CLARIFYING THE CONCEPTS AND TERMINOLOGIES FOR REPRESENTAION OF AI-DRIVEN PARAMETRIC CAD DATASETS

Anonymous authors
Paper under double-blind review

ABSTRACT

A CAD model can be encoded either as an explicit geometric representation, such as a point cloud, a mesh, or a boundary representation (B-Rep), or as an implicit procedural representation of its modeling procedure. Explicit geometric representations primarily support tasks such as 3D visual recognition, segmentation, classification, visualization, rendering, analysis, and machining. However, they capture the geometric shape and visual characteristics of a CAD model rather than its design procedure. Implicit representations may encode either simple Boolean operations procedure (solid modeling, simple sketch-extrude) or feature-based modeling procedure (parametric feature-based modeling, which involves sketches, constraints, feature trees, topological naming, and selection mechanisms). Because the "solid modeling" lacks the core component for parametric modeling, editing and reuse of a CAD model still demand substantial manual effort in industrial scenarios, limiting its ability to support rapid product development in globally competitive markets. Consequently, industrial-grade CAD paradigm has progressively evolved from "solid modeling" to "parametric feature-based modeling." The latter has become the dominant CAD paradigm in contemporary industrial CAD applications, including CATIA, Inventor, Onshape, SolidWorks, Creo, NX and so on. Dataset is the foundation of the AI-driven CAD. Based on two milestone books in the development history of CAD area, this paper clarifies the concepts and terminologies, which are conflated between DeepCAD dataset (the first publicly available dataset for solid modeling) and WHUCAD dataset (the first publicly available dataset for parametric feature-based modeling). Our clarification aligns with the theory researches of CAD area as well as the real-world industrial CAD applications.

1 INTRODUCTION

Two milestone books provide the conceptual foundations for understanding the evolution of solid modeling (Mäntylä, 1988) and parametric feature-based modeling (Shah & Mäntylä, 1995). The former systematically addresses the representation and manipulation of simple Boolean operations procedure, whereas the latter develops the concepts and techniques for organizing models through parametric features and their relationships. These two works establish an important distinction between CAD paradigms, the "Solid modeling" and "Parametric feature-based modeling". This conceptual distinction is essential for understanding how CAD models are represented in datasets.

- One CAD model can be encoded as an explicit geometric representation, such as a point cloud, a mesh, or a boundary representation (B-Rep)(Koch et al., 2019; Rukhovich et al., 2025; Xu et al., 2024b), or as an implicit procedural representation of its modeling procedure (Mäntylä, 1988; Shah & Mäntylä, 1995).
- Explicit geometric representations support tasks such as recognition, segmentation, classification, visualization, rendering, analysis, and machining. They capture the geometric shape and visual characteristics of a CAD model rather than its design procedure.
- Implicit representations may encode as either Boolean operations procedure (solid modeling, such as simple "sketch-extrude") (Mäntylä, 1988; Willis et al., 2021; Xu et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2022; Wan et al., 2026; Xu et al., 2024a) or feature-based modeling procedure (parametric feature-based modeling, which involves sketches, constraints, feature trees, topological naming, and selection mechanisms) (Shah & Mäntylä, 1995; Fan et al., 2025b;a).

Within procedural representations, retaining a construction procedure does not necessarily mean preserving the information required for parametric feature-based modeling. A solid-modeling procedure may consist of geometric construction operations, such as sketching, extrusion, and Boolean combinations, with arguments specifying dimensions and placements (Wu et al., 2021; Khan et al., 2024). Parametric feature-based modeling additionally organizes the design through sketches, constraints, feature trees, topological naming, and selection mechanisms (Shah & Mäntylä, 1995; Fan et al., 2025b). These components specify not only the operations that produce geometry, but also the dependencies and references that connect features and govern their modification (Seff et al., 2022; Kripac, 1997). The distinction therefore concerns more than whether a representation contains parameters or an ordered sequence of operations: it concerns whether the representation also preserves the relationships required for parametric feature-based modeling.

These relationships are fundamental to parametric editing and model reuse. When a representation records construction operations but omits the constraints and references connecting them, modifying a parameter may require manual adjustments to related features. Parametric feature-based modeling addresses this limitation by organizing models around parameters, features, and their relationships, enabling changes to propagate according to the encoded design intent (Shah & Mäntylä, 1995; Seff et al., 2022; Casey et al., 2025). The need to support such editing and reuse has motivated the evolution of industrial CAD paradigm from solid modeling toward parametric feature-based modeling. This paradigm is central to contemporary industrial CAD applications, including CATIA, Inventor, Onshape, SolidWorks, Creo, and NX.

