

AIMold: An Autonomous AI-based Pipeline for Complex Mold Design

Pengyun Qiu^{1*}  Shuo Wang^{1*}  Zeyuan Chen¹ 
Yihao Zhi¹  Chongjie Ye^{3,1}  Xiaoguang Han^{1,2,3} 

¹ School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen

² Guangdong Provincial Key Laboratory of Future Networks of Intelligence

³ FNii-Shenzhen

{pengyunqiu, wangshuo, hanxiaoguang}@cuhk.edu.cn

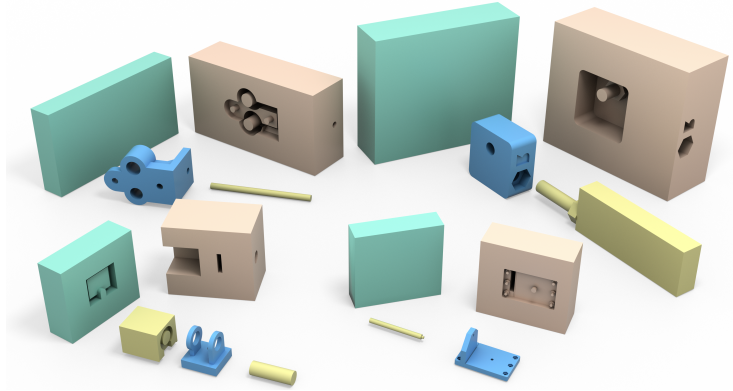


Fig. 1: Given a single-body CAD model (blue), AIMold generate high quality assembly mold parts, including auxiliary components (yellow), upper and lower molds (cyan/brown).

Abstract. Injection molding is the cornerstone of mass-producing plastic components. While current algorithms can automate mold design for basic geometries using standard two-piece molds, complex parts featuring undercuts, side holes, or re-entrant features present a significant challenge. These geometries often necessitate auxiliary components beyond the primary upper and lower molds. In practice, designing these intricate assemblies is a laborious process that relies heavily on expert knowledge. Furthermore, the scarcity of public datasets has hindered the development of effective learning-based solutions. To bridge these gaps, we introduce **MoldCAD**, a curated dataset that pairs complex single-body CAD parts with industry-standard mold assemblies.

* Equal contribution.  Corresponding author.

Each entry includes the upper and lower molds, parting surfaces, demolding orientations, and necessary auxiliary components. The dataset comprises **4,934** CAD models and over **3,850** mold assemblies, totaling more than **23k** individual models. Building upon this dataset, we propose a comprehensive pipeline that predicts demolding orientations, identifies auxiliary components, and constructs parting surfaces to derive a complete, manufacturing-ready mold assembly for downstream CAD/CAM workflows. Our results demonstrate a promising path toward fully automated industrial mold design and contribute to the broader advancement of manufacturing-aware CAD generation. Code is available at <https://github.com/tb2-sy/AIMold>.

1 INTRODUCTION

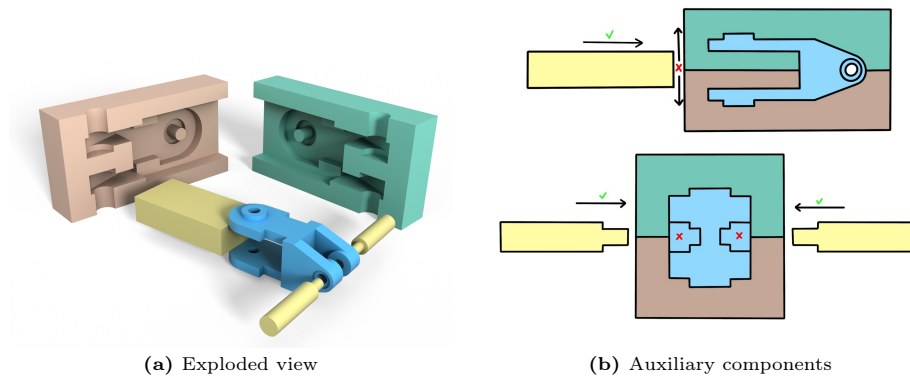


Fig. 2: Mold assembly details. (a) Exploded view showing the input part (blue), molds (cyan/brown), and auxiliary components (yellow). (b) Geometric necessity: Auxiliary components provide lateral freedom for collision-free demolding.

Injection molding is the foundation of global industrial manufacturing. It facilitates the production of a vast array of products, from small components like keyboard keycaps and bottle caps to large-scale items such as integrated chairs and fan housings. In this process, heated materials, like resins or thermoplastics, are injected into a precision mold, solidifying into a specific shape. A well-made mold can typically endure millions of production cycles. However, while the physical fabrication stage is largely automated, mold design persists as a significant bottleneck in many computer-aided manufacturing (CAM) workflows.

In traditional industrial practice, mold design is a staged workflow executed by experienced engineers. The process begins with selecting an optimal demolding orientation, followed by the layout of a parting surface to bifurcate the upper

and lower molds. Finally, engineers must integrate auxiliary components whenever undercuts or side holes obstruct the demolding path of a standard two-piece mold (Figure 2b). The reliance on domain expertise creates a significant bottleneck, limiting the scalability and efficiency of the broader manufacturing pipeline.

Automating this assembly generation remains challenging. We focus on a prevalent setting as shown in Figure 2a. Given a single-body part as input, our objective is to generate a comprehensive assembly comprising the upper and lower molds, the parting surface, and a variable number of auxiliary components. This scope specifically addresses the scenarios that dominate manual design: parts with undercuts and side holes. Such features are usually incompatible with two-piece molds and require the incorporation of auxiliary components.

Data-driven approaches, particularly deep learning, hold significant promise for capturing complex design regularities. However, prior research has typically targeted isolated subproblems—such as determining demolding directions [9, 28] or detecting undercuts [6, 58], rather than addressing the holistic construction of the complete mold assembly. Consequently, these methods often rely on geometric heuristics that are fragile and limited to simple, idealized shapes. As geometric complexity increases, the space of feasible mechanisms expands exponentially, rendering traditional design logic difficult to formalize. In contrast, recent generative models offer a more compelling path forward, as they can learn implicit design patterns that defy formalization and integrate these decisions into a unified generation process. Nevertheless, progress in this direction is fundamentally obstructed by a critical data gap: there are virtually no public datasets that pair single-body parts with professionally designed mold assemblies. To the best of our knowledge, the scarcity of manufacturing-ready CAD datasets containing such paired assemblies makes it nearly impossible to train, evaluate, or even frame mold assembly generation as a viable learning problem. Bridging this gap is essential to achieving a transformative leap in automated mold design [2].

