, respectively. Each point within these structures is denoted by a geometry token,

Table 1
Parameters for representing extrusion.

Description	S-Offset	S-Scaling	Extrusion	Rotation	Transition	Bool Operation
Parameters	x, y	S	E_{up}, E_{down}	R	O_x, O_y, O_z	B

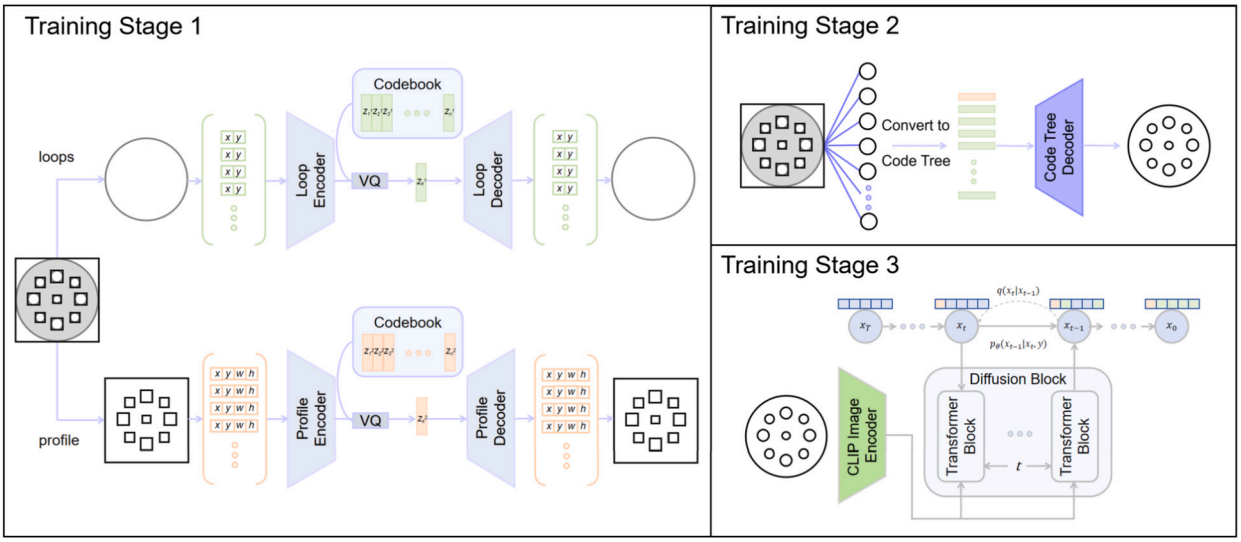


Fig. 2. The illustration of the three-stage training process for transforming 2D sketches into CAD models of the proposed VQ-CAD method. In the first stage, separate VQ-VAEs are trained for "loop" and "profile" elements, generating individual codebook encodings for each object. Then, sketches are converted into corresponding code tree, which are then paired with their respective CAD commands to train the code-tree decoder. Finally, a frozen CLIP image encoder is employed to encode the sketch images, providing conditional guidance for diffusion.

while end-primitive token is employed to signify the termination of each primitive entity. When a face has multiple loops, the first outlines the external boundary, while the rest defines internal holes.

Besides the representation of sketch-and-extrude sequences, we also adopt a hierarchical structure containing three main levels: *loops, profiles, and solids* (Xu et al., 2023). At the foundation of this hierarchy is the loop, which is a basic unit composed of interconnected lines, arcs, or circles, defined by sequences of 2D coordinates. These coordinates are separated by specific tokens, with curves arranged counterclockwise based on their initial coordinates. Moving up the hierarchy, the profile level captures the geometry of the loop, defined by 2D bounding box parameters, organized by the bottom-left corners of these boxes. This representation is sufficient for a 2D CAD sequence. With the addition of an extrusion operation, the hierarchy extends to the solid level, representing the 3D structure created by extruding one or more profiles. This level is defined by 3D bounding box parameters of the extruded profiles, also organized by their bottom-left corners.

3.2. VQ-CAD

When directly conducting the diffusion process on CAD sequences, we found that this naive method produces limited generation performance. A possible reason is that the length of the CAD data sequence is too long to directly model. Inspired by latent diffusion models, we attempted to conduct diffusion in the latent space. With the help of the hierarchical representation, CAD command sequences are converted into a code tree, where each code is looked up from codebooks of VQ-VAE trained specifically for loops, profiles, and solids. This approach enables us to perform discrete diffusion in this discrete latent space. Fig. 2 outlines our proposed VQ-CAD method with a three-stage training process. In the codebook extraction phase, we use the same version of VQ-VAE as HNC-CAD with Masked Autoencoder (He et al., 2022). While developing the code tree decoder, we found that data augmentation disrupts the implicit regularity inherent in the data. This observation has significant implications for our method, which will be further discussed in Section 4.2. Next, we briefly introduce the VQ-Diffusion module.

3.2.1. Discrete diffusion models

To carry out a discrete diffusion process on the code tree, we adopt the approach of VQ-Diffusion. Specifically, for discrete random variables that cover K categories, denoted as $x_t, x_{t-1} \in \{1, \ldots, K\}$, the forward transition probabilities can be expressed as:

$$\left[\boldsymbol{Q}_t\right]_{ji} = q(x_t = j | x_{t-1} = i). \tag{1}$$

Given the one-hot representation of x, denoted as row vector $\boldsymbol{x}$, the relationship is:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \boldsymbol{x}_t^\top \boldsymbol{Q}_t \boldsymbol{x}_{t-1}. \tag{2}$$

Starting from $\boldsymbol{x}_0$, the t-step marginal and the posterior at time $t-1$ are:

$$q\left(\boldsymbol{x}_t|\boldsymbol{x}_0\right) = \boldsymbol{x}_t^\top \overline{\boldsymbol{Q}}_t \boldsymbol{x}_0, \quad \text{where } \overline{\boldsymbol{Q}}_t = \boldsymbol{Q}_1 \boldsymbol{Q}_2 \ldots \boldsymbol{Q}_t, \tag{3}$$

$$\begin{aligned} q\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0\right) &= \frac{q\left(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{x}_0\right) q\left(\boldsymbol{x}_{t-1}|\boldsymbol{x}_0\right)}{q\left(\boldsymbol{x}_t|\boldsymbol{x}_0\right)} \\ &= \frac{\left(\boldsymbol{x}_t^\top \mathbf{Q}_t \boldsymbol{x}_{t-1}\right)\left(\boldsymbol{x}_{t-1}^\top \overline{\mathbf{Q}}_{t-1} \boldsymbol{x}_0\right)}{\boldsymbol{x}_t^\top \overline{\mathbf{Q}}_t \boldsymbol{x}_0}. \end{aligned} \tag{4}$$

