

SldprtNet: A Large-Scale Multimodal Dataset for CAD Generation in Language-Driven 3D Design

Ruogu Li[†] Sikai Li[†] Yao Mu[‡] and Mingyu Ding[†]

Abstract—We introduce SldprtNet, a large-scale dataset comprising over 242,000 industrial parts, designed for semantic-driven CAD modeling, geometric deep learning, and the training/fine-tuning of multimodal models for 3D design. The dataset provides 3D models in both .step and .sldprt formats to support diverse training and testing. To enable parametric modeling and facilitate dataset scalability, we developed supporting tools, an encoder and a decoder, which support 13 types of CAD commands and enable lossless transformation between 3D models and a structured text representation. Additionally, each sample is paired with a composite image created by merging seven rendered views from different viewpoints of the 3D model, effectively reducing input token length and accelerating inference. By combining this image with the parameterized text output from the encoder, we employ the lightweight multimodal language model Qwen2.5-VL-7B to generate a natural language description of each part’s appearance and functionality. To ensure accuracy, we manually verified and aligned the generated descriptions, rendered images, and 3D models. These descriptions, along with the parameterized modeling scripts, rendered images, and 3D model files, are fully aligned to construct SldprtNet. To assess its effectiveness, we fine-tuned baseline models on a dataset subset, comparing image-plus-text inputs with text-only inputs. Results confirm the necessity and value of multi-modal datasets for CAD generation. It features carefully selected real-world industrial parts, supporting tools for scalable dataset expansion, diverse modalities, and ensured diversity in model complexity and geometric features, making it a comprehensive multimodal dataset built for semantic-driven CAD modeling and cross-modal learning.

I. INTRODUCTION

Computer-Aided Design (CAD) plays a critical role in mechanical design and manufacturing, offering significant advantages over traditional paper-based drafting. Unlike manual drawings, CAD allows intuitive visualization of a part’s shape and dimensions and simplifies modifications. SolidWorks, a powerful CAD platform, has become a default choice for many mechanical designers. The .sldprt format, SolidWorks’ native part file, records the feature operations and parameters used in model creation, enabling quick iteration and flexible editing of designs. This parametric, feature-based representation ensures higher precision and editability than discrete 3D formats like point clouds or meshes. Compared to other categories of 3D model datasets, CAD datasets are unique in that each sample must be manually created using specialized software. The high skill and time required for quality CAD modeling result in datasets that are much smaller in scale than image or text datasets. Furthermore, limited data quantity and quality, annotation difficulties, and

[†] University of North Carolina at Chapel Hill [‡] Shanghai Jiao Tong University

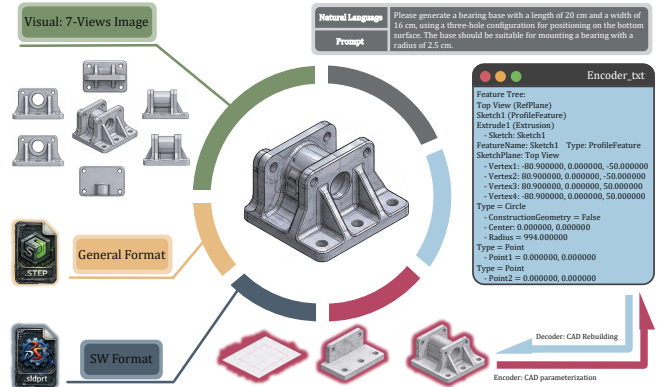


Fig. 1. Overview of SldprtNet dataset.

the lack of a standardized parametric representation format for 3D models have constrained progress in this area. As a result, despite the surge of interest in LLMs, research on semantic-driven CAD modeling tasks remains in its early stages. In this study, our main contributions are summarized as follows:

- **Dataset:** We construct a large-scale CAD dataset of over 242,000 high-quality samples. Each sample contains a multi-view composite image, a structured parametric text representation, and a natural language description of the part’s appearance and functionality. SldprtNet is specifically designed to support semantic-driven CAD modeling tasks and geometric deep learning applications.
- **Parametrization Tools:** Leveraging the SolidWorks API, we develop two tools, encoder and decoder (detailed in Section 4), to convert .sldprt files into parameterized text and to reconstruct 3D parts from these text scripts. These tools enable lossless transformation between CAD models and text, effectively bridging the gap between natural language and CAD modeling. Unlike prior works limited to simple sketch-and-extrude sequences, our tools support 13 types of CAD operations, greatly expanding the diversity of 3D models in the dataset.
- **Multimodal Data Alignment:** For each 3D model, we provide a composite image (merging six orthographic views and one isometric view) and the corresponding parametric modeling script (Encoder.txt). Using these as input, a multimodal language model (Qwen2.5-VL-7B) generates a natural language description (Des.txt) that captures the part’s visual appearance and functional semantics. This yields a fully aligned set of images, parametric scripts, and descriptions for every sample. Such a multimodal dataset offers rich

TABLE I
COMPARATIVE OVERVIEW OF MAJOR CAD AND 3D MODEL DATASETS

Dataset	SldprtNet	ABC	ShapeNet	ModelNet	Fusion 360	Thing10K
Models	242,606	1,000,000+	3,000,000+	151,128	50,000+	10,000
Format	Sldprt	B-Rep	Mesh	Mesh	B-Rep	Mesh
Parametric	✓	✓	×	×	✓	×
Multi-view	✓	×	✓	×	×	×
Reconstructable	✓	×	×	×	×	×
Model Description	✓	×	×	×	×	×

supervision signals and a strong foundation for advancing semantic-driven CAD modeling research.

