Proceedings of the Design Society, Volume 5: ICED25
https://doi.org/10.1017/pds.2025.10102

Challenges and opportunities in the integration of generative AI with computer-aided design

Elias Berger[1,2,✉], Maximilian Peter Dammann[1], Jan Mehlstäubl[2], Bernhard Saske[1], Felix Braun[2] and Kristin Paetzold-Byhain[1]

[1] Technical University Dresden, Germany, [2] MAN Truck & Bus SE, Germany

✉ elias.berger@tu-dresden.de

ABSTRACT: Computer-aided design (CAD) has become essential for hardware product development in our industrial age. However, increasing complexity, shorter lead times, and cost pressures present new challenges. While generative AI has gained significant attention and transformed various business functions, its application in engineering design with CAD remains underdeveloped. Our research aims to explore why generative AI has not yet reached its potential in CAD, despite its prominence in other fields, by identifying key challenges through case studies and a literature review. These challenges include small datasets, difficulty representing mixed data types, proprietary file formats, and lack of advanced CAD modeling commands. We propose future developments such as high-quality datasets, a vendor-neutral format, novel neural network architectures, and expanded generative methods.

KEYWORDS: artificial intelligence, computer aided design (CAD), virtual engineering (VE)

1. Introduction

The human design process, which involves expressing inventions in three-dimensional shapes, is a critical element of our industrial society. Today, Computer-Aided Design (CAD) is the most prominent method for conceiving advanced hardware products (Hirz et al., 2011; Sharma et al., 2023). However, the increasing complexity of industrial product development poses new challenges, including skilled labor shortages, increasing market demands for shorter lead times, and economic pressures to reduce operational costs (Aytac & Wu, 2013; Regenwetter et al., 2022). Consequently, there is a growing demand for innovative tools and methods to improve efficiency and productivity in engineering design and product development workflows. (Tan, 2018).

Generative AI (GenAI) has made significant advancements in text and media generation and is revolutionizing various industries and fields, offering new capabilities across different business areas, such as customer support (Brynjolfsson et al., 2023) or software development (Ebert et al., 2023). This evidence justifies the assumption that GenAI for CAD presents a promising opportunity to address the challenges in engineering design (Kretzschmar et al., 2024).

However, GenAI for CAD appears underdeveloped compared to GenAI for text, source code, or visual media generation as exemplified in Figure 1. As a result, it is not yet mature enough to be reliably used in production environments, which is reflected in the absence of widespread commercial GenAI for CAD offerings. Hence, we want to understand the hurdles holding GenAI for CAD back. This paper investigates why implementing GenAI in CAD is challenging not just for our use cases but remains generally relatively underdeveloped despite the need for such a technology. Therefore, our research is guided by two research questions:

1) What are the key technical barriers preventing the successful application of generative AI techniques to CAD modeling?
2) Which specific advances would enable the effective generation of parametric CAD models?

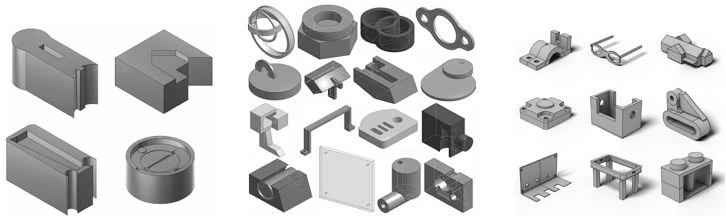

Figure 1. Samples of AI-generated CAD objects from Willis et al., 2021; Wu et al., 2021; Xu et al., 2023 (left to right). While the complexity of the models is increasing with more recent research, more is needed to achieve the human-like performance currently observed in, e.g., text generation with large language models. Composite graphic created by authors using Willis et al., 2021; Wu et al., 2021; Xu et al., 2023

2. Background

2.1. CAD software

In today's engineering companies, CAD is a critical software tool because it enables efficient design, analysis, and optimization. CAD allows for visual inspection of complex 3D models before prototyping, streamlining product development. CAD enhances design engineer productivity and capabilities, making design iteration and simulation easy (Sharma et al., 2023) CAD software tools must provide a constraint solver and a solid modeling kernel. The constraint solver ensures sketch constraints are fulfilled, and the solid modeling kernel finally assembles the three-dimensional (3D) object. These two components rely on a structured sequence of parametric commands and constrained sketches. Several major CAD software vendors have developed proprietary software over decades, while open-source alternatives, like OpenCascade, provide many functions but are considered less feature complete.