To address the scarcity of high-quality annotated CAD data, we present **MoldCAD**, a large-scale paired dataset comprising approximately 5,000 part–mold assembly examples. Constructing this benchmark presents unique challenges beyond simple data aggregation. Specifically, we manually selected candidate parts that satisfy strict manufacturing prerequisites—ensuring they are manifold, watertight, and solid—while specifically retaining complex features such as undercuts and side holes. To ensure diversity, we curated the collection to eliminate near-duplicate geometries. For every part, we provide a complete mold assembly authored by professional engineers, accompanied by consistent semantic annotations for every component within the assembly. This dataset enables researchers to learn and benchmark the generation of both the final CAD shapes and the underlying mechanical structures required for successful demolding.

Building upon MoldCAD, we formulate mold assembly generation as a conditional generation problem, which includes auxiliary components, upper and lower mold generation. Given an input CAD model, we need to estimate its op-

timal orientation for subsequent manufacturing processes. We employ a plane-detection-based method to generate a candidate set of orientations and train a binary classifier using manually curated positive and negative samples to select the most suitable orientation. For assembly generation, we devise a two-stage framework. In the first stage, a coarse structural representation is generated in the voxel domain to establish the global layout and component relationships. In the second stage, we refine this coarse draft into detailed assembly meshes, recovering the fine geometric detail that the low-resolution first stage cannot capture. These meshes are then regularized based on the input model through geometry processing routines and exported as CAD-friendly representations for downstream CAD/CAM verification, without introducing additional learned components.

In summary, we make the following contributions:

- We raise the challenge of mold manufacturing design and, for the first time, employ deep learning to develop a comprehensive, step-by-step approach. This approach covers demolding orientation estimation, auxiliary component generation, upper and lower mold generation, and the conversion of generated meshes into CAD models.
- To support this approach, we introduce **MoldCAD**, a carefully curated dataset designed specifically for the mold design task, which includes rich annotations such as auxiliary components, parting surfaces, demolding orientations, and paired upper and lower molds.
- Extensive experiments demonstrate the promising results of our data-driven approach to transform complex mold design into an automated process, substantially reducing costs associated with traditional industrial design workflows.

2 RELATED WORK

2.1 Mold design

Recent advancements in digital fabrication technology have driven research on mold design in the fields of computer-aided design, computer graphics, and mechanical engineering. Early methods addressed moldability by optimizing accessibility metrics, including undercut area, parting-surface flatness, and parting depth [9, 28]. Later studies formalized these concepts through visibility analysis, leveraging volumetric or mesh-based discretizations to produce dense demoldability maps and valid manufacturing annotations [13, 15, 51].

Rigid molds require strict geometric compliance, motivating extensive work on automating the search for feasible parting directions and lines/surfaces. [18] approximate freeform shapes to minimize the number of mold pieces, and [65] deform inputs into shapes castable with two-piece rigid molds, while [3] segment complex objects into height-field parts via volumetric decomposition without altering the geometry.

For models that violate two-piece constraints, multi-piece molds and side-actions map undercut features to candidate retraction spaces, often via graph-based search or cost optimization [6, 15, 58]. More complex parts are partitioned by reasoning about core regions and generating parting curves through geometry- and graph-based analyses [21, 40].

Parallel efforts use flexible materials and composite molds: [50] generate peelable flexible shells via dynamic simulation, and [5] compute optimal cut layouts for silicone molds. As surface-based analysis often fails on intricate geometries, [4] combine a rigid outer shell with a soft inner part defined by volumetric escape-path analysis. Beyond casting, [86] optimize flat-panel shapes and supports to cast plaster sculptures via fluid-pressure deformation.

2.2 Learning-based CAD modeling

Our work also builds on learning-based CAD generation, which has evolved from recovering modeling sequences to directly generating boundary representations (B-Reps). Early methods treated CAD modeling as sequence prediction, recovering the construction history (e.g., sketch-and-extrude operations): Transformer-based models generate command sequences [71], later refined with decoupled or hierarchical codebooks for finer geometric and topological control [76, 78].

In particular, [80] parse discrete CAD commands through Visual Language Models (VLMs). Similarly, sketch-based approaches either parse hand-drawn strokes into parametric CAD command sequences [33, 34, 57, 74] or directly generate constraint-aware sketches [55, 61]. Although sequence-based approaches ensure validity, they face a fundamental one-to-many mapping challenge: the same final geometry may admit many different operation sequences, which complicates learning and evaluation. A parallel stream of research investigates the direct generation of B-Reps. [75] and [23] utilize autoregressive Transformers and a unified tokenization scheme to achieve end-to-end generation of vertices, edges, and faces. To treat topology and geometry independently, diffusion-based and latent variable models [17, 32, 38, 46, 77] have been widely used.

Other methods reconstruct editable CAD directly from raw geometry. Self-supervised approaches recover shapes by assembling primitives [81] or by learning sketch, feature, and machining operations [36, 37, 79], whereas reverse engineering pipelines fit parametric models to point clouds [14, 27, 47].

Multimodal conditioning further broadens usability: language and image conditioned generation builds structured representations through LLMs and multimodal training [1, 26, 74, 87], and text-guided B-Rep editing operates directly in a learned latent space without the construction history [45]. For representation learning, [24] proposes a grid-graph architecture for B-Rep classification and segmentation, while [49] aligns sequence and geometry representations via contrastive learning. Beyond single parts, [69] mates parts into constrained assemblies.

2.3 3D generative models

Early 3D generation methods primarily relied on Generative Adversarial Networks (GANs) [10, 16, 63, 70] for 3D modeling. Although these approaches demonstrated the feasibility of learning 3D representations [12, 90], they often suffered from limited generation quality and training instability. The emergence of diffusion models [19, 52, 53, 64] marked a conceptual shift in 3D generation, leading to a substantial improvement in generation quality. In the field of image generation [41, 59, 85], diffusion models have achieved remarkable success. This progress has sparked strong interest in lifting 2D generative models to 3D, leading to the emergence of Score Distillation Sampling (SDS) [39, 56, 66, 67]. However, SDS suffers from substantial additional optimization time and limited multi-view consistency. Large Reconstruction Models [20, 35, 43, 44] address the per-scene optimization issue by enabling generation through a single forward pass. With the continued expansion of large-scale 3D datasets [11, 82], improving the scalability of 3D latent diffusion models has become a central research focus, and many representative works [30, 68, 72, 73, 83, 84, 88, 89] have achieved significant progress in efficiently generating high-quality 3D shapes and structures. Different from these large-scale 3D generation methods, such performance is currently difficult to achieve in the mold design domain due to the limited size of available datasets. Therefore, our proposed method, AIMold, explores 3D generation under dataset constraints and achieves promising results.

3 METHOD

3.1 Problem definition

We formulate injection mold design as a conditional generation task that maps a single-body part to an assembly that can reproduce the part geometry and support demolding. We focus on complex parts with undercuts and side holes, where a two-piece mold is often insufficient and auxiliary components are required in addition to the upper and lower molds.