II. RELATED WORK

Text-to-CAD modeling lies at the intersection of 3D geometry processing, CAD, and natural language understanding. While recent advances in mesh-based shape generation and vision-language pre-training have made notable progress, semantic-driven datasets and models specifically designed for CAD modeling remain limited. We briefly review mainstream 3D model datasets and recent work on language-guided CAD generation, and analyze their limitations. A comparative summary of representative datasets is provided in Table I.

• **Non-Parametric 3D Datasets:** Early large-scale datasets such as ModelNet [1] and ShapeNet [2] are widely adopted in vision and graphics research. They represent 3D geometry using meshes or point clouds, capturing only final surface shapes without retaining any feature-based modeling information. For example, ModelNet includes 150,000 models across various object categories, while ShapeNet provides 3 million with rich class labels and annotations. Thing10K [3] offers 10,000 printable models with diverse styles. Point-cloud-specific datasets such as ModelNet40, ScanNet [4], SemanticKITTI [5], and ShapeNetPart have become benchmarks for perception [6], reconstruction, and segmentation. However, none of these—whether mesh or point cloud—preserve design history, parametric operations, or sketch-based features. As a result, they lack parametric editability and cannot support semantically meaningful modifications. For instance, dimensions or constraints in these formats cannot be adjusted directly. While effective for classification, segmentation, and rendering, these representations are inherently limited for CAD automation, parameterized modeling, and language-guided design reasoning.

• **Parametric CAD Datasets:** To address the limitations of purely geometric datasets, several parametric CAD datasets have emerged. The ABC dataset contains over one million B-rep CAD models with precise surface and boundary representations, making it valuable for geometric learning [7]. However, it lacks modeling sequences and any textual annotations, which limits its use for sequential or language-driven generation tasks. The Fusion 360 Gallery dataset [8] includes real-world CAD modeling histories (e.g., sketches and extrusions), offering step-by-step modeling sequences. Nevertheless, it focuses primarily on basic extrusion-based

models and lacks semantic-level textual annotations. Thus, while these datasets improve geometric precision, they remain insufficient for tasks requiring semantic understanding or language-based design generation.

• **Language-Guided CAD Generation:** Recent efforts have explored CAD generative modeling using language or sequential modeling approaches. DeepCAD introduced a transformer-based network to model sequences of CAD commands [9]. It was trained on 170,000 CAD sequences (a subset of ABC with modeling histories) and pioneered framing CAD modeling as a language modeling problem. However, DeepCAD was limited to only two command types (2D sketching and extrusion), lacked support for complex feature workflows, and did not incorporate natural language input. To bridge this gap, Text2CAD proposed generating CAD models guided by textual prompts [10]. It augmented DeepCAD’s data with synthetic language descriptions (ranging from beginner to expert level) to form paired examples of CAD sequences and text. While this established a link between CAD operations and natural language, the synthetic nature of the text can lead to semantic misalignment with the actual geometry. Moreover, the absence of any image modality in Text2CAD’s data restricts perceptual grounding and limits generalization.

• **Toward Multimodal CAD Learning:** Most existing CAD datasets focus on a single modality—geometry only (e.g., ABC), modeling sequence only (e.g., DeepCAD), or text prompts only (e.g., Text2CAD). In contrast, modern multimodal models like CLIP [11], Flamingo [12], and BLIP-2 [13] have shown that aligned cross-modal learning is key to generalization, cross-task transfer, and zero-shot reasoning. In the CAD domain, a multimodal dataset that combines precise geometry (3D CAD models), rendered images (multi-view projections), structured modeling sequences (parametric CAD commands), and natural language descriptions can significantly enhance a model’s capability in understanding and generation. For example, Flamingo demonstrates strong few-shot performance on vision-language tasks by leveraging cross-modal alignment during training, while BLIP-2 shows that even a limited amount of paired data can be leveraged effectively via modality bridging. Therefore, for the Text-to-CAD task, constructing a unified multimodal dataset that integrates images, natural language, 3D geometry, and modeling sequences not only provides richer supervision but also lays a foundation for general-purpose pretraining, improving downstream performance and transferability.

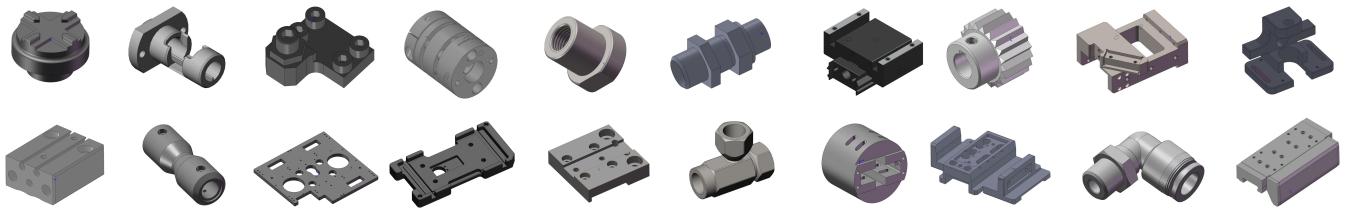


Fig. 2. Twenty randomly sampled CAD parts from SldprtNet, illustrating its industrial orientation and geometric diversity across varied mechanical components.