2.2. CAD data

In essence, CAD data is a sequence of parametrized commands such as sketch, extrude, chamfer, or fillet. A solid modeling kernel can convert the command sequence into a geometry. In engineering design, CAD is preferred due to these parametric modeling capabilities and construction history features (Vukašinovic & Duhovnik, 2019). Constrained sketches are a crucial feature of CAD data, enabling the enforcement of geometric relations like parallelism and perpendicularity. These constraints provide high editability, allowing models to be easily modified locally while maintaining consistency by propagating changes along constraint relations (Sarkar, 2014). Extrusion is then used to expand the two-dimensional sketch into 3D space. The construction history captures the sequence of operations used to create the model, preserving the entire design process rather than just its outcome.

2.3. GenAI models for computer-aided design

Recent advances in deep learning have enabled learning-based methods to recreate the CAD modeling sequence history, sketch constraints, and extrusion, which can be executed in a CAD tool to generate the final 3D model (Wu et al., 2021; Xu et al., 2022, 2023). Other approaches, like Jayaraman et al., 2023 directly generate B-REP data without relying on command sequences. Generating commands offers advantages like human interpretability and ease of editing, while direct B-REP synthesis may be the easier choice when the construction history is not required.

Current GenAI models for CAD are commonly based on the transformer architecture (Vaswani et al., 2017) that outputs a sequence of tokens corresponding to the construction history's commands and parameters, mimicking the human design process. Changes in machine learning architecture led to incremental improvements across different models, while all GenAI models remain using the same underlying training dataset. Modifications to the original transformer architecture have been made to adapt to a CAD model's mix of data types and hierarchical structure. DeepCAD (Wu et al., 2021), for example, uses separate embeddings and loss functions for CAD commands and their parameters. SkexGen (Xu et al., 2022) uses separate transformer models to encode primitives, loops, and sketches.

3. Research process

This research explores emerging domains where substantive theoretical frameworks are still underdeveloped, aiming to build theory rather than test existing hypotheses by following the procedure in Figure 2.

Following empirical observations of GenAI's performance improvements across other business functions, we investigated the potential applications and efficacy of GenAI integration within CAD workflows for engineering applications. To this end, we conducted a series of workshops at a large automotive firm to capture use cases of GenAI in engineering design tasks that involve CAD software. Our data was collected in four workshops with a total of 15 participants, excluding moderators, from the engineering and IT departments. Workshops were chosen because of their interactive nature, allowing real-time collaboration and ideation.

Every workshop began with a knowledge acquisition phase through a structured presentation from a moderator featuring examples of existing use cases and successful analogical transfers. The main part followed an analogical reasoning method from Kim (2017). Participants were asked to generate use case ideas by transducing existing use cases from other business areas and applying them to GenAI for CAD. The workshop concluded with evaluating the generated use cases and confirming everyone 's understanding of the collected use cases.

In subsequent prototyping of the discovered use cases, we encountered and documented recurring obstacles that hindered our progress. Our findings align with similar challenges in the existing literature, leading us to investigate whether broadly applicable barriers contribute to the underdeveloped state of GenAI in CAD.

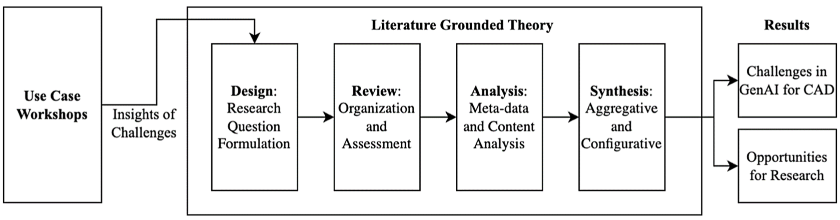

Figure 2. Schematic organization of our research process. Starting from our specific observations, we identified general challenges hindering progress in GenAI for CAD. Graphic created by authors

For our structured investigation, we employed Literature Grounded Theory (LGT) to systematically analyze existing research and identify key themes related to the challenges and opportunities of integrating generative AI in CAD. The LGT method for building theory in research projects is employed here to systematically identify key themes related to the integration of GenAI in CAD and uncover underlying knowledge gaps (Cardoso Ermel et al., 2021; Dunne, 2011).