Dataset is the foundation of AI-driven CAD, making these representational distinctions important for describing what information is available for learning (Koch et al., 2019; Seff et al., 2020; Willis et al., 2021). DeepCAD records ordered sketch-and-extrude operations, and Boolean combinations, providing a procedural representation of a CAD model (Wu et al., 2021). WHUCAD represents parametric feature-based modeling through advanced engineering features and selection and naming mechanisms that connect feature operations to referenced topological entities (Fan et al., 2025b). The distinction between these datasets is therefore not limited to the geometric complexity of CAD models. It also concerns how features, dependencies, and references are represented.

Based on the two milestone books in the development history of CAD area Shah & Mäntylä (1995); Mäntylä (1988), this paper clarifies the concepts and terminologies used to describe AI-driven parametric CAD datasets. We distinguish explicit geometric representations from implicit procedural representations and, within the latter, distinguish solid-modeling procedures from parametric feature-based modeling procedure.

We examine DeepCAD in terms of "simple Boolean operations" and WHUCAD in terms of "parametric feature-based operations" to identify where shared terminology conceals differences in representational scope. Our aim is to establish a professional vocabulary that clarifies the CAD dataset descriptions, providing a clearer foundation for dataset construction and for interpreting claims about parametric capabilities from the view of modern industry CAD applications.

2 CAD Data Representations

CAD data representations differ in the geometric and procedural information they preserve. **Explicit geometric representations** describe the evaluated geometry of a CAD model, whereas **implicit procedural representations** specify the modeling procedure through which that geometry is constructed (Mäntylä, 1988; Wu et al., 2021; Fan et al., 2025b). These representations may coexist within a CAD system or dataset (Willis et al., 2021; Xu et al., 2024a; Man et al., 2025; Kolodiazhnyi et al., 2026; Ataei et al., 2026). Their distinction concerns the information retained, rather than mutually exclusive categories of CAD models. Figure 1 illustrates these representations using a common geometric example and depicts a conversion path from modeling procedure to explicit geometric representations.

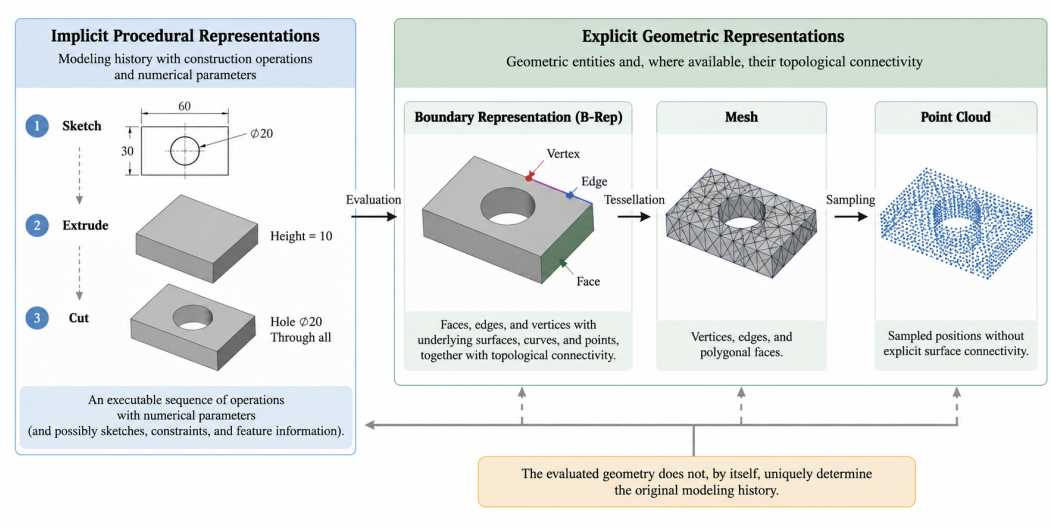


Figure 1: **CAD data representations: geometry and modeling procedure.** An implicit procedural representation records the modeling operations used to construct a CAD model, whereas explicit geometric representations describe its evaluated geometry. The illustrated conversion path comprises the evaluation of a modeling procedure into a boundary representation (B-Rep), the tessellation of the B-Rep into a mesh, and the sampling of the mesh to obtain a point cloud. These representations retain different kinds of information; the evaluated geometry does not, by itself, uniquely determine the original modeling procedure (Wu et al., 2021; Koch et al., 2019).

2.1 Explicit Geometric Representations

Explicit geometric representations describe geometric shape through geometric entities and, where available, their topological connectivity. A **point cloud** consists of sampled positions without explicitly specifying surface connectivity. A **mesh** additionally represents connectivity through vertices, edges, and polygonal faces (Koch et al., 2019). A **boundary representation (B-Rep)** describes the boundary of a solid through faces, edges, and vertices, together with their underlying surfaces, curves, and points. Its geometric information defines the shapes and positions of these entities, while its topological information specifies their connectivity (Mäntylä, 1988; Koch et al., 2019). In contrast to a polygon mesh, a B-Rep can preserve analytic or spline-based curves and surfaces rather than only their polygonal approximations (Xu et al., 2024b).