Beyond the aforementioned datasets, several recent works have attempted to extend large language models toward CAD generation, such as CAD-GPT [14], CAD-MLLM [15], CAD-Coder [16], and CAD-Llama [17]. While these studies demonstrate the potential of multimodal or code-driven approaches, they still exhibit notable limitations in key aspects. CAD-GPT emphasizes the spatial reasoning capabilities of multimodal LLMs, but its data is only a small-scale extension of DeepCAD, restricted to sketching and extrusion commands, and lacks large-scale industrial coverage. CAD-MLLM introduces the modality-rich OmniCAD dataset, yet a significant portion of its samples are synthetically generated rather than real engineering parts, and it does not provide executable modeling sequences for full reconstruction. CAD-Coder is the first open-source vision–language model capable of translating images into CadQuery code, but its reliance on the lightweight CadQuery DSL deviates from industrial CAD workflows, and its dataset size (163k) and command coverage remain limited. CAD-Llama proposes a structured parametric CAD code (SPCC) and hierarchical annotation, but it is primarily evaluated on synthetic or small-scale subsets, and lacks multimodal alignment such as rendered views or natural language paired with real-world parts.

III. DATASET

A. Principles and Requirements

Given the challenges in Text-to-CAD modeling and the growing demands of multimodal generative tasks, we revisit and extend the classical dataset criteria proposed in the ABC dataset [18]. While traditional benchmarks emphasize large scale, geometric fidelity, parametric representation, and category balance, they often fall short in supporting language-driven, editable, and multimodal modeling workflows [19]. To bridge this gap, we define five expanded requirements for next-generation CAD datasets:

- **Multimodality:** CAD modeling inherently involves geometry, procedural logic, visual perception, and natural language [20]. However, most existing datasets provide only geometry or command sequences. For instance, ABC dataset lacks language or visual content, while Text2CAD omits rendered images. An ideal dataset should integrate 3D models, multi-view images, structured modeling instructions, and natural language descriptions to enable joint learning across modalities and support tasks such as text-guided modeling or visual-

to-CAD generation.

- **Bidirectional Representability:** Unlike static output tasks in vision or NLP, CAD modeling emphasizes procedural traceability and logical consistency. An ideal dataset should support both forward modeling, which converts structured instructions into 3D geometry, and reverse modeling, which derives modeling commands from geometry or images. This bidirectional design enables closed-loop training, error detection, synthetic data generation, and controllable model editing [21].

- **Semantic Annotation:** Unlike image datasets with clear class boundaries, CAD parts are shaped by engineering intent, which makes function-level semantics more meaningful than conventional category labels. Annotating parts with functional roles, such as shaft, support, or guide rail, facilitates tasks such as part retrieval, function-aware generation, and abstraction. [22], [23].

- **Editability:** Static mesh datasets do not support structured editing. CAD datasets should retain parametric information and feature hierarchies to enable flexible modifications through standard tools or code-based interfaces. This supports use cases like language-driven editing, prompt tuning, and iterative design refinement.

- **Human-Readable Format:** Modeling scripts should use interpretable syntax, clear feature hierarchies, and consistent naming. This not only enhances debugging and annotation but also facilitates human-AI collaboration and alignment with design intent [24].

B. Composition and Format

We curated SldprtNet by collecting over 242,000 industrial CAD part models from publicly available online repositories, including GrabCAD, McMaster-Carr, and FreeCAD [25]. Each sample in the dataset comprises four components:

- **3D Models:** Each part is provided in two formats.

- (1) sldprt the native SolidWorks part file, obtained via manual modeling. It encodes the feature-tree history (commands, parameters, and modeling sequence) and allows lossless reconstruction and feature-level semantic extraction for our purposes.

- (2) step – a neutral, standardized 3D CAD exchange format that supports cross-platform interoperability. We include this for validation and use with other CAD software.

- **Images:** For each CAD model, we render seven PNG images from different standard views: six orthographic (front,

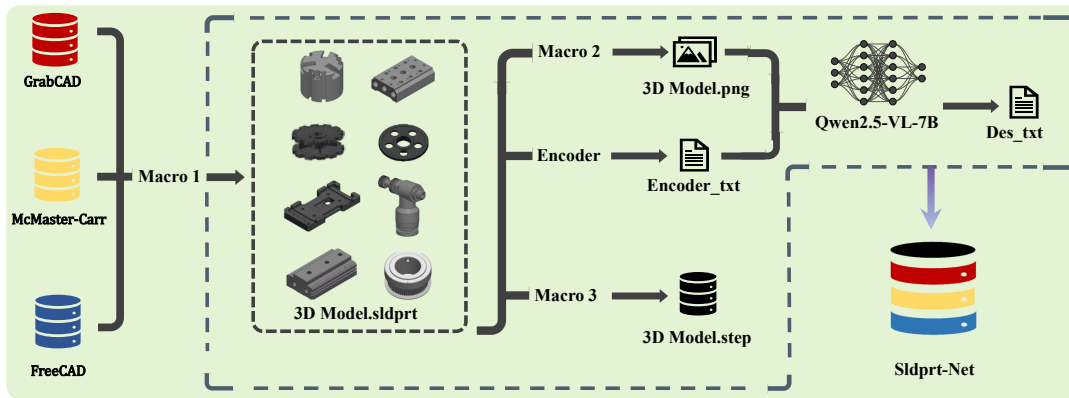


Fig. 3. Overview of the automated data processing pipeline for constructing SldprtNet.

back, left, right, top, bottom) and one isometric. These multi-view images comprehensively capture the 3D geometry in 2D form and serve as visual inputs for vision-language models. To optimize token usage during multimodal model inference, we composite all seven views into a single image. This reduces input length and accelerates processing without compromising visual completeness.