In the reverse denoising process, a straightforward approach is to use a bidirectional transformer encoder blocks to fit the distribution $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$. However, similar to the methods used in prior work (Sohl-Dickstein et al., 2015; Hoogeboom et al., 2021; Austin et al., 2021; Gu et al., 2022), we integrate an additional neural network $\tilde{p}_\theta(\tilde{\boldsymbol{x}}_0|\boldsymbol{x}_t)$. By summing over potential $\tilde{x}_0$ values, it can be transformed into a one-step reverse denoising process, yielding the following representation:

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \frac{\sum_{\tilde{x}_0} \tilde{p}_\theta(\tilde{\boldsymbol{x}}_0|\boldsymbol{x}_t) \cdot q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_0)}{\sum_{\boldsymbol{x}_{t-1}} \sum_{\tilde{x}_0} \tilde{p}_\theta(\tilde{\boldsymbol{x}}_0|\boldsymbol{x}_t) \cdot q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_0)} \propto \sum_{\tilde{x}_0} \tilde{p}_\theta(\tilde{\boldsymbol{x}}_0|\boldsymbol{x}_t) \cdot q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \tilde{\boldsymbol{x}}_0). \tag{5}$$

The loss function, which includes both the conventional variational lower bound $\mathcal{L}_{\text{vlb}}$ and an auxiliary denoising objective, is given by:

$$\mathcal{L}_\lambda = \mathcal{L}_{\text{vlb}} + \lambda \mathbb{E}_{\substack{\boldsymbol{x}_t \sim q(\boldsymbol{x}_t|\boldsymbol{x}_0) \\ \boldsymbol{x}_0 \sim q(\boldsymbol{x}_0)}} \left[-\log \tilde{p}_\theta(\boldsymbol{x}_0|\boldsymbol{x}_t)\right], \tag{6}$$

where λ is a balancing hyper-parameter. VQ-Diffusion introduces an enhancement to $\mathbf{Q}_t$ via the mask-and-replace strategy. The transition matrix $\mathbf{Q}_t$ is:

$$\mathbf{Q}_t = \begin{bmatrix} \alpha_t + \beta_t & \beta_t & \cdots & \beta_t & 0 \\ \beta_t & \alpha_t + \beta_t & \cdots & \beta_t & 0 \\ \vdots & \vdots & \ddots & \beta_t & 0 \\ \beta_t & \beta_t & \beta_t & \alpha_t + \beta_t & 0 \\ \gamma_t & \gamma_t & \gamma_t & \gamma_t & 1 \end{bmatrix}, \tag{7}$$

where α_t, β_t and γ_t are designed such that z_t converges to the [MASK] token as t grows. During testing, we start from $\boldsymbol{x}_t$ with [MASK] tokens and iteratively sample $\boldsymbol{x}_{t-1}$ from $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$.

3.2.2. Unconditional generation

In the task of unconditional CAD sequence generation, the initial step involves compressing loops and profiles within each sequence into codebook using VQ-VAE, as shown in the training stage 1 in Fig. 2. Similar to the training stage 2, a code tree decoder is trained. Each sequence is then represented by code tree, enabling a diffusion process that begins from the mask within the sequence. During the training of this diffusion model, no conditions are applied. In the inference phase, the diffusion model is first used to generate a code tree, which is then decoded by the code tree decoder to produce a complete command sequence.

3.2.3. Conditional generation

To facilitate conditional generation, we channel conditions to the model via the cross-attention of a transformer decoder. To streamline the training process, we limit our conditional generation experiments to sketches. This approach can be expanded to 3D models by training from photographs taken from different angles, akin to methodologies in CLIP-Sculptor and CLIP-Forge. As part of our approach, we utilize a pre-trained and frozen version of CLIP as our encoder.

More specifically, as shown in Fig. 2, during the training phase, our dataset comprises solely of command sequences. We first generate a code tree from the initial sequence, mirroring the approach for unconditional generation. Concurrently, we employ a parser to convert the initial sequence into an image. During the training, we leverage a frozen CLIP image encoder to encode the image and derive its latent vector. This latent vector serves as the condition, with the code tree as the target, to train VQ-Diffusion. Given the robust encoding capabilities of CLIP's image encoder, our inference phase can handle not only correctly parsed images but also hand-drawn sketches, allowing for zero-shot predictions. Moreover, thanks to CLIP's contrastive learning, it can encode semantically similar images and texts into corresponding latent vectors. Through experiments, we discover that to some extent, we can employ CLIP's text encoder for text-driven CAD command synthesis, as depicted in Fig. 3. Inspired by the prompt engineering (Radford et al., 2021), we further enhance the text input. Specifically, for each template in our template list, we fill in the desired object to be generated, creating a set of enriched text prompts. Each of these prompts is then fed into the text encoder separately, yielding a series of feature vectors. We compute the average of these feature vectors to obtain a consolidated feature representation for the desired object. By feeding this refined feature representation into the diffusion transformer block, we significantly boost the efficacy of text-guided CAD sequence generation process.

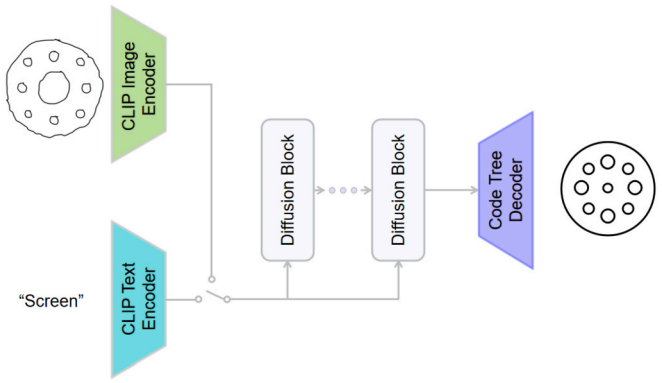


Fig. 3. Inference schematic. Utilizing CLIP's text or image encoder to encode conditional information for guiding the diffusion process.

4. Experimental results

4.1. Implementation details

4.1.1. Datasets

In our study, we utilized the expansive DeepCAD dataset (Wu et al., 2021) that is annotated with sketch-and-extrude models. This dataset encompasses 178,238 such models, divided into 90% for training, 5% for validation, and the remaining 5% for testing. Drawing from methodologies in prior approaches (Xu et al., 2023, 2022; Willis et al., 2021), we identified and omitted duplicated models in the training subset. After delineating the hierarchical attributes for loop, profile, and solid, we proceeded to eliminate repetitive properties at each tier. Each coordinate was converted into a 6-bit number via quantization. Furthermore, our training was limited to CAD models that conformed to specific criteria: a maximum of 5 solids, up to 20 loops for every profile, no more than 60 curves for every loop, and an upper limit of 200 commands within the sketch-and-extrude sequence. Post this stringent duplication and filtration process, our training dataset comprised 102,114 solids, 60,584 profiles, and 150,158 loops for the purpose of codebook learning. Additionally, it contained 124,451 sketch-and-extrude sequences tailored for CAD model creation. For conditional generation, we adopted the approach from SkexGen (Xu et al., 2022) and HNC-CAD (Xu et al., 2023), and extracted sketches directly from DeepCAD dataset. Post the elimination of duplicates, our training was based on a total of 99,650 sketches. To demonstrate the capability of our conditional generation model, we utilized the test set to extract sketches for testing, resulting in a total of 4,842 sketches.