Input Let P denote a single-body part represented as a CAD solid or a watertight mesh. Our pipeline operates on a voxelized representation $V(P) \in \{0, 1\}^{N \times N \times N}$, obtained by rasterizing P in a canonical coordinate frame.

Output The target output is an assembly

$$\mathcal{A} = \{M_u, M_l, S_{part}, \mathcal{C}\}$$

where M_u and M_l are the upper and lower molds, S_{part} is the parting surface, and $\mathcal{C} = \{C_1, \dots, C_K\}$ are the auxiliary components. Note that K is not fixed but implicitly determined by the geometric features of P . Each element in $\mathcal{A}$ is represented as a high quality mesh to support downstream operations.

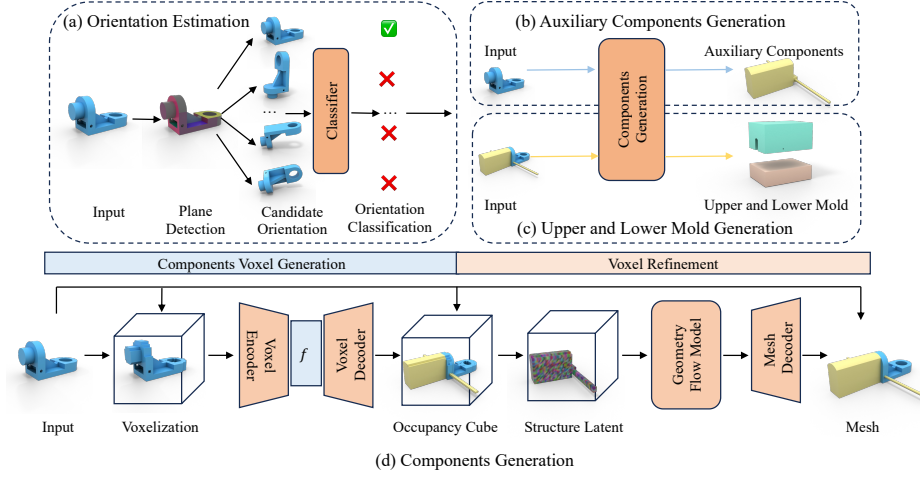


Fig. 3: Pipeline overview of our method. The framework first estimates the optimal model orientation via plane detection and classification (a), followed by the generation of auxiliary components (b), and upper and lower molds (c). The detailed generation process (d) operates in two sequential phases: Components Voxel Generation, which predicts a coarse occupancy cube from the voxelized input, and Voxel Refinement, which leverages a geometry flow model on structure latents to synthesize the faithful mesh.

Challenges This task is challenging for three reasons. First, the output is structured: the assembly consists of multiple interacting elements rather than an isolated entity. Second, the mapping from P to $\mathcal{A}$ is inherently one-to-many. Different demolding directions, parting surfaces, and auxiliary-component configurations can all be acceptable for the same part, which makes direct regression ill-posed. Third, paired data is scarce and highly niche: industrial mold assemblies follow strong design regularities that are difficult to design manually without expert knowledge.

Accordingly, we learn a conditional generative model $p(\mathcal{A} | V(P))$ from engineering designs. Our goal is to generate geometrically plausible assemblies that match common industrial patterns and can serve as structured drafts for downstream CAD/CAM verification and refinement. We do not explicitly enforce demolding constraints during generation; instead, we rely on the implicit manufacturability priors present in the training data.

3.2 Components Generation

We propose a two-stage pipeline for generating auxiliary components, upper and lower molds. Figure 3 shows an overview, with details described below. We first use a voxel encoder $\mathcal{E}$ to encode voxelized 3D model $\mathcal{O}$ to features $\mathbf{f}$ and decode features using a voxel decoder $\mathcal{D}$ as shown in Figure 3 (d). The voxel decoder’s outputs will be refined through our voxel refinement stage. This stage

predicts the 3D shape in the occupied regions based on the outputs from the first stage. In the second stage, we generate latents $\{\mathbf{z}_i\}_{i=1}^L$ given the structure $\{\mathbf{p}_i\}_{i=1}^L$ using a transformer $\mathcal{G}_L$ designed for sparse structures, where $\mathbf{p}_i \in \mathbb{R}^3$ denotes the spatial coordinates of the i -th active voxel within a voxel grid, and $\mathbf{z}_i$ encodes the associated voxel features. We use Rectified Flow [42] for the latent $\mathbf{z}$ generation.

$$\mathcal{L}_{FM}(\theta) = \mathbb{E}_{t, \mathbf{z}_0, \epsilon} \|\mathbf{v}_\theta(\mathbf{z}_t, t) - (\epsilon - \mathbf{z}_0)\|_2^2. \quad (1)$$

The model interpolates between data $\mathbf{z}_0$ and noise ϵ over time t via $\mathbf{z}_t = (1 - t)\mathbf{z}_0 + t\epsilon$. We parameterize $\mathbf{v}$ with a neural network $\mathbf{v}_\theta$ and train it using the Flow Matching (FM) objective as Eq. (1).

Representation of 3D Object To facilitate network training, we represent each 3D object using a voxel-based formulation with two attributes: $\mathcal{O} \triangleq \{\mathbf{z}_i, \mathbf{p}_i\}_{i=1}^L$. These features $\{\mathbf{z}_i\}_{i=1}^L$ are obtained by aggregating multi-view DINOv2 features [54] and subsequently compressed using a 3D variational autoencoder (VAE) encoder. Here, L denotes the total number of active voxels. In the first stage, we rely solely on the voxel coordinates $\mathbf{p}_i$ to voxelize the 3D objects. In the second stage, both voxel coordinates $\mathbf{p}_i$ and their associated features $\mathbf{z}_i$ are used to construct structural latent representations. The structural latents can be decoded into mesh representations [73].

Auxiliary Components Generation In the components generation stage, a 3D model without auxiliary components is translated into a voxel representation with auxiliary components with our components voxel generation model. Based on a voxelized representation $V(P) \in \{0, 1\}^{N \times N \times N}$, coarse geometry with union auxiliary components $\mathcal{C} = \{C_1, \dots, C_K\}$ is produced using a voxel-to-voxel paradigm. Although a CAD model may contain multiple auxiliary components, these components rarely intersect with each other. Therefore, naively predicting multiple auxiliary components within a shared coordinate system is feasible, and they can be separated in a straightforward manner. The voxel encoder and voxel decoder share the same transformer structure like a 3DVAE [73]. In our experiments, we observe that, at the current scale of the Mold dataset, this paradigm aligns more closely with the input 3D models than implicit representation methods [68, 88], as shown in Table 1.