- **Parametric Text:** Using our custom Encoder tool (Section 4), each .sldprt file is converted into a structured text script encoding the CAD modeling instructions. The output (Encoder.txt) is a human- and machine-readable description of the model’s construction. Table II shows an excerpt of this format, including (1) a Feature Tree listing all features used (e.g., Extrude, Fillet) in modeling order with their names and parent-child relationships, and (2) detailed parameters for each feature (dimensions, constraints, sketch entities, dependencies). This structured format exposes the modeling logic explicitly and facilitates learning of CAD sequences by generative models [26], [27].

- **Natural Language Descriptions:** Unlike prior works (e.g., Text2CAD) that rely on modeling sequences or sketches alone to generate text, our dataset includes full visual information for each part [28]. We employ a multimodal LLMs, Qwen2.5-VL-7B, providing it both the composite rendered image and the parametric text as input to produce a descriptive caption for the part [28]. The inference process was run on 12 NVIDIA A100 GPUs for 368 GPU-hours to generate more than 242,000 parts descriptions, each capturing both functional and geometric aspects of the design. Qwen2.5-VL-7B uses a vision transformer to integrate visual features with the modeling text, enabling fine-grained alignment between the geometry and the description [29], [30]. This allows the model to pick up details like hole patterns, contour outlines, and aspect ratios in context with the part’s functional intent [31]. The resulting descriptions are structurally consistent with the CAD modeling steps, clearly convey the part’s functionality and features, and remain well-aligned with the visual geometry of the model [32].

C. Processing Pipeline

To construct the multimodal SldprtNet dataset encompassing graphical, parametric, and textual modalities, we designed a fully automated pipeline for data acquisition and annotation in Fig. 3. First, we collected approximately 680,000 industrial part models in .sldprt format from the three public CAD platforms mentioned above using an automated script (Macro 1). We then filtered these to ensure modeling completeness and diversity: we retained over 242,000 models that contain at least one of 13 representative feature types (the supported feature list is detailed in Section 4) [33]. For each remaining model, we performed the following steps within SolidWorks:

- (1) Image Rendering (Macro 2): We rendered six orthographic views (front, back, left, right, top, bottom) and one isometric view of the part, then merged them into a single composite image (PNG). Fig. 6 shows an example of this multi-view composite.
- (2) Parametric Command Extraction (encoder): We invoked our command-line encoder tool to extract the modeling sequence from the .sldprt file. This produces the structured text file (Encoder.txt) described above, which serves as input to downstream learning models.
- (3) Standard Format Conversion (Macro 3): We converted the .sldprt file into the standard .step format for cross-platform geometry compatibility and validation.

To generate natural language descriptions aligned with both the appearance and function of the 3D model, we take the rendered 3D model image (3D Model.png) and the parametric modeling script (Encoder.txt) as inputs and perform inference using the Qwen2.5-VL-7B multimodal LLMs. Within Qwen2.5-VL-7B, a vision encoder aligns visual features with the parameterized modeling scripts across modalities, thereby generating semantically accurate and geometrically consistent natural language descriptions (Des.txt). As a result, each sample in SldprtNet contains five aligned modalities: the source model (.sldprt), the standard geometry model (.step), the rendered image (.png), the parametric modeling script (Encoder.txt), and the natural language description (Des.txt). This pipeline not only ensures consistency across modalities

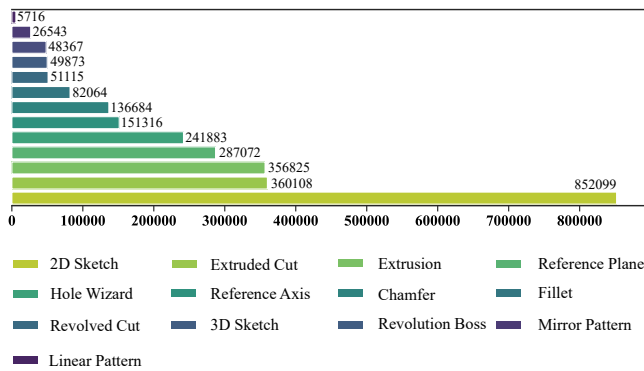


Fig. 4. Distribution of CAD model complexity in the SldprtNet dataset based on the number of feature operations per part. The chart presents four complexity levels with their corresponding proportions and sample counts.

but also enhances the extensibility and applicability of the dataset for a wide range of multimodal learning tasks.

D. Statistics and Analysis

Following the completion of data acquisition and preprocessing, we analyzed the Feature Tree information from the Encoder_txt files in the SldprtNet dataset [34]. Specifically, we extracted all the CAD feature types used across the 3D models, as well as their corresponding frequencies, to construct a comprehensive distribution chart, shown in Fig. 4. The chart summarizes the usage statistics of 13 core CAD features, including 2D Sketch, Extrusion, Linear Pattern, among others. The key observations are as follows:

- (1) 2D Sketch emerges as the most frequently used feature type, serving as the foundational step in nearly all models regardless of their complexity. This aligns with standard CAD workflows, where most 3D geometries originate from planar sketches.
- (2) Chamfer and Fillet—commonly employed in mechanical component design for edge treatment and stress reduction—also exhibit high usage frequencies, suggesting that the dataset predominantly consists of industrial-grade parts.
- (3) Linear Pattern and Mirror Pattern appear less frequently. These features are typically used for repetitive or symmetric geometries and can often be substituted through manual replication of base features. Their relatively low presence may also reflect a preference for explicit design over procedural patterning in the original models.