Our resulting contribution is identifying overarching challenges that generalize beyond our specific use cases and reflect broader issues within the field. We will structure these challenges in this paper and propose future research directions to address the technical limitations and conceptual gaps in applying GenAI to CAD.

4. Use cases of GenAI in CAD

After our workshop series at an automotive company, we synthesized the identified opportunities for GenAI in CAD into eight distinct use cases, as shown in Table 1. Further, we provide detailed descriptions of three high rated use cases: CAD geometry generation (1), B-REP to CAD (2), and generating variants (3).

1) A GenAI model could draft an initial CAD prototype based on textual specifications during the requirements engineering phase. Engineers can provide early feedback on the prototype, allowing for immediate adjustments, while project managers can use the early prototype to facilitate more

informed discussions with stakeholders. Iterative refinement of the generated prototype based on feedback would enable more effective exploration of the design space before detailed design begins.

2) Converting CAD models from B-REP back to parametric models with sketches and design history is a significant unresolved challenge, often done manually. B-REP models are commonly exchanged for compatibility or nondisclosure reasons, losing parametric information. GenAI could automate this complex reverse engineering task. In engineering companies, external suppliers are usually providing B-REP files for exchanging CAD objects. Consequently, engineers often need to manually reverse engineer these geometries using their internally used CAD software, which is a time-consuming process.
3) Engineers often design and simulate multiple component variations in the early development phase to evaluate trade-offs and optimize performance. Generating these design alternatives is time-consuming when done manually. GenAI could automate the creation of design variations based on key parameters, constraints, and performance targets. This would enable rapid design space exploration, streamline iterations, and allow engineers to focus on higher-level design decisions and analysis rather than manually modeling each variant.

Table 1. Distribution of identified GenAI-enabled CAD use cases across four workshops. Each checkmark (✓) indicates that the use case was independently proposed or supported in the respective workshop, demonstrating convergence on critical applications across different participant groups

Name	Workshop 1	Workshop 2	Workshop 3	Workshop 4	Total
2D Sketch to CAD	✓				1
3D Mesh to CAD	✓				1
B-REP to CAD		✓		✓	2
CAD Generation given specifications	✓		✓	✓	3
Generating Variants		✓		✓	2
Fix or optimized CAD Structure		✓		✓	2
Similarity Detection	✓		✓		2
Re-Assembly of existing components	✓				1

5. Challenges in implementing GenAI in CAD

5.1. Training data availability

For deep-learning models, improvement in accuracy metrics is generally correlated with larger training datasets (Kaplan et al., 2020). However, most available 3D datasets for machine learning are limited in size, particularly in construction histories and constrained sketches, which are crucial for CAD modeling. The first and still most prominent dataset resembling CAD is the **ABC** dataset (Koch et al., 2019). This dataset is parsed from data available on PTC OnShape, a free-for-personal-use CAD platform where publicly accessible models are created by users. However, these models are neither filtered nor curated, and the dataset only includes the models in STEP file format, which lacks the constrained sketches and construction history essential for professional CAD modeling.

SketchGraphs (Seff et al., 2020) makes use of OnShape as well. However, in contrast to the ABC dataset, it extracts the constrained sketches but does not extrude them into 3D space and the entire construction history. The result is a dataset of 2D drawings together with dimensional and geometric constraints.

DeepCAD (Wu et al., 2021) extends the ABC dataset by introducing an enhanced extraction script that captures the construction history of the models, though it is limited to only sketch and extrude operations, excluding more complex operations (see Figure 3). This results in 129,624 models available in a custom JSON and STEP format, with complete construction histories but lacking geometric constraints.