Upper and Lower Mold Generation For CAD models equipped with auxiliary components, we subsequently generate the corresponding upper and lower molds. The upper and lower molds generation procedure follows a pipeline similar to that used for auxiliary component generation, differing only in the input and the generation targets. For auxiliary component generation, we take only $V(P)$ as input and predict the union voxel representation of $\mathcal{C}$. In contrast, for upper and lower mold generation, the input consists of the joint voxelized representation of $V(P)$ and $\mathcal{C}$ in a shared voxel coordinate system, and the outputs are the voxel representations of the upper mold M_u and the lower mold

M_l . Unlike auxiliary components generation, which formulates the output space as a binary occupancy prediction (occupied or empty), upper and lower mold generation requires partitioning the space into two non-overlapping yet fully occupied volumetric results. To this end, we redefine the output space to include three categories: $\text{occupy}_{\text{upper}}$, $\text{occupy}_{\text{lower}}$, and empty. With this formulation, the upper and lower molds can be generated independently while ensuring spatial consistency.

Voxel Refinement Although direct operations in voxel space can produce reasonable coarse geometry, they are limited by low resolution and lack fine details, making post-processing more challenging. We employ a geometry flow model to refine the generation of auxiliary components, upper and lower molds. Building upon the approach in [73], we adapt the structural latent generation stage to refine voxel structures. By removing the original text/image conditions, we input only noisy voxel representations to generate structured latents $\mathbf{z}$. After training on our dataset, geometric features such as curves, planes, and edges are refined to more closely resemble CAD representations. The structured latents will decode into meshes $\mathcal{C}'_{\mathcal{M}}$ via decoders $\mathcal{D}_{\mathcal{M}}$ [73]. The decoding process is as follows:

$$\mathcal{D}_{\mathcal{M}}: \{(\mathbf{z}_i, \mathbf{p}_i)\}_{i=1}^L \rightarrow \{(\mathbf{w}_i^j, d_i^j)\}_{j=1}^{64}\}_{i=1}^L, \quad (2)$$

where $\mathbf{w}_i^j \in \mathbb{R}^{45}$ are the flexible parameters in FlexiCubes [62] and $d_i^j \in \mathbb{R}^8$ is signed distance values for the eight vertices of the corresponding voxel.

Mesh to CAD Although our generative model is able to predict the approximate structure of the auxiliary components and molds, the raw output often lacks precision. We introduce a post-processing stage to regularize these geometries based on the contact constraints imposed by the input P .

We first convert the raw prediction $\mathcal{C}'_{\mathcal{M}}$ into a watertight manifold $\tilde{\mathcal{C}}'_{\mathcal{M}}$ mesh using Poisson surface reconstruction (screened Poisson) [25]. To identify the orientation, we analyze the local neighborhood of P adjacent to $\tilde{\mathcal{C}}'_{\mathcal{M}}$ to estimate the dominant axis $\mathbf{d}$. This allows us to robustly segment $\tilde{\mathcal{C}}'_{\mathcal{M}}$ into an embedded portion and an external portion. For the embedded portion, we extract an intersection polyline or curve from the target cavity and then generate a swept “mating volume” along $\mathbf{d}$. For the external portion, we improve regularity by fitting standard geometry primitives via RANSAC [60]. The final shape is obtained by merging these segments via boolean composition. Simultaneously, we reconstruct the upper and lower mold (M_u, M_l) from the coarse outputs. We estimate a separating parting surface S_{part} (via planar or quadratic fitting) from the interface, then split an inflated bounding volume of the assembly along S_{part} . The final molds are carved by the part P and optimized auxiliary components $\tilde{\mathcal{C}}'_{\mathcal{M}}$ from the respective blocks. Finally all reconstructed models (M_u, M_l and $\tilde{\mathcal{C}}'_{\mathcal{M}}$) are optimized via ShapeUp [7], followed by quad-remeshing [22]. The resulting meshes are converted to NURBS surface patches [8] and exported as a STEP/STP file that supports operations and editing in standard CAD software.

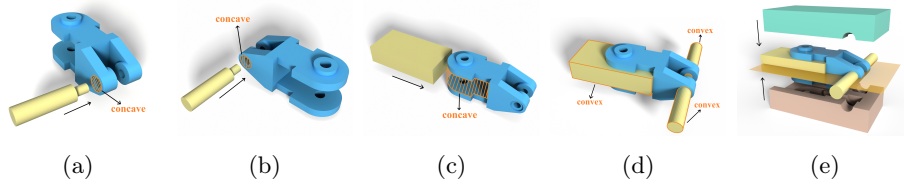


Fig. 4: Illustration of the processing pipeline for manual mold assembly design. (a)-(c) Analyze the part along a chosen demolding direction, localize concave (non-demoldable) regions and determine feasible auxiliary components. (d) Generate auxiliary component solids to address these localized collision issues. (e) Create a parting surface to separate upper and lower molds.

3.3 Orientation Estimation

Given an input CAD model, we need to estimate its optimal orientation for subsequent manufacturing processes. To this end, we employ a plane-detection-based method to generate a candidate set of orientations, $\mathcal{Pl} = \{Pl_1, \dots, Pl_m\}$, effectively discretizing the continuous pose space into a finite set of feasible solutions. Here, m denotes the total number of detected planar regions. Based on these candidate orientations, we train a binary classifier using manually curated positive and negative samples to select the most suitable orientation, as shown in Figure 3. Specifically, the ground-truth orientation is used to define positive samples: candidate orientations whose rotation axes are aligned with the ground-truth orientation are labeled as positive. In contrast, negative samples are constructed from plane-detection-derived candidate orientations whose rotation axes are not aligned with the ground-truth orientation. To address class imbalance, we apply data augmentation to both positive and negative samples, expanding each class to four times the scale of the original dataset. For the classifier architecture, we adopt the same voxel representation $V(P)$ and voxel encoder described in Section 3.2. The original decoder is replaced with a lightweight binary classifier, which functions similarly to an anomaly detector by distinguishing valid orientations from invalid ones.

4 DATASET

MoldCAD is a paired dataset of single-body parts and professionally designed injection-mold assemblies. Each sample links an input part to an output assembly that includes the upper and lower molds, a parting surface, and a variable number of auxiliary components required for demolding.