In addition to feature-type distribution, we conducted a complexity analysis of the dataset based on criteria adapted from DeepCAD [9], using the total number of modeling features (i.e., command operations) in each part as a proxy for its complexity. As shown in Fig. 5, to quantify model complexity, we analyze the number of unique CAD commands recorded in the Feature Tree of each part’s Encoder_txt file. This count reflects the number of sequential modeling steps involved in constructing the 3D geometry—higher values indicate more intricate design logic and geometric detail [35]. For instance, a model with a complexity score of 2 might involve two commands, such as Sketch and Extrude. Based

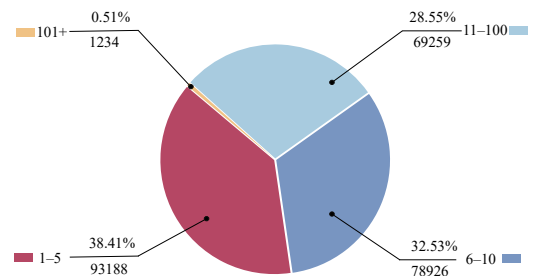


Fig. 5. Distribution of CAD features and total counts in the SldprtNet dataset. The figure summarizes the usage frequency of 13 parametric operations across all parts.

on this definition, we categorize model complexity into four distinct levels:

- **Level 1 (Simple):** 1–5 features, 93,188 samples.
- **Level 2 (Moderate):** 6–10 features, 78,926 samples.
- **Level 3 (Advanced):** 11–100 features, 69,259 samples.
- **Level 4 (Expert):** over 100 features, 1,234 samples.

Key insights from the complexity distribution include:

- (1) The majority of the dataset is composed of simple, moderate, and advanced models in relatively balanced proportions. This diversity ensures robust training coverage across varying levels of modeling intricacy, making SldprtNet well-suited for multi-stage generative modeling tasks.
 - (2) Although expert-level models are rare, they represent highly intricate CAD workflows—often involving nested assemblies, complex constraints, and hybrid features. These models provide critical benchmarks for evaluating the reasoning depth and sequence modeling capacity of multimodal language models such as LLMs or vision-language transformers [36].
 - (3) This hierarchical complexity taxonomy also supports stratified evaluation of downstream models. Given the wide variance in geometric logic and command length across complexity levels, performance can be better assessed through complexity-specific benchmarks.
 - (4) Moreover, the tiered classification offers a practical foundation for implementing curriculum learning. Initial model training may begin with simpler instances to bootstrap geometric comprehension, followed by gradual exposure to higher-complexity samples. This staged approach can improve convergence rates and model generalization, especially in long-horizon CAD sequence generation [37], [38].
- In summary, SldprtNet demonstrates a high degree of diversity in feature types and structural complexity, offering both broad coverage and deep granularity across the dataset.

E. Base Line

To evaluate the effectiveness of the multimodal dataset in CAD generation tasks, we fine-tuned two baseline models on a 50K-sample subset of the SldprtNet: Qwen2.5-7B (trained with Encoder_text only) and Qwen2.5-7B-VL (trained with both image and Encoder_text). As shown in Table III, the multimodally trained Qwen2.5-7B-VL outperforms the text-only Qwen2.5-7B across several key metrics. Specifically,

TABLE II
EXAMPLE OF PARAMETRIC TEXT REPRESENTATION

Feature Tree:
Top View (RefPlane)
Sketch1 (ProfileFeature)
Extrude1 (Extrusion)
- Sketch: Sketch1

FeatureName: Top View Type: RefPlane
- Vertex1: -80.900000, 0.000000, -50.000000
- Vertex2: 80.900000, 0.000000, -50.000000
- Vertex3: 80.900000, 0.000000, 50.000000
- Vertex4: -80.900000, 0.000000, 50.000000

FeatureName: Sketch1 Type: ProfileFeature
SketchPlane: Top View
- Vertex1: -80.900000, 0.000000, -50.000000
- Vertex2: 80.900000, 0.000000, -50.000000
- Vertex3: 80.900000, 0.000000, 50.000000
- Vertex4: -80.900000, 0.000000, 50.000000
Type = Circle
- ConstructionGeometry = False
- Center: 0.000000, 0.000000
- Radius = 994.000000
Type = Point
- Point1 = 0.000000, 0.000000
Type = Point
- Point2 = 0.000000, 0.000000

FeatureName: Extrude1 Type: Extrusion
sd: True
flip: False
dir: False
t1: 6
t2: 0
d1: 150.000000
d2: 0.000000
dang1: 0
dang2: 0
merge: True
useFeatScope: True
useAutoselect: False
t0: 0
startOffset: 0
flipStartOffset: False

it achieves higher scores in Exact Match Score (0.0099 vs. 0.0058), Command-Level F1 (0.3670 vs. 0.3247), and Partial Match Rate (0.6162 vs. 0.5554), indicating a stronger alignment between the generated CAD commands and ground-truth modeling instructions. These results demonstrate that incorporating rendered geometric views as an auxiliary modality enhances the model’s understanding of geometric semantics and procedural logic.