These representations provide geometric information for 3D visual recognition, segmentation, classification, visualization, rendering, analysis, and machining. Their suitability depends on the geometric accuracy and topological information required by the application (Mäntylä, 1988; Koch et al., 2019). For instance, the curves and surfaces of a B-Rep can be sampled or discretized to obtain point clouds and meshes at different resolutions (Koch et al., 2019; Kolodiazhnyi et al., 2026). Explicit geometric representations therefore support both visual descriptions and computations on geometric shape, without necessarily retaining the original modeling procedure.

The preservation of geometric information must nevertheless be distinguished from the preservation of the original design procedure. In a B-Rep, boundary topology specifies how geometric entities are connected; it does not, by itself, encode the sequence of modeling operations or the feature relationships that produced those entities (Fan et al., 2025b). Similarly, the parametric definitions of curves and surfaces describe boundary geometry, but do not by themselves encode the sketches, constraints, and feature dependencies of a parametric feature-based modeling procedure (Shah & Mäntylä, 1995; Xu et al., 2024b). Consequently, the presence of parametric curves or surfaces should not be interpreted as evidence that the original parametric feature-based modeling procedure has been preserved. This distinction concerns the information represented, not whether the resulting geometry can be manipulated (Liu et al., 2026).

2.2 Implicit Procedural Representations

An implicit procedural representation encodes a CAD model through an executable modeling procedure, including construction operations, and any retained modeling relationships. Such a procedure may specify sketch construction, extrusion, and Boolean combinations, defining dimensions and placements (Wu et al., 2021; Willis et al., 2021; Rukhovich et al., 2025). The term "implicit procedural representation" is used here to denote geometry specified through an executable modeling procedure, rather than through an implicit scalar field. The modeling procedure is recorded in the representation, while the resulting geometry is obtained through its evaluation.

Evaluating a valid modeling procedure in a compatible CAD system can produce a B-Rep, which can subsequently be converted into a mesh or a point cloud (Wu et al., 2021; Koch et al., 2019; Rukhovich et al., 2025). Figure 1 depicts one such conversion path: the modeling procedure is evaluated to obtain a B-Rep, the B-Rep is tessellated into a mesh, and the mesh is sampled to obtain a point cloud. This path illustrates the relationships between the representations rather than a mandatory conversion sequence. However, the resulting geometry does not generally identify the original modeling procedure. Different sequences of operations can produce the same B-Rep; therefore, geometric equivalence does not establish equivalence of the underlying design procedures (Wu et al., 2021). A representation containing only the evaluated geometry consequently preserves different information from one that also records the modeling procedure.

Implicit procedural representations further differ in the scope of the modeling information they retain (Ganin et al., 2021). A procedure used for **solid modeling** may record geometric construction operations. **Parametric feature-based modeling** additionally organizes the model through sketches, constraints, feature trees, topological naming, and selection mechanisms that establish relationships between features and their referenced entities (Shah & Mäntylä, 1995; Fan et al., 2025b). Accordingly, the presence of executable operations does not, by itself, establish that the relationships required for parametric feature-based modeling have been preserved. The distinction between explicit geometric representations and implicit procedural representations must therefore be complemented by an examination of the modeling information encoded within the latter. Section 3 develops this distinction between solid modeling and parametric feature-based modeling.

3 CAD Modeling Paradigms and Terminological Distinctions

The distinction between explicit geometric representations and implicit procedural representations does not, by itself, establish which modeling paradigm a dataset supports. Within implicit procedural representations, the relevant distinction concerns how construction operations and feature relationships are organized. Drawing on the conceptual foundations of solid modeling and parametric feature-based modeling, this section distinguishes these components and their roles in model construction and modification (Mäntylä, 1988; Shah & Mäntylä, 1995).

3.1 Solid Modeling: Geometric Operations

Solid modeling concerns the construction, representation, and manipulation of three-dimensional solid geometry (Mäntylä, 1988). A procedure of solid-modeling may construct solids from geometric primitives or extruded sketch profiles and combine them through Boolean operations, such as union, difference, and intersection. Parameters specify the dimensions, orientations, and placements involved in these operations (Xu et al., 2022; Zhang et al., 2025). DeepCAD provides an example of this representation: its construction procedure records sketch geometry, sketch-plane placements, extrusion distances, and Boolean operation types (Wu et al., 2021).

The presence of parameters permits changes to the geometry produced by a construction procedure (Yuan et al., 2025). However, a value of parameter specifies an operation's current configuration, not necessarily a relationship that must be maintained when another operation changes. For example, a sketch-plane coordinate may coincide with the height of an existing face without encoding a dependency on that face. Sequential Boolean evaluation can still transmit geometric changes to subsequent results; this interaction should not be confused with an explicitly retained support relationship (Wu et al., 2021).