4.1 Data Sourcing and Filtering

We initialized our candidate pool using the ABC [29] and BRepNet [31] datasets, which provide a rich variety of industrial CAD models. However, raw

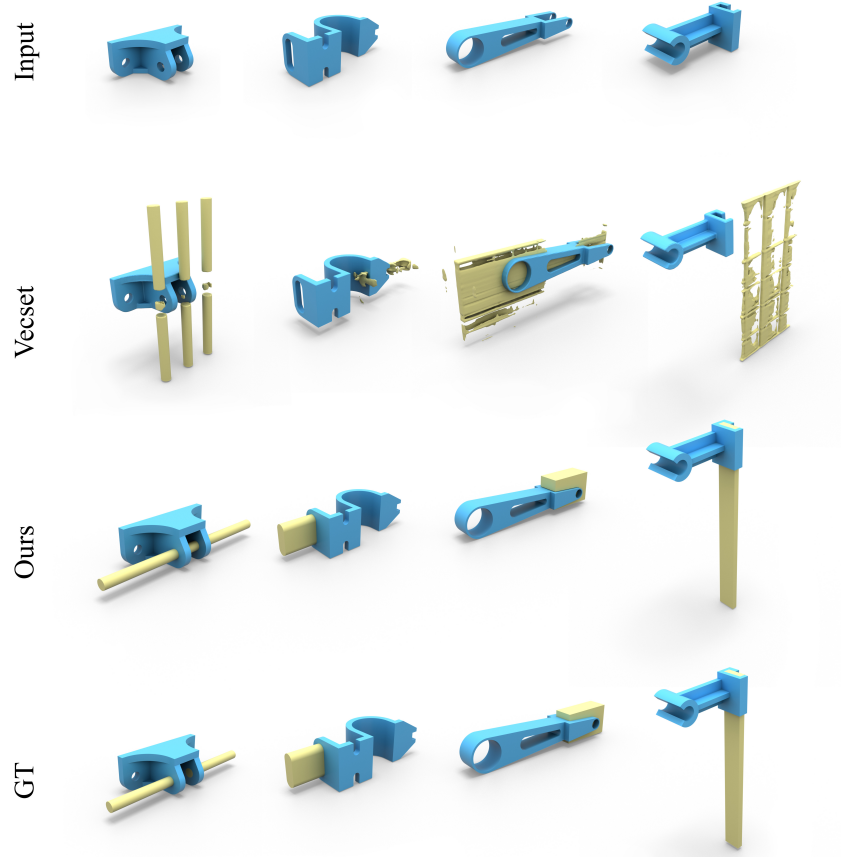


Fig. 5: Visual comparison on MoldCAD dataset.

data from these repositories often contains artifacts that cannot be manufactured by injection molding (e.g., non-demoldable geometries, multi-body parts) or simple geometries that do not require complex molding strategies. We first filtered single-body parts that satisfy basic injection molding prerequisites, such as water-tightness and appropriate wall thickness, and then explicitly retained and prioritized geometries with prominent undercuts and side holes. These features necessitate auxiliary molding components, rather than simple two-piece

Method	COV %↑	MMD ↓	JSD ↓	Method	COV %↑	MMD ↓	JSD ↓
VecSet-based	21.24	0.02900	0.105	VecSet-based	9.21	0.0252	0.277
Ours-only first stage	55.68	0.0140	0.054	Ours-only first stage	46.55	0.0096	0.073
AIMold	57.29	0.0147	0.050	AIMold	49.01	0.0091	0.061

Table 1: Auxiliary keys generation fidelity of different representations method.**Table 2:** Upper and lower mold generation fidelity of different representations method.

molds, thereby ensuring that the dataset captures the inherent challenges of actual manufacturing.

4.2 Professional Mold Design and Annotation

For each curated part, we commission professional mold engineers to design a complete injection-mold assembly following standard industrial practice. Figure 4 summarizes the typical workflow on a representative part with such features. Engineers first select a draft direction and analyze demolding capability along that direction to identify areas that might be mechanically locked during upper and lower mold separation. In practice, these problematic regions often appear as concave or side features relative to the mold opening motion. To resolve these geometric constraints, engineers design auxiliary components by determining feasible translation vectors and segmenting the geometry to ensure collision-free movement. Finally, the parting surface is constructed to split the mold into two halves. The resulting assemblies reflect real design decisions made in production settings and serve as precise ground truth for learning-based generation.

Each mold assembly is annotated at the instance level, providing distinct labels for the upper mold, lower mold, parting surfaces, and all auxiliary components. To handle the combinatorial complexity introduced by auxiliary mechanisms, MoldCAD explicitly preserves these parts as discrete entities rather than merging them into a single geometry. It enables models to learn discrete part generation rather than mere surface reconstruction and supports more granular evaluation metrics.

5 EXPERIMENTS

5.1 Implementation

For the two-stage training, we use our carefully constructed MoldCAD dataset. In the first-stage training, we adopt the AdamW optimizer [48] with a learning rate of 1×10^{-4} and follow [73] setup in second-stage geometry flow model training. For structure latents encoding, we render 12 images per CAD model. All experiments are conducted on NVIDIA A100 GPUs. The dataset is split into training and test sets with a ratio of 9:1, where the test set accounts for 10% of the total data and is used for evaluation.

Method	Accuracy % $\uparrow$	Recall % $\uparrow$	F1-Score $\uparrow$
Regressive-based	12.66	12.79	0.2267
Ours	91.17	89.65	0.9129

Table 3: Model orientation estimation accuracy of different method.

Method	COV % $\uparrow$	MMD $\downarrow$	JSD $\downarrow$
From Scratch	52.82	0.0173	0.0682
Less Render View	55.44	0.0146	0.050
Ours-only first stage	55.68	0.0140	0.054
Ours	57.29	0.0147	0.050

Table 4: Auxiliary keys generation fidelity ablation study.

5.2 Comparison

Quantitative Evaluation. Tables 1 and 2 report the averaged generation quality scores for auxiliary components, upper and lower molds on our Mold-CAD dataset. We primarily compare representative implicit methods operating in the latent space, which employ VecSet-based representation architectures like [68, 88]. These approaches have demonstrated strong performance on large-scale 3D datasets. However, in the context of mold design tasks, the lack of sufficiently large and diverse datasets makes it difficult for latent-space methods to achieve proper alignment, as evidenced by the results in Figure 5 and Table 1. AIMold consistently outperforms both baselines across almost all metrics, achieving better COV, MMD, and JSD scores (details in Appendix A). These results indicate improved generation quality and a closer match to the ground-truth data distribution. Furthermore, the second-stage voxel refinement leads to more accurate quantitative metrics and produces CAD models with more evident geometric features.

For orientation estimation, directly regressing a low-dimensional orientation vector from a high-dimensional 3D representation is highly challenging. As shown in Table 3, a naive regression-based approach fails to produce reliable results and exhibits very poor performance. In contrast, our plane-detection-based method significantly reduces the solution space by discretizing the continuous orientation domain into a small set of candidate orientations. This strategy effectively addresses the orientation estimation problem and provides a robust and accurate orientation initialization for the subsequent generation pipeline.