Although the text-only model slightly outperforms in Parameter Tolerance Accuracy (0.5016 vs. 0.4630), this may suggest a tendency toward overfitting on numeric values rather than capturing structural semantics. The baseline experiments are reproducible, and we will release all relevant code and dataset subsets through the anonymous repository referenced in the Code section. Overall, these findings validate the effectiveness of multimodal supervision in improving the performance of language-driven CAD generation models, and further underscore the importance of richly annotated datasets like SldprtNet for advancing research in generative CAD modeling.

TABLE III
BASELINE MODEL PERFORMANCE COMPARISON

Metric	Qwen2.5-7B	Qwen2.5-7B-VL
<i>Exact Match Score</i>	0.0058	0.0099
<i>BLEU Score</i>	97.1827	97.9309
<i>Test Samples</i>	3644	3644
<i>Command-Level F1</i>	0.3247	0.3670
<i>Tolerance Accuracy</i>	0.5016	0.4630
<i>Partial Match Rate</i>	0.5554	0.6162

IV. SUPPORTED APPLICATION

This section focuses on two key tools developed during the construction of the SldprtNet dataset, namely the encoder and decoder, as well as the dataset’s applicability to the representative task of Text-to-CAD generation.

A. Tools

The encoder and decoder are two essential auxiliary tools developed as part of the dataset pipeline. Built upon the COM interface provided by SolidWorks and implemented via macro-based scripting, these tools enable automatic traversal and reconstruction of .sldprt part models. Together, they support 13 common CAD feature types and allow for a lossless, bidirectional transformation between parameterized CAD models and structured textual representations. Their implementations are as follows:

• **Encoder: CAD → Text:**

(1) Upon execution, the encoder automatically traverses the Feature Tree of an .sldprt file, extracting the feature types, feature names, and parent-child relationships in the order dictated by the modeling history [39].

An example is provided in the first five rows of Table II. (2) Based on the feature types retrieved from the Feature Tree, the encoder sequentially calls corresponding internal modules to extract detailed parameters for each feature. These parameter details are illustrated in Appendices Table II following the initial rows.

(3) For each supported feature type, a consistent, human-readable, and machine-interpretable text format is adopted. After completing the above steps, the encoder generates an Encoder.txt file, completing the parametric representation of the original .sldprt file.

• **Decoder: Text → CAD:**

(1) Upon execution, the decoder first initializes the modeling environment by creating a blank .sldprt part document.

(2) It then reads the encoder.txt file and parses the entire Feature Tree section. Guided by the feature order and parent-child hierarchy, it sequentially accesses each feature’s parameters and invokes the SolidWorks API to create the corresponding feature within the modeling environment.

(3) Following the original modeling sequence, the decoder reconstructs each feature step-by-step, ensuring geometric and topological consistency with the source model.

Together, the encoder and decoder form a closed-loop modeling system, enabling the round-trip transformation from model to instruction and back to model. This capability

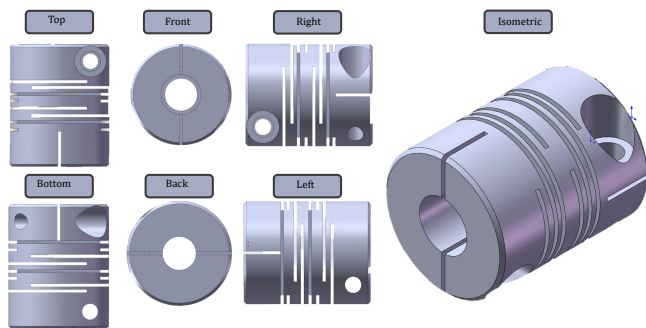


Fig. 6. Composite image of a CAD model from seven standard views, including top, bottom, left, right, front, back, and isometric.

provides structural verification, supports data augmentation, and opens avenues for the expansion of SldprtNet in future applications [40].

B. Capability in Supporting Text-to-CAD Tasks

Based on its design and supporting tools, SldprtNet offers robust capabilities for Text-to-CAD tasks:

(1) Natural Language–Driven Parametric Modeling: The encoder converts CAD models into structured, semantically meaningful command sequences, mapping natural language descriptions to modeling operations [8]. During training, models learn the alignment between textual inputs and CAD procedures, and during inference, the language model maps natural language input (e.g., “Create a bracket with four threaded holes”) into structured CAD commands [41], executed by the decoder to generate the model. This “language → instruction → geometry” pipeline enhances the modeling process by capturing both intent and execution logic, significantly expanding the operational capacity of language in design [42].

(2) Multimodal Input and Cross-Modal Modeling: Each sample in SldprtNet contains rendered images, modeling instruction text, and natural language descriptions [43], making it a rich input source for vision-language models (VL-LLMs) [44]. This enables tasks such as image-text alignment, multi-view geometric reasoning, and cross-modal prompt learning.

(3) Controllable Modeling and Parameter-Level Editing: The encoder’s structured output provides semantic clarity, allowing the decoder to execute model-generated instructions directly [45]. This enables not only model creation from scratch but also fine-grained, controllable edits to existing models [46], such as adjusting dimensions or directions [47].