Another dataset is the **Fusion360 Gallery** Dataset (Willis, Pu, et al., 2021), which includes around 8,625 models sourced from the Autodesk Community Gallery and published by Autodesk researchers. These models are published in a custom JSON format, including constrained sketches and the entire construction history. Again, it is limited to sketch and extrude command types.
The characteristics of the discussed CAD datasets are compared in Table 2, highlighting substantial differences in construction history availability and primitive shape distributions.

Table 2. Comparison of CAD model datasets: size, construction history, and primitive shape distribution. Table created by authors

	Size(sample count)	Construction History	% Cuboids	% Cylinders
ABC	1M	No	-	-
SketchGraphs	14M	No	-	-
DeepCAD	130T	Yes	28.9%	15.5%
Fusion360	9T	Yes	7.6%	10.0%

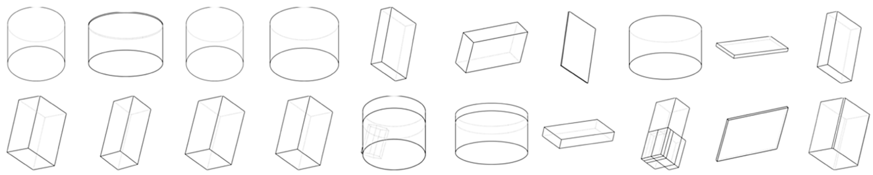

Figure 3. Randomly sampled CAD objects from the DeepCAD dataset by Wu et al., 2021, exemplifying the simple nature of the training data. Graphic created by authors

While 3D printing, computer graphics, and similar interest groups publish CAD data collected in a dataset, this content predominantly does not represent industrial applications. As a result, a significant deficiency exists in professional-grade CAD data. This is reflected in the DeepCAD and Fusion360 Gallery datasets, containing trivial objects such as a single cylinder or cuboid. Compared to CAD models utilized in engineering firms, publicly available datasets often lack the complexity and detail characteristic of CAD models, thus limiting their applicability in professional settings. Another common issue across these datasets is the presence of geometric errors, such as self-intersecting edges, misaligned edges, and duplicate vertices, which complicate their use in training machine learning models. Finally, the CAD models within each dataset are devoid of annotations, such as captions, names, labels, or any additional information that could facilitate training for subsequent tasks such as Text-To-CAD integration.

5.2. Data representation

A key aspect of GenAI for CAD research involves developing techniques for encoding complex parametric data structures into formats suitable for machine learning.
Effective data representation is a critical factor in the success of deep learning models (Goodfellow et al., 2016). Properly structured and comprehensive representations enable the model to capture essential features, improving learning efficiency, generalization, and overall performance in each task.
A significant challenge in CAD modeling stems from its sequential and parametric nature. The order of operations is critical, as each operation must be executed in a specific sequence (Wu et al., 2021). Furthermore, for each operation, the generative AI model must identify the appropriate parameterization to ensure accurate representation and functionality of the design.
The parametrizations of each sequence step are, in turn, challenging as the parameters are a mix of discrete data (for example, clockwise sign), continuous data (for example, extrusion distance), and relational data (for example, parallel constraint between two lines). Relational constraints can exist between various elements, such as edges (e.g., collinear constraints) or points (e.g., coincidence constraints). The added complexity arises from handling constraints between edges and points on those

edges, necessitating advanced strategies for effectively representing and maintaining these relationships within the model.
Current research (Wu et al., 2021; Xu et al., 2023) has simplified the representation of CAD data by omitting constraints. This method represents a model's construction history by encoding CAD commands and their parameters into sequential vectors. Although this method is computationally efficient, it overlooks constraints—an essential feature in CAD.

5.3. Proprietary file formats

Commercial CAD vendors protect their data files through proprietary binary formats that are accessible only via licensed software, keeping innovation in the hands of the commercial vendors (Stroud & Nagy, 2011). The licenses are often costly, limiting accessibility for academic researchers and startups. Additionally, incompatible file formats across different vendors create artificial barriers, hindering data aggregation from diverse sources and generalizing research findings. A neutral format, in contrast, would provide independence from vendor-specific formats, promoting broader interoperability and compatibility. While the STEP and IGES formats aim to improve interoperability, they come at the expense of losing constrained sketches and construction history information.
Existing research has introduced custom JSON formats to represent CAD model information, but their structure remains closely tailored to the specific scope of each project (Willis et al., 2021; Wu et al., 2021). File format of open-source alternatives like FreeCAD is as well closely coupled with the main software and is not designed for interoperability. Defining a comprehensive CAD format is challenging, as it must represent geometric shapes and encode details such as tolerances, materials, and annotations.