5.3 Ablations

Two-stage Generation To demonstrate the effectiveness of our generation approach, we conduct an ablation study to examine whether both stages of the pipeline are necessary. Specifically, we remove the second stage and retain only the first stage. This modification not only eliminates the ability to convert voxel representations into mesh, but also leads to a noticeable degradation in quantitative performance as shown in Table 4. In addition, we investigate a weakened variant of the second stage by reducing the number of DINO feature views used

for structural latent generation. This ablation further degrades geometric accuracy, indicating that sufficient multi-view features are critical for accurate structure modeling. The "Less Render View" setting uses only four views to construct the structural latent representations.

3D Encoder Pretraining To study the impact of large-scale 3D data pre-training for our task, we employ a 3D encoder [73] pretrained on a large-scale dataset such as [11]. As shown in Table 4, without pretraining, the performance clearly degrades, indicating that current large-scale 3D data provides a valuable prior for our mold design task.

5.4 Failure Cases

AIMold suffers from two common categories of failure cases: 1) Thin or small structures, which are challenging due to resolution limitations. Future improvements in resolution efficiency could help mitigate this issue. 2) Missing faces, resulting in non-watertight solids, a failure that is anticipated, as AIMold cannot guarantee watertight outputs. Optimization with representations that guarantee watertightness could serve as a potential solution.

6 LIMITATIONS AND FUTURE WORK

Although our MoldCAD dataset for the mold design task has already yielded promising results at its current scale, the scale of MoldCAD still exhibits a significant gap when compared to primarily single-body CAD datasets [29, 71] or open-vocabulary 3D datasets [11]. The scale of our curated dataset, MoldCAD, can be further improved with additional effort. We observe the trade-off in the number of render views used when transferring the mesh to the structured latent space, as shown in Table 4, which limits the capture of occlusion-related structural information. Exploring more efficient structure encoding methods or improved representations may help mitigate this trade-off.

7 CONCLUSION

We introduce AIMold, a conditional generation model for complex mold design. We present our newly collected MoldCAD dataset and conduct extensive experiments to validate the effectiveness of our method. By decomposing the complex design task into structured generation steps, we address the challenges inherent in mold design. We believe this approach demonstrates a promising pathway for predicting demolding orientations, identifying auxiliary components, and constructing parting surfaces, ultimately enabling the generation of a complete, manufacturing-ready mold assembly.

8 Acknowledgments

The work was supported in part by Guangdong ST Programme with Grant No. 2024B0101030002, the Basic Research Project No. HZQB-KCZYZ-2021067 of Hetao Shenzhen-HK ST Cooperation Zone, the Shenzhen Outstanding Talents Training Fund 202002, the NSFC with Grant No. 62293482, the Guangdong Provincial Key Laboratory of Future Networks of Intelligence (Grant No. 2022B1212010001), the Shenzhen Key Laboratory of Big Data and Artificial Intelligence (Grant No. SYSPG20241211173853027), the Guangdong Province Radio Science Data Center with grant No. 2025B1212070001, the National Key RD Program of China with grant No. 2018YFB1800800.

References

1. Alam, M.F., Ahmed, F.: Gencad: Image-conditioned computer-aided design generation with transformer-based contrastive representation and diffusion priors. arXiv preprint arXiv:2409.16294 (2024)
2. Alderighi, T., Malomo, L., Auzinger, T., Bickel, B., Cignoni, P., Pietroni, N.: State of the art in computational mould design. *Comput. Graph. Forum* **41**(6), 435–452 (2022). <https://doi.org/10.1111/CGF.14581>
3. Alderighi, T., Malomo, L., Bickel, B., Cignoni, P., Pietroni, N.: Volume decomposition for two-piece rigid casting. *ACM Trans. Graph.* **40**(6), 272:1–272:14 (2021). <https://doi.org/10.1145/3478513.3480555>
4. Alderighi, T., Malomo, L., Giorgi, D., Bickel, B., Cignoni, P., Pietroni, N.: Volume-aware design of composite molds. *ACM Trans. Graph.* **38**(4), 110:1–110:12 (2019). <https://doi.org/10.1145/3306346.3322981>
5. Alderighi, T., Malomo, L., Giorgi, D., Pietroni, N., Bickel, B., Cignoni, P.: Meta-molds: computational design of silicone molds. *ACM Trans. Graph.* **37**(4), 136 (2018). <https://doi.org/10.1145/3197517.3201381>
6. Banerjee, A.G., Gupta, S.K.: Geometrical algorithms for automated design of side actions in injection moulding of complex parts. *Computer-Aided Design* **39**(10), 882–897 (2007)
7. Bouaziz, S., Deuss, M., Schwartzburg, Y., Weise, T., Pauly, M.: Shape-up: Shaping discrete geometry with projections. In: *Computer Graphics Forum*. vol. 31, pp. 1657–1667. Wiley Online Library (2012)
8. Catmull, E., Clark, J.: Recursively generated b-spline surfaces on arbitrary topological meshes. In: *Seminal graphics: pioneering efforts that shaped the field*, pp. 183–188 (1998)
9. Chakraborty, P., Reddy, N.V.: Automatic determination of parting directions, parting lines and surfaces for two-piece permanent molds. *Journal of Materials Processing Technology* **209**(5), 2464–2476 (2009)
10. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: *IEEE/CVF International Conference on Computer Vision* (2022)
11. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems* **36**, 35799–35813 (2023)
12. Deng, Y., Yang, J., Xiang, J., Tong, X.: Gram: Generative radiance manifolds for 3d-aware image generation. In: *IEEE/CVF International Conference on Computer Vision* (2022)
13. Doñate, C.M., Paramio, M.R.: New methodology for demoldability analysis based on volume discretization algorithms. *Computer-Aided Design* **45**(2), 229–240 (2013)
14. Dupont, E., Cherenkova, K., Mallis, D., Gusev, G., Kacem, A., Aouada, D.: Trancad: A hierarchical transformer for cad sequence inference from point clouds. In: *European Conference on Computer Vision*. pp. 19–36. Springer (2024)
15. Fu, M.: The application of surface demoldability and moldability to side-core design in die and mold cad. *Computer-Aided Design* **40**(5), 567–575 (2008)
16. Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., Fidler, S.: Get3d: A generative model of high quality 3d textured shapes learned