(4) Complexity-Based Evaluation Mechanism: Models are categorized by the number of CAD features into complexity tiers (1–5, 6–10, 11–100, 101+), supporting curriculum-based training, benchmarking, and skill evaluation across Text-to-CAD tasks.

(5) Interpretability and Reverse Engineering Support: The closed-loop conversion between encoder and decoder enables semantic verification and geometric reconstruction of command sequences, ensuring both linguistic accuracy and geometric fidelity, thus improving modeling process explainability.

V. CONCLUSION

In this study, we presented SldprtNet, a large-scale multimodal CAD dataset designed to advance research in semantic-driven 3D modeling, geometric deep learning, and cross-modal learning. Each of the 242,000+ samples includes a CAD model in .step and .sldprt formats, a multi-view composite image, a parameterized modeling script, and a description of the part’s appearance and function, capturing both precise geometry and design intent. By aligning these modalities, SldprtNet offers a rich training ground for models connecting language, vision, and CAD geometry [48].

To assess its effectiveness, we conducted baseline experiments comparing unimodal (Encoder_text only) and multimodal (image + Encoder_text) models using various evaluation metrics. Results clearly show that the multimodal dataset yields superior performance in CAD generation tasks.

We further augmented the dataset with custom encoder and decoder tools, enabling lossless conversion between CAD models and textual instructions. This supports fully parametric and interpretable text-to-CAD modeling: language models can output executable CAD operations, which can be converted back into text for verification. SldprtNet thus facilitates semantic-driven CAD generation, where high-level concepts in natural language are translated into precise 3D designs [49]. We believe the dataset and toolchain will support future research in language-based CAD generation, cross-modal engineering understanding, and human-AI collaborative design. In future work, we aim to expand the dataset’s scale and feature coverage, and evaluate its utility in real-world text-to-CAD scenarios to further enable intelligent 3D design automation.

REFERENCES

- [1] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3d shapenets: A deep representation for volumetric shapes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1912–1920.
- [2] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, “Shapenet: An information-rich 3d model repository,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [3] Q. Zhou and A. Jacobson, “Thing10k: A dataset of 10,000 3d-printing models,” *arXiv preprint arXiv:1605.04797*, 2016.
- [4] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, “Scannet: Richly-annotated 3d reconstructions of indoor scenes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 9297–9307.
- [5] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, “Semantickitti: A dataset for semantic scene understanding of lidar sequences,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9297–9307.
- [6] L. Yi, H. Su, X. Guo, and L. J. Guibas, “Syncspec-cnn: Synchronized spectral cnn for 3d shape segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6584–6592.
- [7] S. Koch, H. Su, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo, “Abc: A big cad model dataset for geometric deep learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9601–9611.