Accordingly, **parameterization** and **parametric feature-based modeling** should not be treated as interchangeable terms. The former identifies adjustable quantities, whereas the latter additionally organizes modeling information through features and their relationships. This distinction does not imply that solid modeling and parametric feature-based modeling are incompatible: feature operations can be implemented using solid-modeling operations. Rather, the distinction concerns whether a modeling procedure records geometric construction alone or additionally preserves the relationships governing feature modification (Shah & Mäntylä, 1995).

3.2 Parametric Feature-Based Modeling: Sketches, Constraints, and Feature Trees

Parametric feature-based modeling organizes a CAD model through parameters, features, and their relationships (Shah & Mäntylä, 1995). A feature represents a modeling operation or design element with a defined role, rather than merely an individual boundary entity. For example, a hole feature is distinct from the faces and edges resulting from its evaluation. A feature-based representation retains information about how such elements are constructed and related, in addition to their resulting geometry.

At the sketch level, **sketch constraints** specify relationships among geometric entities, including coincidence, parallelism, perpendicularity, and dimensional requirements (Ganin et al., 2021; Lin et al., 2026). These relationships differ from the coordinates of the entities at a particular configuration. A sketch whose lines happen to be perpendicular is not equivalent, as a representation, to one that explicitly records a perpendicularity constraint. The latter makes the relationship available for constraint-based modification, rather than requiring it to be inferred from the current geometry (Seff et al., 2020; 2022; Karadeniz et al., 2024).

At the model level, a **feature tree** organizes the modeling procedure, while references and dependencies connect features to their supporting sketches, datum geometry, and previously constructed entities (Shah & Mäntylä, 1995). For instance, a sketch can depend on a support face generated by an earlier feature. A sequence of feature names alone does not establish that this dependency has been retained. Similarly, preserving constraints within an individual sketch does not, by itself, specify the relationships between that sketch and subsequent three-dimensional features. The information retained at the sketch and model levels must therefore be distinguished when characterizing a dataset.

3.3 Topological Naming and Selection Mechanisms

Selection mechanisms identify the entities used as operands or references by a feature operation (Onshape, n.d.). Examples include the edges processed by a fillet and the face supporting a sketch. A fillet radius alone does not specify the selected edges (Fan et al., 2025b). Selection information is therefore part of the modeling procedure.

Topological naming provides a means of identifying and retrieving topological entities within a modeling procedure. Its role must be distinguished from that of selection: selection specifies which entity an operation uses, whereas naming supports the resolution of the corresponding reference. The further requirement of **persistent naming** concerns maintaining identification when upstream modifications change the resulting topology. Faces and edges may split, merge, or disappear, so an identifier assigned to an initial boundary entity does not alone guarantee that the intended reference can be resolved after modification (Kripac, 1997; Jing et al., 2007; 2009; Wu et al., 2018; Liu et al., 2025).

These mechanisms also differ from **association**, which expresses a dependency to be re-evaluated during model modification. Preserving an association requires both a retained relationship and a regeneration procedure that resolves its references and updates the dependent feature. Open CASCADE's naming documentation explicitly distinguishes storing a selection from resolving it after the associated shape has changed (Open CASCADE, n.d.). Consequently, selection, topological naming, and association are related components of parametric feature-based modeling, but none should be used as a synonym for the others. Their presence should not be interpreted as a guarantee of successful regeneration under arbitrary topology changes.

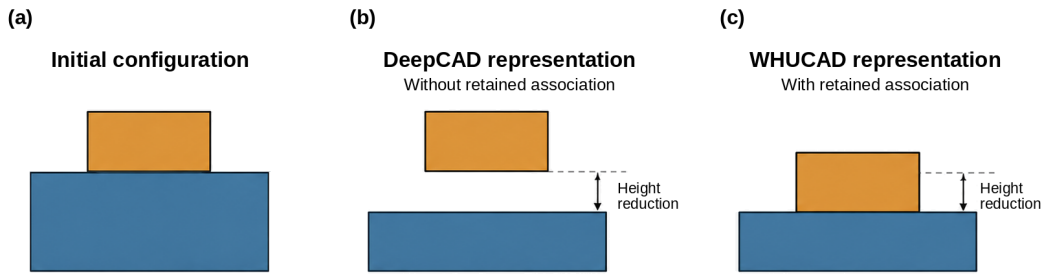


Figure 2: **Parameter changes in DeepCAD and WHUCAD representations.** (a) Initial configuration under a shared attachment requirement. (b) Lowering the base leaves the positioned boss detached. (c) Retaining the support association allows the boss to follow the modified top face. The base bottom and boss height remain unchanged. Schematically redrawn from the example in Fan et al. (2025b), Figs. 8–9.