- from images. *Advances In Neural Information Processing Systems* **35**, 31841–31854 (2022)
17. Guo, H., Huang, X., Bai, Y., Gan, H., Shi, Y., et al.: Brepdiff: Lightweight generation of complex b-rep with 3d gat diffusion. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 26587–26596 (2025)
 18. Herholz, P., Matusik, W., Alexa, M.: Approximating free-form geometry with height fields for manufacturing. *Comput. Graph. Forum* **34**(2), 239–251 (2015). <https://doi.org/10.1111/CGF.12556>
 19. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems* **33**, 6840–6851 (2020)
 20. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. In: *ICLR* (2024)
 21. Hou, B., Huang, Z., Zhou, H., Li, D.: A hybrid approach for automatic parting curve generation in injection mold design. *The International Journal of Advanced Manufacturing Technology* **95**(9), 3985–4001 (2018)
 22. Jakob, W., Tarini, M., Panozzo, D., Sorkine-Hornung, O.: Instant field-aligned meshes. *ACM transactions on graphics (TOG)* **34**(6), 1–15 (2015)
 23. Jayaraman, P.K., Lambourne, J.G., Desai, N., Willis, K.D., Sanghi, A., Morris, N.J.: Solidgen: An autoregressive model for direct b-rep synthesis. *arXiv preprint arXiv:2203.13944* (2022)
 24. Jayaraman, P.K., Sanghi, A., Lambourne, J.G., Willis, K.D., Davies, T., Shayani, H., Morris, N.: Uv-net: Learning from boundary representations. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 11703–11712 (2021)
 25. Kazhdan, M., Hoppe, H.: Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)* **32**(3), 1–13 (2013)
 26. Khan, M.S., Sinha, S., Sheikh, T.U., Stricker, D., Ali, S.A., Afzal, M.Z.: Text2cad: Generating sequential cad designs from beginner-to-expert level text prompts. *Advances in Neural Information Processing Systems* **37**, 7552–7579 (2024)
 27. Khan, M.S., Dupont, E., Ali, S.A., Cherenkova, K., Kacem, A., Aouada, D.: Cad-signet: Cad language inference from point clouds using layer-wise sketch instance guided attention. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4713–4722 (2024)
 28. Khardekar, R., Burton, G., McMains, S.: Finding feasible mold parting directions using graphics hardware. In: *Proceedings of the 2005 ACM symposium on Solid and Physical Modeling*. pp. 233–243 (2005)
 29. Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., Panozzo, D.: Abc: A big cad model dataset for geometric deep learning. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2019)
 30. Lai, Z., Zhao, Y., Zhao, Z., Liu, H., Lin, Q., Huang, J., Guo, C., Yue, X.: Lattice: Democratize high-fidelity 3d generation at scale. *arXiv preprint arXiv:2512.03052* (2025)
 31. Lambourne, J.G., Willis, K.D., Jayaraman, P.K., Sanghi, A., Meltzer, P., Shayani, H.: Brepnet: A topological message passing system for solid models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 12773–12782 (2021)
 32. Lee, M., Zhang, D., Jambon, C., Kim, Y.M.: Brepdiff: Single-stage b-rep diffusion model. In: *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. pp. 1–11 (2025)

33. Li, C., Pan, H., Bousseau, A., Mitra, N.J.: Sketch2cad: Sequential cad modeling by sketching in context. *ACM Transactions on Graphics (TOG)* **39**(6), 1–14 (2020)
34. Li, C., Pan, H., Bousseau, A., Mitra, N.J.: Free2cad: Parsing freehand drawings into cad commands. *ACM Transactions on Graphics (TOG)* **41**(4), 1–16 (2022)
35. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In: *ICLR* (2024)
36. Li, P., Guo, J., Li, H., Benes, B., Yan, D.M.: Sfmcad: Unsupervised cad reconstruction by learning sketch-based feature modeling operations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4671–4680 (2024)
37. Li, P., Guo, J., Zhang, X., Yan, D.M.: Secad-net: Self-supervised cad reconstruction by learning sketch-extrude operations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 16816–16826 (2023)
38. Li, P., Zhang, W., Chen, J., Yan, D.: Stitch-a-shape: Bottom-up learning for b-rep generation. In: *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. pp. 1–12 (2025)
39. Liang, Y., Yang, X., Lin, J., Li, H., Xu, X., Chen, Y.: Luciddreamer: Towards high-fidelity text-to-3d generation via interval score matching. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 6517–6526 (2024)
40. Lin, A.C., Quang, N.H.: Automatic generation of mold-piece regions and parting curves for complex cad models in multi-piece mold design. *Computer-Aided Design* **57**, 15–28 (2014)
41. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 300–309 (2023)
42. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747* (2022)
43. Liu, M., Shi, R., Chen, L., Zhang, Z., Xu, C., Wei, X., Chen, H., Zeng, C., Gu, J., Su, H.: One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 10072–10083 (2024)
44. Liu, M., Zeng, C., Wei, X., Shi, R., Chen, L., Xu, C., Zhang, M., Wang, Z., Zhang, X., Liu, I., et al.: Meshformer: High-quality mesh generation with 3d-guided reconstruction model. *arXiv preprint arXiv:2408.10198* (2024)
45. Liu, Y., Dutt, N.S., Li, C., Mitra, N.J.: B-repler: Language-guided editing of cad models. *arXiv preprint arXiv:2508.10201* (2025)
46. Liu, Y., Xu, D., Yu, X., Xu, X., Cohen-Or, D., Zhang, H., Huang, H.: Hola: B-rep generation using a holistic latent representation. *ACM Transactions on Graphics (TOG)* **44**(4), 1–25 (2025)
47. Liu, Y., Obukhov, A., Wegner, J.D., Schindler, K.: Point2cad: Reverse engineering cad models from 3d point clouds. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3763–3772 (2024)
48. Loshchilov, I.: Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017)
49. Ma, W., Xu, M., Li, X., Zhou, X.: Multicad: Contrastive representation learning for multi-modal 3d computer-aided design models. In: *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. pp. 1766–1776 (2023)

50. Malomo, L., Pietroni, N., Bickel, B., Cignoni, P.: Flexmolds: automatic design of flexible shells for molding. *ACM Trans. Graph.* **35**(6), 223:1–223:12 (2016). <https://doi.org/10.1145/2980179.2982397>
51. Mercado-Colmenero, J.M., Paramio, M., Perez-Garcia, J.M., Martin-Doñate, C.: A new hybrid method for demoldability analysis of discrete geometries. *Computer-Aided Design* **80**, 43–60 (2016)
52. Müller, N., Siddiqui, Y., Porzi, L., Bulo, S.R., Kotschieder, P., Nießner, M.: Diffrrf: Rendering-guided 3d radiance field diffusion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4328–4338 (2023)
53. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751* (2022)
54. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023)
55. Para, W., Bhat, S., Guerrero, P., Kelly, T., Mitra, N., Guibas, L.J., Wonka, P.: Sketchgen: Generating constrained cad sketches. *Advances in Neural Information Processing Systems* **34**, 5077–5088 (2021)
56. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: *ICLR* (2023)
57. Qi, D., Wang, C., Xu, J., Chu, T., Zhao, Z., Liu, W., Ding, W., Ma, Y., Gao, S.: Pointer-cad: Unifying b-rep and command sequences via pointer-based edges & faces selection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 17377–17387 (2026)
58. Ran, J., Fu, M.: Design of internal pins in injection mold cad via the automatic recognition of undercut features. *Computer-Aided Design* **42**(7), 582–597 (2010)
59. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 10684–10695 (2022)
60. Schnabel, R., Wahl, R., Klein, R.: Efficient ransac for point-cloud shape detection. In: *Computer graphics forum*. vol. 26, pp. 214–226. Wiley Online Library (2007)
61. Seff, A., Zhou, W., Richardson, N., Adams, R.P.: Vitruvion: A generative model of parametric cad sketches. *arXiv preprint arXiv:2109.14124* (2021)
62. Shen, T., Munkberg, J., Hasselgren, J., Yin, K., Wang, Z., Chen, W., Gojcic, Z., Fidler, S., Sharp, N., Gao, J.: Flexible isosurface extraction for gradient-based mesh optimization. *ACM Trans. Graph.* **42**(4) (jul 2023). <https://doi.org/10.1145/3592430>
63. Skorokhodov, I., Siarohin, A., Xu, Y., Ren, J., Lee, H.Y., Wonka, P., Tulyakov, S.: 3d generation on imagenet. *arXiv preprint arXiv:2303.01416* (2023)
64. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: *International Conference on Machine Learning*. pp. 2256–2265. PMLR (2015)
65. Stein, O., Jacobson, A., Grinspun, E.: Interactive design of castable shapes using two-piece rigid molds. *Comput. Graph.* **80**, 51–62 (2019). <https://doi.org/10.1016/J.CAG.2019.03.001>
66. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In: *ICLR* (2024)