- [8] K. D. D. Willis, Y. Pu, J. Luo, H. Chu, T. Du, J. G. Lambourne, A. Solar-Lezama, and W. Matusik, "Fusion 360 gallery: A dataset and environment for programmatic cad reconstruction," *arXiv preprint arXiv:2006.10211*, 2021.
- [9] R. Wu, C. Xiao, and C. Zheng, "Deepcad: A deep generative network for computer-aided design models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 6772–6782.
- [10] M. A. Khan, S. Sinha, T. U. Sheikh, D. Stricker, S. A. Ali, and M. Z. Afzal, "Text2cad: Generating sequential cad models from beginner to expert level text prompts," in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 7552–7579.
- [11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," *arXiv preprint arXiv:2103.00020*, 2021.
- [12] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, *et al.*, "Flamingo: A visual language model for few-shot learning," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 23 716–23 736.
- [13] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.
- [14] J. Xu, Z. Li, Y. Chen, and H. Zhang, "Cad-mlm: Unifying multimodality-conditioned cad generation with mlm," *arXiv preprint arXiv:2411.04954*, 2024.
- [15] A. Sanghi, P. K. Jayaraman, J. G. Lambourne, and K. D. D. Willis, "Editnet: Editable neural implicit representations for parametric shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 160–176.
- [16] S. Wang, M. Liu, Q. Zhao, and J. Sun, "Cad-gpt: Synthesising cad construction sequence with spatial reasoning-enhanced multimodal llms," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 8, 2025.
- [17] A. C. Doris, R. Patel, K. Huang, and T. Lee, "Cad-coder: An open-source vision-language model for computer-aided design code generation," *arXiv preprint arXiv:2505.14646*, 2025.
- [18] B. R. Hunde and A. D. W. Woldeyohannes, "Future prospects of computer aided design (cad): A review from the perspective of artificial intelligence (ai), extended reality, and 3d printing," *Results in Engineering*, vol. 14, p. 100478, 2022.
- [19] E. Dupont, K. Cherenkova, A. Kacem, S. A. Ali, I. Aryhannikov, G. Gusev, and D. Aouada, "Cadops-net: Jointly learning cad operation types and steps from boundary representations," in *International Conference on 3D Vision (3DV)*, 2022, pp. 1–10.
- [20] J. G. Lambourne, K. D. D. Willis, P. K. Jayaraman, V. Liu, H. Shayani, A. H. Khasahmadi, S. Sridhar, and D. Ritchie, "Clip sculptor: Zero shot generation of high fidelity and diverse shapes from natural language," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 18 339–18 349.
- [21] W. Para, S. Bhat, P. Guerrero, T. Kelly, N. Mitra, L. J. Guibas, and P. Wonka, "Sketchgen: Generating constrained cad sketches," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 5077–5086.
- [22] B. Manda, S. Dhayarkar, S. Mitheran, V. K. Vekash, and R. Muthuganapathy, "CADSketchNet—an annotated sketch dataset for 3d cad model retrieval with deep neural networks," *Computers & Graphics*, vol. 99, pp. 100–113, 2021.
- [23] Y. Chen, S. Wang, Y. Liu, X. Wang, Y. Qiao, and W. Zuo, "Transunet: Transformers make strong encoders for medical image segmentation," *arXiv preprint arXiv:2102.04306*, 2021.
- [24] Y. Yamamoto and K. Nakakoji, "Interaction design of tools for fostering creativity in the early stages of information design," *International Journal of Human-Computer Studies*, vol. 63, no. 4, pp. 513–535, 2005.
- [25] F. Machado, N. Malpica, and S. Borromeo, "Parametric cad modeling for open source scientific hardware: Comparing openscad and freecad python scripts," *PLOS ONE*, vol. 14, no. 12, p. e0225795, 2019.
- [26] K. Cherenkova, D. Aouada, and G. Gusev, "Pvdeconv: Point-voxel deconvolution for autoencoding cad construction in 3d," in *Proceedings of the International Conference on 3D Vision (3DV)*, 2020, pp. 2741–2745.
- [27] S. Hong, J. Oh, H. Kim, D. Yoo, and I. S. Kweon, "Align your gaussians: Text-driven 3d gaussian splatting," *arXiv preprint arXiv:2310.08597*, 2023.
- [28] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero shot text guided object generation with dream fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 857–866.
- [29] B. Poole, A. Jain, B. Mildenhall, M. Tancik, P. Abbeel, and J. T. Barron, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.
- [30] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 10012–10022.
- [31] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.
- [32] Y. He, Y. Bai, M. Lin, W. Zhao, Y. Hu, J. Sheng, R. Yi, J. Li, Y.-J. Liu, and Y.-J. Liu, "T3bench: Benchmarking current progress in text-to-3d generation," *arXiv preprint arXiv:2310.02977*, 2023.
- [33] D. Mallis, A. S. Aziz, E. Dupont, K. Cherenkova, A. Kacem, S. S. Karadeniz, M. S. Khan, G. Gusev, and D. Aouada, "Sharp challenge 2023: Solving cad history and parameters recovery from point clouds and 3d scans," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 1786–1795.
- [34] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.
- [35] V. Shapiro and D. L. Vossler, "Construction and optimization of csg representations," *Computers & Industrial Engineering*, vol. 23, no. 1–2, pp. 4–20, 1991.
- [36] X. Chen, Y. Zhang, W. Li, Z. Yao, H. Wang, X. Xu, and M. Soltani, "Polygen: An autoregressive generative model of 3d meshes," in *International Conference on Machine Learning (ICML)*, 2022, pp. 7220–7229.
- [37] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, and D. Zhou, "Chain of thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.
- [38] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Computation*, vol. 1, no. 2, pp. 270–280, 1989.
- [39] A. Seff, Y. Ovdadia, W. Zhou, and R. Adams, "Sketchgraphs: A large-scale dataset for modeling relational geometry in computer-aided design," *arXiv preprint arXiv:2007.08506*, 2020.
- [40] W. Mao, Y. Huang, Y. Xu, S. Wang, and X. Zuo, "Parametric model-aware cad reconstruction from sparse point clouds," in *European Conference on Computer Vision (ECCV)*, 2022, pp. 131–148.
- [41] X. Xu, P. K. Jayaraman, K. D. D. Willis, L. J. Guibas, and A. Fu, "Hierarchical neural coding for controllable cad model generation," *arXiv preprint arXiv:2307.00149*, 2023.
- [42] D. Ritchie, P. Guerrero, R. K. Jones, N. J. Mitra, A. Schulz, K. D. D. Willis, and J. Wu, "Neurosymbolic models for computer graphics," *Computer Graphics Forum*, vol. 42, no. 3, pp. 545–568, 2023.
- [43] A. Seff, W. Zhou, N. Richardson, and R. Adams, "Vitruvian: A generative model of parametric cad sketches," in *International Conference on Learning Representations (ICLR)*, 2021.
- [44] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3d: High-resolution text-to-3d content creation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [45] Y. Ganin, S. Bartunov, Y. Li, E. Keller, and S. Saliceti, "Computer-aided design as language," *Advances in Neural Information Processing Systems*, vol. 34, pp. 234–245, 2021.
- [46] N. J. Mitra, A. Singh, and M. Pauly, "Structure-aware shape processing," *ACM Computing Surveys*, vol. 45, no. 2, pp. 1–23, 2013.
- [47] T. Du, J. P. Inala, Y. Pu, A. Spielberg, A. Schulz, D. Rus, A. Solar-Lezama, and W. Matusik, "Inversecsg: Automatic conversion of 3d models to csg trees," *ACM Transactions on Graphics*, vol. 37, no. 6, p. 213, 2018.
- [48] T. Du, C. Gan, J. P. Inala, A. Kar, W. Zhang, Q. Fan, Y. Zhao, J. Wu, and W. Matusik, "Neural shape parser for constructive solid geometry," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 755–13 765.
- [49] G. Jacomino, R. Cezaro, A. Sablayrolles, A. Mensch, C. Bamford, C. D. S. Chaplot, G. Lample, L. Saulnier, *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.