5.4. CAD operations

Current generative AI approaches for CAD modelling primarily rely on Sketch and Extrude operations, which represents a significant limitation for practical applications. While these operations form the foundation of CAD modelling, professional designers routinely employ a broader set of operations including Revolution, Sweep, Loft, Blend, Fillet, and Chamfer to create complex geometries (Heidari & Iosifidis, 2024). Operations like Fillets and Chamfers can only be applied to B-REP edges—elements that only emerge after converting the Sketch and Extrude sequence into a B-REP model (Jayaraman et al., 2023). This technical constraint forces designers to manually add these features as post-processing steps, diminishing the efficiency gains promised by generative AI. While the Sketch and Extrude paradigm provides a useful starting point for research, recent work on highlights the necessity of incorporating a wider range of operations to generate realistic engineering components (Li et al., 2023; Zhang et al., 2023).

6. Opportunities for further research

6.1. GenAI architecture for CAD data

A critical challenge in developing generative AI models for computer-aided design is effectively encoding the mix of categorical, continuous, and relational data types. As it is difficult for a single model to comprehend a mix of data types (Borisov et al., 2024), we propose a mix of expert models (see Figure 4) to capture each of the essential features of the CAD object because it may be challenging for just a single model to handle different types of data. However, the implementation of expert model ensembles will likely still require domain-specific fine-tuning or retraining to accommodate the diverse use cases described in Section 4. This comes with the practical constraint of requiring additional computational resources.
Our method assumes that CAD objects are constructed through an alternating sequence of two fundamental operations: sketch and extrusion. Each of these operations is handled by dedicated generators, which take a conditioning vector as input to control the generation process. The conditioning vector (Mirza & Osindero, 2014) can vary depending on the specific use case and could be derived from various sources such as embeddings of feature specifications, images, or any other relevant information. This flexibility allows our method to adapt to different requirements and generate CAD objects based on the provided conditioning vector input. The sketch generator consists of two submodules: a primitive generator and a constraint generator (Para et al., 2021).
The primitive generator is a decoder-only transformer (Radford et al., 2019) that produces lines, arcs, and circles through an autoregressive process to accommodate variable-length sketches. Consistent with previous work (Wu et al., 2021; Xu et al., 2022, 2023), we employ the technique of quantizing

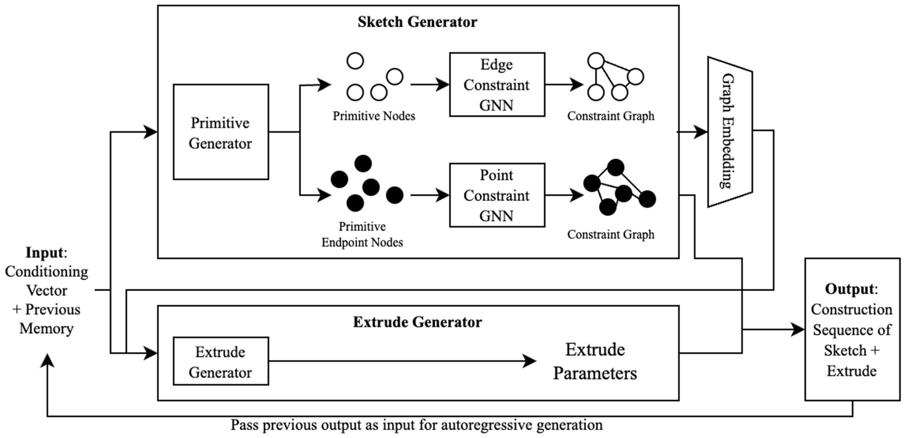

Figure 4. Schematic of our proposed GenAI architecture for CAD with separate sketch and extrude generators for autoregressive generative of CAD command sequences. Graphic created by authors

continuous parameters into discrete values. However, in our approach, we utilize a higher level of numerical precision to achieve more accurate results.