Table 1: **Terminological distinctions for CAD data representations.** These distinctions summarize the preceding discussion; the presence of one component does not establish that the others are preserved.

Term	Represented information	Distinction
Parameters	Values controlling dimensions, placements, and operations.	Values alone do not specify feature dependencies.
Modeling procedure	Operations used to construct a model.	Operation order alone does not establish associative relationships.
Sketch constraints	Geometric and dimensional relationships within a sketch.	Sketch-level relationships do not describe all model-level dependencies.
Selection mechanisms	Entities used as operands or references.	Selection must be distinguished from reference persistence.
Topological naming	Identification and resolution of referenced topological entities.	Initial identification does not guarantee persistence under all edits.
Associative editing	Modification through encoded dependencies and references.	Preservation concerns the relationships actually encoded.

3.4 Parameter Changes, Associative Editing, and Design Intent

Associative editing propagates parameter changes through encoded feature relationships, rather than modifying values in isolation. In this context, **design intent** refers to the relationships expressed through the model's constraints, dependencies, and references (Shah & Mäntylä, 1995; Casey et al., 2025). This usage concerns the intent represented in the modeling procedure, not every functional or manufacturing requirement that may have motivated the design.

Figure 2 schematically adapts the example in Fan et al. (2025b). The boss is intended to remain attached to the base's top face. Reducing the base height leaves the positioned boss detached in the illustrated DeepCAD construction. The corresponding WHUCAD construction retains the support association and moves the boss with the face. The schematic does not report a new experiment.

The gap violates the attachment requirement specified for this example; geometric separation is not intrinsically an editing error. A different design that deliberately fixes the boss to an independent datum could require a different response. The relevant distinction is therefore whether modification follows the encoded relationship, not whether the resulting bodies always remain in contact. Restoring contact through a separate manual adjustment would likewise not demonstrate that the original representation retained the association.

The example illustrates why adjustable parameters do not establish the preservation of design intent. Both constructions contain modeling operations and parameters, but they differ in the support relationship retained for subsequent modification. Table 1 summarizes the terminology used in this paper. Section 4 applies these distinctions to the information retained in DeepCAD and WHUCAD.

4 REPRESENTATIONAL ANALYSIS OF DEEPCAD AND WHUCAD

The previous sections distinguish CAD data representations from the modeling paradigms they express. This section applies those distinctions to DeepCAD and WHUCAD, examining the construction operations, parameters, and modeling relationships retained in their representations. The objective is to clarify their representational scope and the terminology appropriate to that scope, rather than to compare the algorithms trained on the datasets.

4.1 COMPARISON SCOPE, DIMENSIONS, AND EVIDENCE SOURCES

The comparison concerns the representations documented in the original DeepCAD and WHUCAD publications as well as the Github documents [1]. The principal evidence comprises their command vocabularies, parameter definitions, and descriptions of feature references: Section 3.1 and Table 1 of Wu et al. (2021), and Sections 3.2–3.4 and Tables 1–2 of Fan et al. (2025b). We distinguish information explicitly encoded in these representations from relationships that might be inferred from geometry or introduced during subsequent procedure.

The analysis follows the distinctions established in Section 3: parameters, sketch geometry and constraints, feature organization, selection mechanisms, topological naming, and associative editing. These dimensions are considered separately. An ordered construction procedure does not establish that all feature dependencies are retained, just as a geometric configuration satisfying a constraint does not establish that the constraint itself is encoded.

4.2 MODELING INFORMATION RETAINED IN DEEPCAD

DeepCAD encodes a CAD model through ordered sketch-and-extrude commands. Its sketch profiles comprise lines, arcs, and circles organized into loops. Extrusion commands specify sketch-plane orientation and placement, profile scale, extrusion distances, and operation types, including Boolean combinations with previously constructed geometry (Wu et al., 2021). DeepCAD consequently provides an **implicit procedural representation**: its data retain a construction procedure and adjustable parameters, rather than only an explicit geometric representation of the resulting shape.

The documented vocabulary does not include separate commands for geometric or dimensional sketch constraints, selected topological entities, or advanced engineering features such as fillets and chamfers (Wu et al., 2021). This distinction concerns the construction information retained, not merely the appearance of the resulting geometry. For example, creating a circular opening through a cutting extrusion does not imply that the procedure contains a distinct hole feature. Similarly, coordinates that happen to satisfy a geometric relationship should not be interpreted as an explicit representation of that relationship.

At the model level, construction order and Boolean combinations provide interaction between operations. Nevertheless, the placement of a sketch does not encode which previously generated face it should follow. The separation in Figure 2(b) illustrates this missing support association for the specified construction (Wu et al., 2021; Fan et al., 2025b). It does not imply that every upstream modification leaves all subsequent geometry unchanged. Under the terminology adopted here, DeepCAD is therefore characterized as a **parameterized solid modeling procedure**, rather than as a representation that preserves the feature references required for the illustrated associative edit.