4.1.2. Optimization and configuration

Models are optimized using an Nvidia A100 GPU. The Transformer backbone employs pre-layer normalization and consists of 4 layers, each with 8 attention heads. The input size for the embedding layer is set to 256, and the feed-forward network scales to a dimension of 512. During training, a dropout rate of 0.1 is applied to prevent the omission of features. We adopt a VQ-VAE codebook similar to the one utilized in HNC-CAD (Xu et al., 2023). For conditioned generation, we leverage the ViT-B/32 model from CLIP (Radford et al., 2021), based on the Vision Transformer architecture (Dosovitskiy et al., 2020), and freeze it to serve as our text and image encoder. The parameter λ in Equation (6) is set to 0.1 as Inoue et al. (2023), and the number of diffusion timesteps T is 100. For optimization, the AdamW optimizer (Loshchilov and Hutter, 2018) is used with a learning rate of 5.0×10^{-4}, $\beta_1 = 0.9$, and $\beta_2 = 0.98$.

4.1.3. Evaluations

To measure the capability of generating CAD models, we convert them to point clouds, and then use the metrics *Coverage (COV), Jensen-Shannon Divergence (JSD), and Maximum Mean Discrepancy (MMD)* for comparison (Achlioptas et al., 2018; Wu et al., 2021; Xu et al., 2021, 2023). COV is computed as the percentage of ground-truth models that have at least one closely matching generated model. Formally,

$$\mathrm{COV}(S,G) = \frac{\left|\left\{\arg\min_{Y\in S} d(X,Y) \mid X \in G\right\}\right|}{|S|}, \tag{8}$$

where $d(X,Y)$ denotes the chamfer distance between two point clouds of X and Y, S represents the ground-truth set, and G represents the generated model set. MMD calculates the average minimum distance between each point in the generated set and the points in the ground-truth set. Formally,

$$\mathrm{MMD} = \frac{1}{N}\sum_{i=1}^{N} \min_j d(M_i, G_j), \tag{9}$$

where M_i represents the i-th model in the ground-truth set, G_j represents the j-th generated model, d is a distance metric, and N is the total number of models in the generated set. JSD measures the similarity between two probability distributions. In the context of point clouds, it can be calculated using the following formula,

$$\mathrm{JSD}(P,Q) = \frac{1}{2} D_{KL}(P \,||\, M) + \frac{1}{2} D_{KL}(Q \,||\, M), \tag{10}$$

where P and Q are the probability distributions of the ground-truth and generated data, M is the average of P and Q, and D_{KL} is the Kullback-Leibler divergence.

To assess novelty, we further compute *Novel and Unique* values between the generated command sequences and the training set (Wu et al., 2021; Xu et al., 2021, 2023). Novel measures the novelty of the generated samples. Let G represent the set of generated samples and T represent the set of training samples, then Novel is calculated as:

$$\mathrm{Novel} = \frac{|G \setminus T|}{|G|} \times 100\%, \tag{11}$$

where $G \setminus T$ represents the set of unique samples in G that do not appear in T and a "unique sample" refers to a generated model that does not match any model in the training set. Unique metric measures the uniqueness of the generated samples. Let G represent the set of generated samples, and U represent the set of unique samples that appear only once in G. Formally, Unique is calculated as:

$$\mathrm{Unique} = \frac{|U|}{|G|} \times 100\%. \tag{12}$$

To evaluate the regularity of generated sketches, we further introduce a symmetry metric, which assesses symmetry through horizontal and vertical flipping based on the sketch's centroid coordinates. The symmetry metric is given by:

$$\mathrm{Symmetry}(I) = \frac{1}{N} \sum_{i=1}^{h} \sum_{j=1}^{w} \left| I(i,j) - I(i, 2O_y - j) \right| + \frac{1}{N} \sum_{i=1}^{h} \sum_{j=1}^{w} \left| I(i,j) - I(2O_x - i, j) \right|. \tag{13}$$

Here, w and h are the width and height of the image, respectively, N is the total number of pixels in the image, and O represents the centroid with coordinates (O_x, O_y).

4.2. Data augmentation and its effects on implicit structure

In the realm of CAD sequence generation, previous works such as SkexGen and HNC-CAD have employed data augmentation techniques to enhance model robustness and performance. Specifically, they introduced noise to the input of transformer decoders, whose objective remains the generation of correct sequences even in the presence of noise. However, upon examining the generated sequences of HNC-CAD, anomalies became evident, *i.e.*, certain generated sequences exhibited characteristics that defied CAD logic, such as non-vertical sides of rectangles or asymmetrical objects that should ideally be symmetrical, as demonstrated in the following experiments. A deeper dive into this phenomenon suggested that the employed data augmentation might inadvertently disturb the intrinsic symmetries in the data distribution. Many tasks, especially those involving sequence generation, involve data where elements have inherent relationships or dependencies. Data augmentation, like noise introduction, is commonly employed to improve model generalizability. Yet, this can sometimes have unintended ramifications on the data's implicit structure.

More specifically, consider a rudimentary dataset containing sequences $[0,0]$ and $[1,1]$. Let the first position in the sequence be X and the second be Y. Here, Y is inextricably linked to X. The conditional probabilities for this setup are:

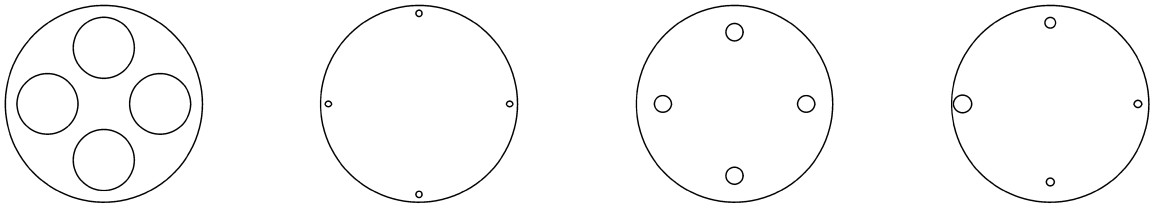
$$P(Y=0|X=0) = P(Y=1|X=1) = 1, \quad P(Y=1|X=0) = P(Y=0|X=1) = 0.$$

Assuming a data augmentation strategy introduces noise that alters the first position value with the probability equating to 0.5. This leads to potential sequence variations. Subsequent to this modification, the conditional probabilities evolve:

$$P(Y=0|X=0) = P(Y=1|X=0) = P(Y=1|X=1) = P(Y=0|X=1) = 0.5.$$

The advent of noise alters the model's optimization objectives. From comprehending deterministic relationships in the original data, the model gravitates towards learning a uniform conditional probability distribution. This metamorphosis underscores the nuances and pivotal considerations imperative when wielding data augmentations, as they can unintentionally modify a dataset's intrinsic fabric.