67. Tang, J., Wang, T., Zhang, B., Zhang, T., Yi, R., Ma, L., Chen, D.: Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 22819–22829 (2023)
68. Team, F.A., Bhat, K., Khanna, N., Channa, K., Zhou, T., Zhu, Y., Sun, X., Shang, C., Sudarshan, A., Chu, M., Li, D., Deng, K., Fauconnier, J.P., Verhulsdonck, T., Agrawala, M., Fatahalian, K., Weiss, A., Reiser, C., Chirravuri, R.K., Kandur, R., Pelaez, A., Garg, A., Palleschi, M., Wang, J., Litz, S., Liu, L., Li, A., Harmon, D., Liu, D., Feng, L., Goupil, D., Kuczynski, L., Yoon, J., Marri, N., Zhuang, P., Zhang, Y., Yin, B., Jiang, H., van Workum, M., Lane, T., Erickson, B., Pathare, S., Price, K., Han, S., Wang, Y., Singh, A., Baszucki, D.: Cube: A roblox view of 3d intelligence (2025), <https://arxiv.org/abs/2503.15475>
69. Willis, K.D., Jayaraman, P.K., Chu, H., Tian, Y., Li, Y., Grandi, D., Sanghi, A., Tran, L., Lambourne, J.G., Solar-Lezama, A., et al.: Joinable: Learning bottom-up assembly of parametric cad joints. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15849–15860 (2022)
70. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems* **29** (2016)
71. Wu, R., Xiao, C., Zheng, C.: Deepcad: A deep generative network for computer-aided design models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6772–6782 (2021)
72. Xiang, J., Chen, X., Xu, S., Wang, R., Lv, Z., Deng, Y., Zhu, H., Dong, Y., Zhao, H., Yuan, N.J., et al.: Native and compact structured latents for 3d generation. *arXiv preprint arXiv:2512.14692* (2025)
73. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21469–21480 (2025)
74. Xu, J., Wang, C., Zhao, Z., Liu, W., Ma, Y., Gao, S.: Cad-mllm: Unifying multimodality-conditioned cad generation with mllm. *arXiv preprint arXiv:2411.04954* (2024)
75. Xu, X., Jayaraman, P., Lambourne, J., Liu, Y., Malpure, D., Meltzer, P.: Autobrep: Autoregressive b-rep generation with unified topology and geometry. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–12 (2025)
76. Xu, X., Jayaraman, P.K., Lambourne, J.G., Willis, K.D., Furukawa, Y.: Hierarchical neural coding for controllable cad model generation. *arXiv preprint arXiv:2307.00149* (2023)
77. Xu, X., Lambourne, J., Jayaraman, P., Wang, Z., Willis, K., Furukawa, Y.: Brep-gen: A b-rep generative diffusion model with structured latent geometry. *ACM Transactions on Graphics (TOG)* **43**(4), 1–14 (2024)
78. Xu, X., Willis, K.D., Lambourne, J.G., Cheng, C.Y., Jayaraman, P.K., Furukawa, Y.: Skexgen: Autoregressive generation of cad construction sequences with disentangled codebooks. *arXiv preprint arXiv:2207.04632* (2022)
79. Yavartanoo, M., Hong, S., Neshatavar, R., Lee, K.M.: Cnc-net: self-supervised learning for cnc machining operations. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9816–9825 (2024)
80. You, Y., Uy, M.A., Han, J., Thomas, R., Zhang, H., Du, Y., Chen, H., Engelmann, F., You, S., Guibas, L.: Img2cad: Reverse engineering 3d cad models from images through vlm-assisted conditional factorization. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–12 (2025)

81. Yu, F., Chen, Z., Li, M., Sanghi, A., Shayani, H., Mahdavi-Amiri, A., Zhang, H.: Capri-net: Learning compact cad shapes with adaptive primitive assembly. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11768–11778 (2022)
82. Yu, X., Xu, M., Zhang, Y., Liu, H., Ye, C., Wu, Y., Yan, Z., Zhu, C., Xiong, Z., Liang, T., et al.: Mvingnet: A large-scale dataset of multi-view images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9150–9161 (2023)
83. Zhang, B., Tang, J., Niessner, M., Wonka, P.: 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions on Graphics (TOG)* **42**(4), 1–16 (2023)
84. Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., Yu, J.: Clay: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Transactions on Graphics (TOG)* **43**(4), 1–20 (2024)
85. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3836–3847 (2023)
86. Zhang, X., Fang, G., Skouras, M., Gieseler, G., Wang, C.C.L., Whiting, E.: Computational design of fabric formwork. *ACM Trans. Graph.* **38**(4), 109:1–109:13 (2019). <https://doi.org/10.1145/3306346.3322988>
87. Zhang, Z., Sun, S., Wang, W., Cai, D., Bian, J.: Flexcad: Unified and versatile controllable cad generation with fine-tuned large language models. *arXiv preprint arXiv:2411.05823* (2024)
88. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., et al.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202* (2025)
89. Zhao, Z., Liu, W., Chen, X., Zeng, X., Wang, R., Cheng, P., Fu, B., Chen, T., Yu, G., Gao, S.: Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. *Advances in neural information processing systems* **36**, 73969–73982 (2023)
90. Zheng, X., Liu, Y., Wang, P., Tong, X.: Sdf-stylegan: Implicit sdf-based stylegan for 3d shape generation. In: *Computer Graphics Forum*. vol. 41, pp. 52–63 (2022)