4.3 MODELING INFORMATION RETAINED IN WHUCAD

WHUCAD also uses a sequential representation, but its vocabulary includes advanced engineering features and explicit selection operations. Alongside sketch geometry and extrusion, it records operations such as revolve, pocket, groove, shell, fillet, chamfer, and mirror (Fan et al., 2025b). Thus, the distinction from DeepCAD is not sequence form itself, but the feature information and references expressed within the sequence.

A **selection operation** records the selected object's type, the originating entity's type and index, and the selected object's index. These fields identify the operand or reference used by a subsequent

[1] https://github.com/rundiwu/DeepCAD; https://github.com/fazhihe/WHUCAD

Table 2: **Modeling information retained in DeepCAD and WHUCAD.** The comparison concerns the representations documented in the original publications, not all capabilities of their source CAD systems or subsequent extensions. Entries describe representation content rather than algorithmic performance.

Dimension	**DeepCAD** (Wu et al., 2021)	**WHUCAD** (Fan et al., 2025b)
Data representation	Sketch-and-extrude procedure	Feature procedure with selection references
Sketch geometry	Lines, arcs, and circles	Lines, arcs, circles, and splines
Parameters	Sketch geometry, placement, and extrusion parameters	Sketch geometry and feature-specific parameters
Explicit sketch constraints	No separate constraint commands	Not separately specified in the original feature schema[a]
Advanced engineering features	No distinct fillet, chamfer, or shell commands	Distinct fillet, chamfer, shell, and other feature commands
Feature organization	Construction order and Boolean combinations	Feature order and selected-entity relationships
Selection mechanisms	No explicit topological selection command	Selected-object and originating-entity information
Topological naming	No named support-face references	Feature-related names for selected entities
Support association in Fig. 2	Sketch placement without the support reference	Retained support reference for associative updating

[a]This entry identifies a documentation boundary, not a claim about every native model or later extension. HPSketch's explicit sketch constraints are documented separately (Fan et al., 2025a).

feature. For example, a fillet operation can be specified through both its radius and the selected edge, rather than through a radius alone. The accompanying naming mechanism relates selected entities to the features that generate them (Fan et al., 2025b). Feature identity and topological references are consequently retained as parts of the modeling procedure.

These references also support relationships between features. In Figure 2(c), selecting the base's top face establishes the support association used to update the boss. The example illustrates the role of retained references in **parametric feature-based modeling**; it is not a guarantee of successful regeneration under arbitrary topology changes (Fan et al., 2025b; Kripac, 1997). Sketch-level constraints require a separate qualification. Explicit sketch-constraint support is documented in WHUCAD-HPSketch, a separate parametric CAD sketch dataset (Fan et al., 2025a) [2].

4.4 Comparative Summary of Representational Scope and Terminology

Table 2 summarizes the comparison. Both datasets retain implicit procedural representations, but parameters and construction order describe only part of the information relevant to parametric feature-based modeling. The distinction identified here concerns whether feature operations also retain the references governing their operands and dependencies. Accordingly, **implicit procedural representation**, **parameterization**, and **parametric feature-based modeling** should not be used as interchangeable descriptions (Shah & Mäntylä, 1995; Wu et al., 2021; Fan et al., 2025b; 2026).

The comparison also separates **geometric expressiveness** from **preservation of modeling relationships**. A representation may construct a desired shape without retaining the feature organization through which an engineer would modify it (Fan et al., 2026). For this reason, neither geometric complexity nor the number of supported commands alone establishes the scope of parametric editing.

Within this scope, the distinction between DeepCAD's solid-modeling procedure and WHUCAD's parametric feature-based modeling procedure is grounded in the information retained.

[2]https://github.com/fazhihe/HPSketch

4.5 Scope and Limitations

The present analysis is a conceptual and representational comparison grounded in the two milestone books and the documented representations of DeepCAD and WHUCAD (Mäntylä, 1988; Shah & Mäntylä, 1995; Wu et al., 2021; Fan et al., 2025b). It is not an exhaustive classification of CAD datasets and does not establish comparative algorithmic performance. Moreover, identifying information that a representation can express does not establish that every model in a dataset contains that information or that every subsequent system uses it. Representational scope and the coverage of particular modeling components within a dataset must therefore remain distinct.

The analysis of **design intent preservation** is restricted to relationships encoded through constraints, dependencies, and references. The illustrated support association does not establish successful regeneration under arbitrary topology changes: the presence of an initial identifier alone does not guarantee persistent identification of the intended entity after modification (Kripac, 1997; Open CASCADE, n.d.). Nor does the comparison establish completeness with respect to assembly relationships, cross-system reconstruction, or all functional and manufacturing requirements. Its conclusions concern the terminology appropriate to the modeling information retained in a dataset, rather than a general guarantee of industrial applicability.