To validate the correctness of our analysis, we selected a sketch sharing the same code tree as a mini-dataset. This dataset comprises 27 images, all constructed from one large outer circle and four smaller inner circles. Theoretically, this scenario mirrors the previously discussed example. As shown in Fig. 4, models trained with data augmentation produced instances where the sizes of the four inner circles vary, disrupting the desired symmetry. In contrast, models trained without data augmentation better preserved the symmetry of the four inner circles, validating our observation.



(a) Ground Truth (b) Ground Truth (c) Without Augmentation (d) With Augmentation

Fig. 4. Sketch examples: (a)&(b) are random training set sketches. (c) A symmetry-preserving sketch generated without data augmentation. (d) A sketch from training with data augmentation, losing the inner circle symmetry.

Table 2
Quantitative evaluations on the CAD generation task. We adopt COV, MMD, JSD, and Percentage Scores for Unique and Novel for result assessment. **Bold** fonts indicate the best generator.

Method	COV % ↑	MMD ↓	JSD ↓	Novel % ↑	Unique % ↑
DeepCAD (Wu et al., 2021)	80.62	1.10	3.29	91.7	85.8
SkexGen (Xu et al., 2022)	84.74	1.02	0.90	**99.1**	99.8
HNC-CAD (Xu et al., 2023)	87.73	**0.96**	0.68	93.9	99.7
Ours	**88.11**	1.05	**0.64**	98.0	**99.9**

4.3. Unconditional generation

To demonstrate the capabilities of our diffusion module, we conducted experiments on the well-studied 3D generation task and compared it with state-of-the-art methods. Specifically, we still adopted the learning process depicted in the diagram, but did not use conditions for supervision. During generation, we performed random diffusion.

As illustrated in Table 2, our method surpasses the baselines in two evaluation metrics: COV and JSD, demonstrating significant improvements in quality and diversity. In terms of Unique and Novel scores, our performance either matches or closely approaches the state of the art, indicating that VQ-Diffusion has a better representation of the sample distribution and can produce more diverse results. Our generations exhibit enhanced diversity and novelty similar to SkexGen, yet with substantially improved COV and JSD scores. When pitted against HNC-CAD, not only do our COV and JSD scores fare better, but we also register significantly higher Novel score. Although our MMD lags slightly behind SkexGen, this is primarily due to the greater size diversity in our generated samples. This size variance leads to larger minimum distances between some generated samples and the test set, reflecting our model's expansive generative capacity. However, this increase in size diversity can impact the MMD metric's performance. Despite this, the ability to generate a wide range of sizes demonstrates our model's adaptability and highlights its potential to explore data's inherent distribution more broadly. The JSD metric underscores that the point clouds we generated align closer to the ground truth when viewed from the vantage point of probability distribution.

4.4. Conditional generation

4.4.1. Picture to CAD

To demonstrate the capability of our algorithm, we modified the HNC-CAD algorithm to use the same condition for training. In our evaluation focusing on symmetry, our algorithm showed a notable advantage. The ground truth (GT) scored 0.03907 in symmetry, whereas HNC-CAD scored slightly higher at 0.04017, reflecting less symmetry. Our method achieved a lower score of 0.03921, closer to the ground truth, indicating superior symmetry compared to HNC-CAD. This underscores our algorithm's effectiveness in mirroring essential symmetrical aspects of CAD design. As shown in Fig. 5, our algorithm can produce CAD sequences that adhere more closely to the implicit rules, which is more in line with the rules of CAD design.

4.4.2. Zero-shot CAD generation

In the domain of text-to-image generation, works such as VQ-Diffusion and Latent Diffusion have demonstrated that diffusion models possess certain advantages over transformer models. Given this, we further explored this phenomenon in the context of CAD sequence generation. In our model, we replaced the diffusion mechanism with a transformer module, maintaining all other components unchanged. Considering the limited size of the CAD sequence dataset, which cannot generate meaningful CAD sequences for all texts, we selected specific texts that can produce meaningful CAD sequences for comparative experiments. Building on this foundation and incorporating prompt engineering, as shown in Fig. 6, observations revealed that for fundamental concepts like circle and rectangle, transformers can synthesize more precise and succinct CAD command sequences. However, when dealing with complex concepts such as volcano and rocket, transformer fails to deliver the expected output.

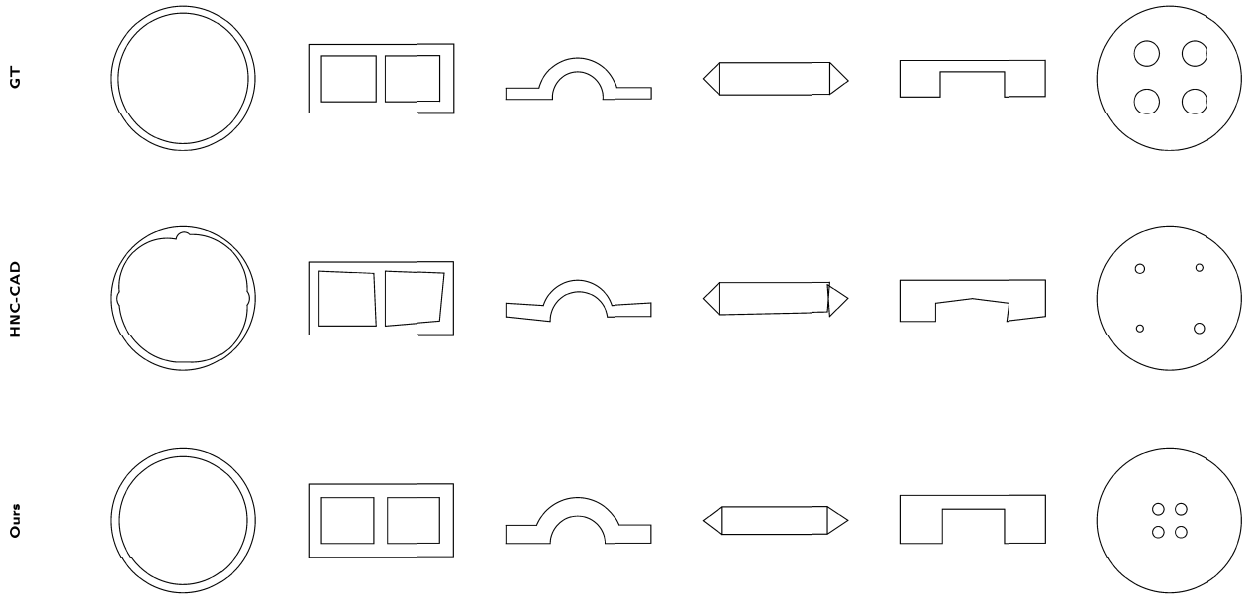


Fig. 5. Picture-to-Command result comparison: The first row displays input images. The second and third rows show the reconstruction results of HNC-CAD (Xu et al., 2023) and ours, respectively.

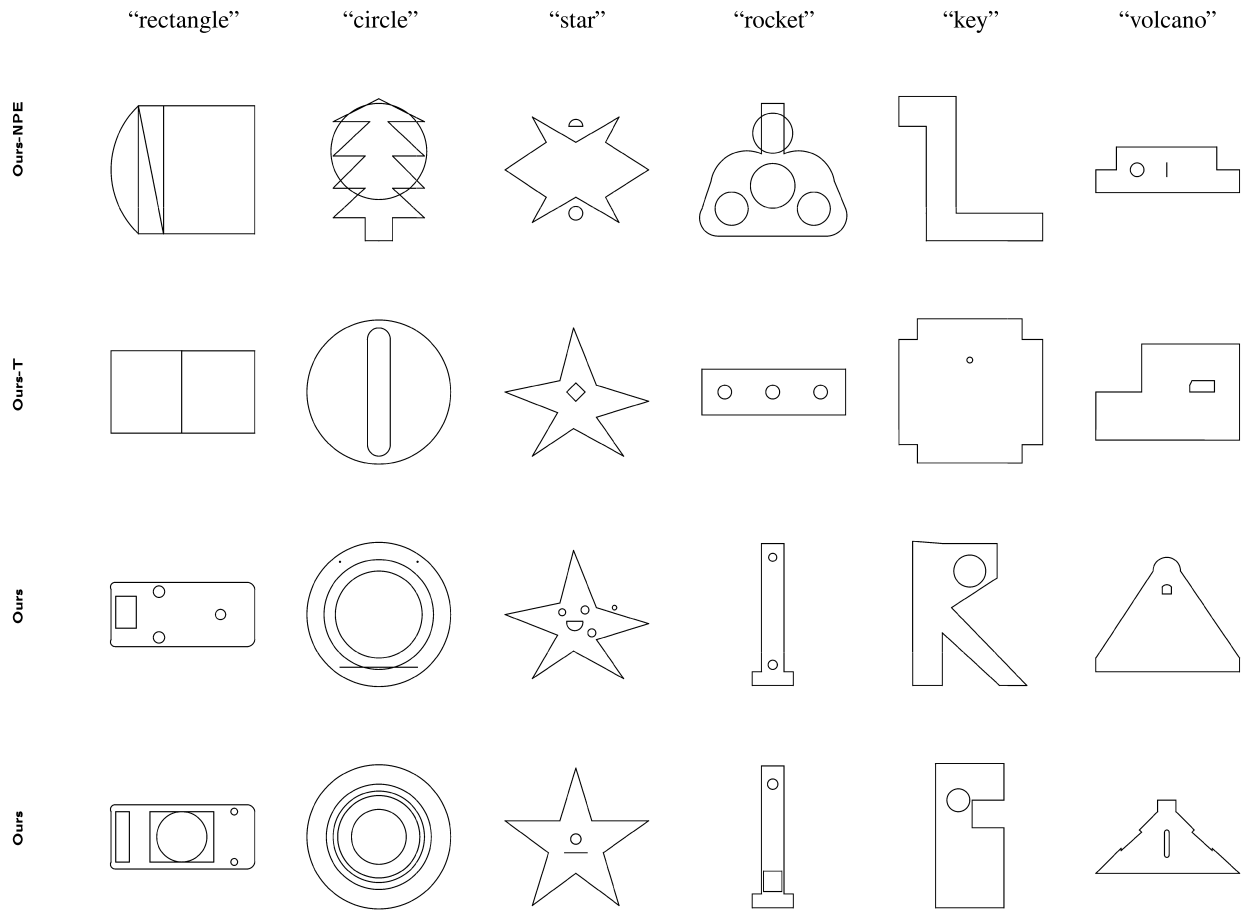


Fig. 6. Text-to-Command generation results of different methods: Given the top text as a condition, each row of images below represents the output of various models based on the provided condition. In this comparison, "Ours-NPE" denotes "Ours with No Prompt Engineering," and "Ours-T" refers to "Ours with Transformer."

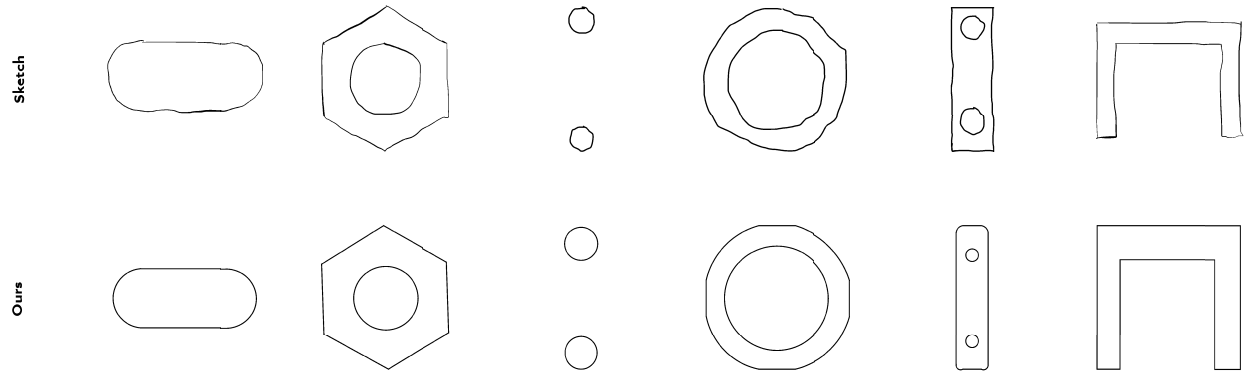


Fig. 7. Hand-drawn Sketch-to-Command generation results: The first row displays hand-drawn sketches, the second row shows our generation results.

Table 3
Ablation studies.