5 Conclusion

Drawing on the two milestone books on solid modeling and parametric feature-based modeling (Mäntylä, 1988; Shah & Mäntylä, 1995), this paper clarifies the concepts and terminology used to describe AI-driven parametric CAD datasets. We distinguish **explicit geometric representations**, which describe evaluated geometry without by themselves retaining the original design procedure, from **implicit procedural representations**, which encode modeling procedures. Within procedural representations, we further clarify the distinction between **solid modeling** and **parametric feature-based modeling**. The latter organizes a design through sketches, constraints, feature trees, topological naming, and selection mechanisms, rather than relying on construction operations and parameters alone.

The comparison of DeepCAD and WHUCAD relates these conceptual distinctions to the modeling information retained in their representations. DeepCAD records a solid-modeling procedure, whereas WHUCAD additionally records advanced engineering features and selection and naming information that connect feature operations to their referenced entities (Wu et al., 2021; Fan et al., 2025b). The relevant distinction is therefore not merely the presence of adjustable parameters, the number of commands, or the complexity of the resulting geometry, but the preservation of relationships governing feature modification. When the required relationships are omitted, parameter changes may require manual adjustments to dependent features, limiting associative editing and model reuse.

These representational differences clarify the significance of the evolution from solid modeling to parametric feature-based modeling for industrial CAD applications. As datasets provide the foundation for AI-driven CAD, their descriptions should distinguish geometric shape, construction procedure, and retained feature relationships. The resulting terminology provides a consistent basis for characterizing CAD datasets and aligning claims about parametric editing and model reuse with the modeling information they preserve.

In summary, our clarification aligns with the theory researches from two milestone books of CAD area as well as the real-world industrial CAD applications from CATIA, Inventor, Onshape, SolidWorks, Creo, NX and so on.

AI Use Statement

In this work, we used ChatGPT to assist with language polishing, including improving grammar, clarity, and readability. We reviewed all AI-assisted revisions to ensure that they preserved the intended meaning and technical accuracy of the manuscript. We take full responsibility for the final content of this work, including all text revised with the assistance of generative AI.

REFERENCES

Mohammadmehdi Ataei, Farzaneh Askari, Kamal Rahimi Malekshan, and Pradeep Kumar Jayaraman. Zero-to-CAD: Agentic synthesis of interpretable CAD programs at million-scale without real data. OpenReview, 2026. Accepted by Transactions on Machine Learning Research.

Evan Casey, Tianyu Zhang, Shu Ishida, John Roger Thompson, Amir Khasahmadi, Joseph George Lambourne, Pradeep Kumar Jayaraman, and Karl D. D. Willis. Aligning constraint generation with design intent in parametric CAD. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8613–8622, 2025.

Rubin Fan, Fazhi He, Yuxin Liu, and Jing Lin. A history-based parametric CAD sketch dataset with advanced engineering commands. *Computer-Aided Design*, 182:103848, 2025a.

Rubin Fan, Fazhi He, Yuxin Liu, Yupeng Song, Linkun Fan, and Xiaohu Yan. A parametric and feature-based CAD dataset to support human-computer interaction for advanced 3D shape learning. *Integrated Computer-Aided Engineering*, 32(1):73–94, 2025b.

Rubin Fan, Fazhi He, Yuxin Liu, Jing Lin, Ruibo Wan, Xuecheng Zhang, and Qingchen Kong. AI+CAD data representation architecture: From DeepCAD solid modeling to WHUCAD industrial-level parametric feature modeling. arXiv preprint arXiv:2606.16797v3, 2026.

Yaroslav Ganin, Sergey Bartunov, Yujia Li, Ethan Keller, and Stefano Saliceti. Computer-aided design as language. In *Advances in Neural Information Processing Systems*, volume 34, pp. 5885–5897, 2021.

Shuxu Jing, Fazhi He, and Huajun Liu. A survey of persistent naming problem for topological entities. *Journal of Computer-Aided Design & Computer Graphics*, 19(5):545–552, 2007.

Shuxu Jing, Fazhi He, Soonhung Han, Xiantao Cai, and HuaJun Liu. A method for topological entity correspondence in a replicated collaborative CAD system. *Computers in Industry*, 60(7): 467–475, 2009.

Ahmet Serdar Karadeniz, Dimitrios Mallis, Nesryne Mejri, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. DAVINCI: A single-stage architecture for constrained CAD sketch inference. In *Proceedings of the 35th British Machine Vision Conference*. BMVA, 2024.

Mohammad Sadil Khan, Sankalp Sinha, Talha Uddin Sheikh, Didier Stricker, Sk Aziz Ali, and Muhammad Zeshan Afzal. Text2CAD: Generating sequential CAD designs from beginner-to-expert level text prompts. In *Advances in Neural Information Processing Systems*, volume 37, pp. 7552–7579, 2024.

Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. ABC: A big CAD model dataset for geometric deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9601–9611, 2019.

Maksim Kolodiazhnyi, Denis Tarasov, Dmitrii Zhemchuzhnikov, Alexander Nikulin, Ilya Zisman, Anna Vorontsova, Anton Konushin, Vladislav Kurenkov, and Danila Rukhovich. cadrille: Multi-modal CAD reconstruction with reinforcement learning. In *International Conference on Learning Representations*, 2026.

Jiri Kripac. A mechanism for persistently naming topological entities in history-based parametric solid models. *Computer-Aided Design*, 29(2):113–122, 1997.

Jing Lin, Fazhi He, and Rubin Fan. UniSketch: A unified framework for parametric sketch generation and constraint prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 23532–23540, 2026.

Yilin Liu, Niladri Shekhar Dutt, Changjian Li, and Niloy J. Mitra. B-repLer: Language-guided editing of CAD models. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12. ACM, 2026.

Yuxin Liu, Fazhi He, Rubin Fan, and Zhihao Zong. Namingcad: A naming method for unique identification toward cad modeling learning. In *Proceedings of the 7th ACM International Conference on Multimedia in Asia*. Association for Computing Machinery, 2025.

King Yiu Brandon Man, Ghadi Nehme, Md Ferdous Alam, and Faez Ahmed. Videocad: A dataset and model for learning long-horizon 3d cad ui interactions from video. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, 2025.

Martti Mäntylä. *An Introduction to Solid Modeling*. Computer Science Press, Rockville, 1988.

Onshape. FeatureScript standard library documentation: Queries. Online documentation, n.d.

Open CASCADE. Open CASCADE Technology: OCAF user's guide. Online documentation, n.d.

Danila Rukhovich, Elona Dupont, Dimitrios Mallis, Kseniya Cherenkova, Anis Kacem, and Djamila Aouada. CAD-Recode: Reverse engineering CAD code from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9801–9811, 2025.

Ari Seff, Yaniv Ovadia, Wenda Zhou, and Ryan P. Adams. SketchGraphs: A large-scale dataset for modeling relational geometry in computer-aided design. In *ICML Workshop on Object-Oriented Learning*, 2020.

Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P. Adams. Vitruvion: A generative model of parametric CAD sketches. In *International Conference on Learning Representations*, 2022.

Jami J. Shah and Martti Mäntylä. *Parametric and Feature-Based CAD/CAM: Concepts, Techniques, and Applications*. John Wiley & Sons, 1995.

Ruibo Wan, Fazhi He, and Rubin Fan. Is fusion 360 reconstruction dataset really not enough for training when compared with deepcad dataset? *IEEE Computer Graphics and Applications*, 46 (3):27–39, 2026.

Karl D. D. Willis, Yewen Pu, Jieliang Luo, Hang Chu, Tao Du, Joseph G. Lambourne, Armando Solar-Lezama, and Wojciech Matusik. Fusion 360 Gallery: A dataset and environment for programmatic CAD construction from human design sequences. *ACM Transactions on Graphics*, 40 (4):54:1–54:24, 2021.

Rundi Wu, Chang Xiao, and Changxi Zheng. 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.

Yiqi Wu, Fazhi He, Dejun Zhang, and Xiaoxia Li. Service-oriented feature-based data exchange for cloud-based design and manufacturing. *IEEE Transactions on Services Computing*, 11(2): 341–353, 2018.

Jingwei Xu, Chenyu Wang, Zibo Zhao, Wen Liu, Yi Ma, and Shenghua Gao. Cad-mllm: Unifying multimodality-conditioned cad generation with mllm, 2024a.

Xiang Xu, Karl D. D. Willis, Joseph G. Lambourne, Chin-Yi Cheng, Pradeep Kumar Jayaraman, and Yasutaka Furukawa. SkexGen: Autoregressive generation of CAD construction sequences with disentangled codebooks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 24698–24724, 2022.

Xiang Xu, Joseph G. Lambourne, Pradeep Kumar Jayaraman, Zhengqing Wang, Karl D. D. Willis, and Yasutaka Furukawa. BrepGen: A B-rep generative diffusion model with structured latent geometry. *ACM Transactions on Graphics*, 43(4):119:1–119:14, 2024b.

Yu Yuan, Shizhao Sun, Qi Liu, and Jiang Bian. CAD-Editor: A locate-then-infill framework with automated training data synthesis for text-based CAD editing. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73588–73603, 2025.

Zhanwei Zhang, Shizhao Sun, Wenxiao Wang, Deng Cai, and Jiang Bian. FlexCAD: Unified and versatile controllable CAD generation with fine-tuned large language models. In *International Conference on Learning Representations*, 2025.