Method	COV % ↑	MMD ↓	JSD ↓	Novel % ↑	Unique % ↑
Ours w/o diffusion	**88.20**	**1.02**	**0.64**	78.1	98.8
Ours w/o VQ-VAE	84.74	1.20	3.17	96.8	99.8
Ours w/ data augmentation	87.73	1.08	0.66	**99.1**	**99.9**
Ours	88.11	1.05	**0.64**	98.0	**99.9**

We also endeavored to generate CAD sequences from sketches using models trained on regular images. As depicted in Fig. 7, even with sketches that have substantial random noise, the model is still capable of generating reasonably coherent CAD sequences.

4.5. Ablation study

Table 3 shows the ablation study conducted on the key components of VQ-CAD. Four different configurations are delineated: (1) without diffusion, replacing it with a transformer; (2) without using VQ-VAE, directly diffusing on the command sequence; (3) employing data augmentation; (4) our final proposed VQ-CAD. Table 3 reports the numerical results across various metrics including COV, MMD, JSD, Novel, and Unique.

In the setting of "Ours w/o diffusion", the quality indicator is high, while the diversity is limited. This may be attributed to the insufficiency of the dataset size, making the Transformer prone to overfitting on such a dataset, thereby compromising the model's generalization capability.

For "Ours w/o VQ-VAE", the quality of the generated results is inferior, especially for the JSD metric, which is fourfold that of our baseline method. This implies that diffusing on the code tree representation in VQ-VAE significantly enhances the quality of generation. In the "Ours with data augmentation" configuration, the quality of generation is comparable to our baseline, with better divergence performance. However, this configuration tends to overlook some implicit rules, as observed in the random generation results presented in Fig. 9 and Fig. 8. Specifically, Fig. 8 demonstrates that our method consistently produces symmetrical holes and corners across all generated models. In contrast, Fig. 9, which involves data augmentation, frequently exhibits non-vertical corners and asymmetrical holes. Our approach, therefore, yields more uniform and regular outcomes.

In summary, our model manages to attain a good quality of generation while ensuring diversity. A commendable trade-off between quality and diversity is achieved, retaining the implicit rules. This balanced achievement demonstrates the efficacy of our proposed VQ-CAD model, making a persuasive case for its application in conditional generation tasks.

4.6. Limitations and future work

We recognize that while our work is novel and effective, it bears certain limitations. The constrained size and lack of diversity in the dataset impede our capabilities in zero-shot generation. To mitigate this, future research could concentrate on amassing a more extensive set of CAD command data. Moreover, we can enhance the geometric accuracy of our model by incorporating relevant loss functions, which presents further opportunities for refinement in subsequent studies. Our approach can be expanded to 3D models by training from photographs taken from different angles, enabling the generation of CAD sequences from text and hand-drawn sketches seamlessly.

5. Conclusions

In this work, we introduce an approach to CAD model generation by leveraging the capabilities of VQ-Diffusion. Our methodology distinguishes itself through its inherent ability to discern implicit design constraints, suggesting potential automated mechanisms for

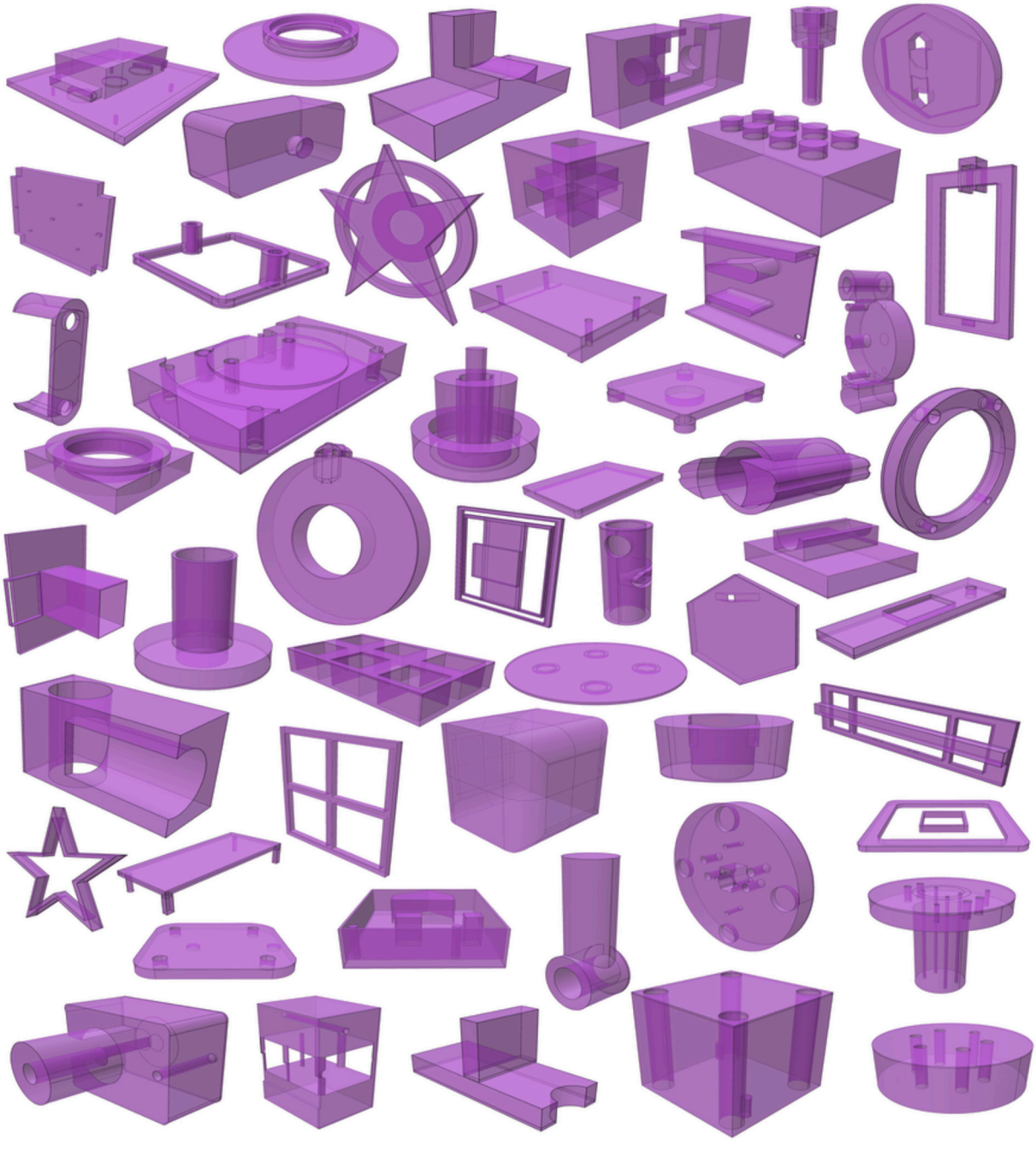


Fig. 8. Randomly generated 3D CAD models without data augmentation.

error correction in CAD designs. By integrating the robust features of CLIP, we unveil an enhanced system for generating CAD models. This synergistic approach facilitates a seamless transition from image sketches and textual annotations directly to CAD sequences, achieving a level of design regularity. This methodology signifies not only an advancement for practitioners in design but also inaugurates new avenues in CAD research, transitioning from a command-centric to a more agile, learning-focused design paradigm.

CRediT authorship contribution statement

Hanxiao Wang: Writing – original draft, Methodology. **Mingyang Zhao:** Writing – review & editing. **Yiqun Wang:** Software. **Weize Quan:** Validation. **Dong-Ming Yan:** Writing – review & editing, Supervision, Project administration.

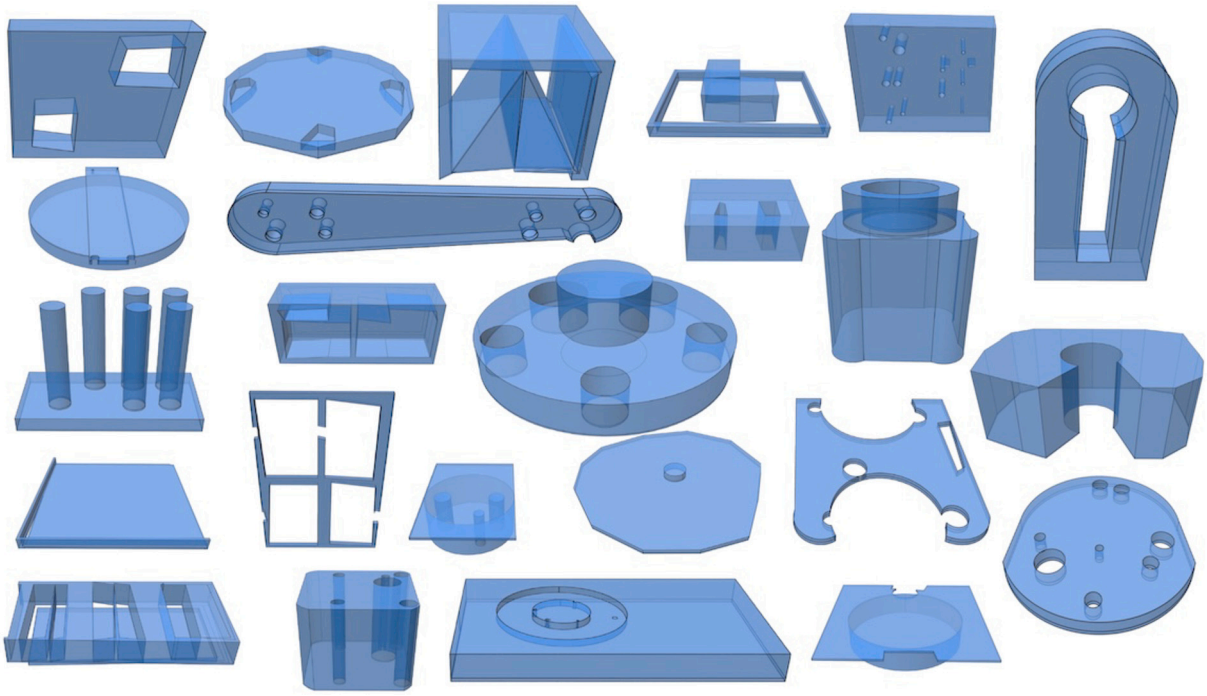


Fig. 9. Randomly generated 3D CAD models with data augmentation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (62172415, 62102418, 62202076), the Strategic Priority Research Program of the Chinese Academy of Sciences (XDB0640000), the Beijing Science and Technology Plan Project (Z231100005923033), and the Open Projects Program of State Key Laboratory of Multimodal Artificial Intelligence Systems.

References

Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L., 2018. Learning representations and generative models for 3D point clouds. In: Proc. Int. Conf. Mach. Learning, pp. 40–49.

Austin, J., Johnson, D.D., Ho, J., Tarlow, D., Van Den Berg, R., 2021. Structured denoising diffusion models in discrete state-spaces. Adv. Neural Inf. Process. Syst. 34, 17981–17993.

Camba, J.D., Contero, M., Company, P., 2016. Parametric CAD modeling: an analysis of strategies for design reusability. Comput. Aided Des. 74, 18–31.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2020. An image is worth 16x16 words: transformers for image recognition at scale. In: Proc. Int. Conf. Learning Representations.

Frans, K., Soros, L., Witkowski, O., 2022. CLIPDraw: exploring text-to-drawing synthesis through language-image encoders. Adv. Neural Inf. Process. Syst. 35, 5207–5218.

Ganin, Y., Bartunov, S., Li, Y., Keller, E., Saliceti, S., 2021. Computer-aided design as language. Adv. Neural Inf. Process. Syst. 34, 5885–5897.

Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B., 2022. Vector quantized diffusion model for text-to-image synthesis. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 10696–10706.

Hähnlein, F., Li, C., Mitra, N.J., Bousseau, A., 2022. CAD2Sketch: generating concept sketches from CAD sequences. ACM Trans. Graph. 41, 1–18.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R., 2022. Masked autoencoders are scalable vision learners. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 16000–16009.

Ho, J., Jain, A., Abbeel, P., 2020. Denoising diffusion probabilistic models. Adv. Neural Inf. Process. Syst. 33, 6840–6851.

Hoogeboom, E., Nielsen, D., Jaini, P., Forré, P., Welling, M., 2021. Argmax flows and multinomial diffusion: learning categorical distributions. Adv. Neural Inf. Process. Syst. 34, 12454–12465.

Ikubanni, P.P., Adeleke, A.A., Agboola, O.O., Christopher, C.T., Ademola, B.S., Okonkwo, J., Adesina, O.S., Omoniyi, P.O., Akinlabi, E.T., 2022. Present and future impacts of Computer-Aided Design/Computer-Aided Manufacturing (CAD/CAM). J. Eur. Syst. Autom. 55.

Inoue, N., Kikuchi, K., Simo-Serra, E., Otani, M., Yamaguchi, K., 2023. LayoutDM: discrete diffusion model for controllable layout generation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 10167–10176.
Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B., 2022. Zero-shot text-guided object generation with dream fields. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 867–876.
Kania, K., Zieba, M., Kajdanowicz, T., 2020. UCSG-NET-unsupervised discovering of constructive solid geometry tree. Adv. Neural Inf. Process. Syst. 33, 8776–8786.
Lambourne, J.G., Willis, K., Jayaraman, P.K., Zhang, L., Sanghi, A., Malekshan, K.R., 2022. Reconstructing editable prismatic CAD from rounded voxel models. In: SIGGRAPH Asia 2022 Conference Papers, pp. 1–9.
Li, C., Pan, H., Bousseau, A., Mitra, N.J., 2020. Sketch2CAD: sequential CAD modeling by sketching in context. ACM Trans. Graph. 39, 1–14.
Li, C., Pan, H., Bousseau, A., Mitra, N.J., 2022. Free2CAD: parsing freehand drawings into CAD commands. ACM Trans. Graph. 41, 1–16.
Li, P., Guo, J., Zhang, X., Yan, D.M., 2023. SECAD-Net: self-supervised CAD reconstruction by learning sketch-extrude operations. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 16816–16826.
Loshchilov, I., Hutter, F., 2018. Decoupled weight decay regularization. In: Proc. Int. Conf. Learning Representations.
Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L., 2022. Repaint: inpainting using denoising diffusion probabilistic models. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 11461–11471.
Mohammad Khalid, N., Xie, T., Belilovsky, E., Popa, T., 2022. CLIP-Mesh: generating textured meshes from text using pretrained image-text models. In: SIGGRAPH Asia 2022 Conference Papers, pp. 1–8.
Oh, J.Y., Stuerzlinger, W., Danahy, J., 2006. SESAME: towards better 3D conceptual design systems. In: Proc. Conf. Des. Interact. Syst., pp. 80–89.
Para, W., Bhat, S., Guerrero, P., Kelly, T., Mitra, N., Guibas, L.J., Wonka, P., 2021. Sketchgen: generating constrained CAD sketches. Adv. Neural Inf. Process. Syst. 34, 5077–5088.
Poole, B., Jain, A., Barron, J.T., Mildenhall, B., 2022. Dreamfusion: text-to-3D using 2D diffusion. In: Proc. Int. Conf. Learning Representations.
Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 2021. Learning transferable visual models from natural language supervision. In: Proc. Int. Conf. Mach. Learning, pp. 8748–8763.
Ren, D., Zheng, J., Cai, J., Li, J., Jiang, H., Cai, Z., Zhang, J., Pan, L., Zhang, M., Zhao, H., et al., 2021. CSG-Stump: a learning friendly CSG-like representation for interpretable shape parsing. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 12478–12487.
Ren, D., Zheng, J., Cai, J., Li, J., Zhang, J., 2022. ExtrudeNet: unsupervised inverse sketch-and-extrude for shape parsing. In: Proc. Eur. Conf. Comput. Vis. Springer, pp. 482–498.
Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B., 2022. High-resolution image synthesis with latent diffusion models. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 10684–10695.
Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M., 2022. Image super-resolution via iterative refinement. IEEE Trans. Pattern Anal. Mach. Intell. 45, 4713–4726.
Sanghi, A., Chu, H., Lambourne, J.G., Wang, Y., Cheng, C.Y., Fumero, M., Malekshan, K.R., 2022. CLIP-Forge: towards zero-shot text-to-shape generation. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 18603–18613.
Sanghi, A., Fu, R., Liu, V., Willis, K.D., Shayani, H., Khasahmadi, A.H., Sridhar, S., Ritchie, D., 2023a. CLIP-Sculptor: zero-shot generation of high-fidelity and diverse shapes from natural language. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 18339–18348.
Sanghi, A., Jayaraman, P.K., Rampini, A., Lambourne, J., Shayani, H., Atherton, E., Taghanaki, S.A., 2023b. Sketch-a-Shape: zero-shot sketch-to-3D shape generation. arXiv preprint. arXiv:2307.03869.
Seff, A., Zhou, W., Richardson, N., Adams, R.P., 2021. Vitruvion: a generative model of parametric CAD sketches. In: Proc. Int. Conf. Learning Representations.
Shahin, T.M., 2008. Feature-based design–an overview. Comput-Aided Des. Appl. 5, 639–653.
Sharma, G., Goyal, R., Liu, D., Kalogerakis, E., Maji, S., 2018. CSGNET: neural shape parser for constructive solid geometry. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 5515–5523.
Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S., 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In: Proc. Int. Conf. Mach. Learning, pp. 2256–2265.
Uy, M.A., Chang, Y.Y., Sung, M., Goel, P., Lambourne, J.G., Birdal, T., Guibas, L.J., 2022. Point2cyl: reverse engineering 3D objects from point clouds to extrusion cylinders. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 11850–11860.
Vahdat, A., Kreis, K., Kautz, J., 2021. Score-based generative modeling in latent space. Adv. Neural Inf. Process. Syst. 34, 11287–11302.
Vahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K., et al., 2022. Lion: latent point diffusion models for 3D shape generation. Adv. Neural Inf. Process. Syst. 35, 10021–10039.
Van Den Oord, A., Vinyals, O., Kavukcuoglu, K., 2017. Neural discrete representation learning. Adv. Neural Inf. Process. Syst. 30.
Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J., 2023. Prolificdreamer: high-fidelity and diverse text-to-3D generation with variational score distillation. arXiv preprint. arXiv:2305.16213.
Willis, K.D., Jayaraman, P.K., Lambourne, J.G., Chu, H., Pu, Y., 2021. Engineering sketch generation for Computer-Aided Design. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 2105–2114.
Wu, R., Xiao, C., Zheng, C., 2021. DeepCAD: a deep generative network for Computer-Aided Design models. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 6772–6782.
Xu, X., Peng, W., Cheng, C.Y., Willis, K.D., Ritchie, D., 2021. Inferring CAD modeling sequences using zone graphs. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 6062–6070.
Xu, X., Willis, K.D., Lambourne, J.G., Cheng, C.Y., Jayaraman, P.K., Furukawa, Y., 2022. SkexGen: autoregressive generation of CAD construction sequences with disentangled codebooks. In: Proc. Int. Conf. Mach. Learning, pp. 24698–24724.
Xu, X., Jayaraman, P.K., Lambourne, J.G., Willis, K.D., Furukawa, Y., 2023. Hierarchical neural coding for controllable CAD model generation. arXiv preprint. arXiv:2307.00149.
Yang, D., Yu, J., Wang, H., Wang, W., Weng, C., Zou, Y., Yu, D., 2023. DiffSound: discrete diffusion model for text-to-sound generation. IEEE/ACM Trans. Audio Speech Lang. Process.
Yu, F., Chen, Z., Li, M., Sanghi, A., Shayani, H., Mahdavi-Amiri, A., Zhang, H., 2022. CAPRI-Net: learning compact CAD shapes with adaptive primitive assembly. In: Proc. IEEE Conf. Comput. Vis. Pattern Recog., pp. 11768–11778.
Yu, F., Chen, Q., Tanveer, M., Amiri, A.M., Zhang, H., 2023. DualCSG: learning dual CSG trees for general and compact CAD modeling. arXiv preprint. arXiv: 2301.11497.