Constraints between edges of a sketch are well represented as a constraint graph (CG), where primitives serve as nodes, and constraints in the sketch relate to annotated edges in the CG (Ding, 2014; Seff et al., 2020). Binary constraints such as *parallel* or *perpendicular* can be directly mapped from and to the CG. For unary constraints such as *vertical* and *horizontal*, we introduce virtual nodes representing coordinate axes, allowing these constraints to be modelled as binary constraints in the CG.

Constraints such as *coincidence* and *concentric* don't operate on primitives but on the endpoints of primitives, making a separate CG necessary to generate coincidence and concentric constraints. Other constraint types exist, but the mentioned constraints already cover most cases as seen in Figure 5 While earlier methods employed pointer networks to model CGs (Seff et al., 2022), our constraint generator is a Graph Neural Network (GNN) to fully leverage the inherent graph structure. For training, we assemble CGs from existing CAD objects. At inference, our GNN acts as a link predictor using primitive nodes for input and outputting a CG. Link prediction using machine learning has been demonstrated in Zhu et al., 2022.

The extrusion generator is also a decoder-only transformer (Radford et al., 2019) and determines appropriate extrusion parameters, for example extrusion distance or taper angle based on sketch geometry, construction history, and conditioning vectors. Again, continuous parameters quantized.

Instead of using end-of-sequence tokens within each generative module, our architecture employs a dedicated stopping classifier. This modular approach separates the generation logic from termination decisions, allowing each component to focus on its specific task while the classifier independently determines when the model has produced a complete CAD object based on the construction history and conditioning vector.

Combining sketches and extrude parameters creates a construction sequence that can be converted into a 3D object by a solid modelling kernel. This approach enables the generation of complex, manufacturable CAD models. To manage model complexity, we propose to pre-train each module separately and then fine-tune the entire ensemble on a smaller data set.

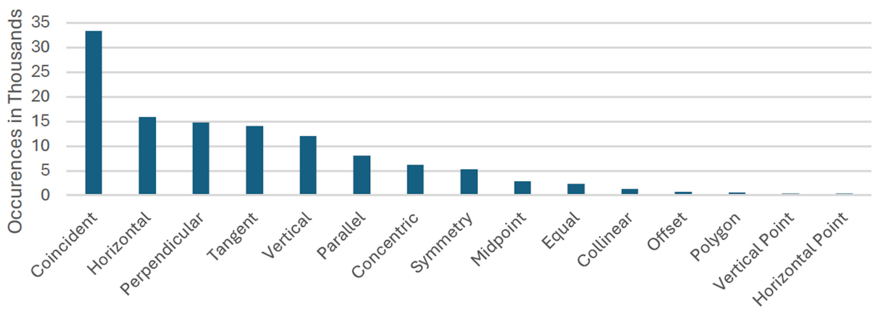

Figure 5. Distribution of constraints in the Fusion360 gallery dataset by Willis et al., 2021. With the mentioned constrains about 93.6% of all constraints are covered by our proposed GenAI model

6.2. Accumulation of a large dataset

Engineering and manufacturing companies create detailed CAD models as technical specifications for production but are reluctant to make these models publicly available. This presents research opportunities to address the lack of accessible industrial CAD data while considering company requirements. Potential solutions include data anonymization through feature removal and geometric transformations, and the generation of synthetic datasets. However, machine-learning models trained on synthetic data typically achieve lower accuracy when validated against real data (Rankin et al., 2020).

By pursuing these research directions, academia can help bridge the gap between industry and research in professional-grade modelling while respecting the proprietary nature of industrial CAD data. Since industry cooperation is crucial for data access, this could involve exploring collaborative opportunities and developing data-sharing mechanisms that benefit both research institutions and companies.

6.3. Expanding CAD command palette

Current GenAI methods for CAD primarily focus on generating solid-body components with the two basic operations sketch and linear extrusion, limiting their applicability to real-world design tasks. Future research should prioritize expanding these methods to support a broader range of complex modeling operations commonly used in industry, such as fillet, chamfer, loft, sweep, and draft (Heidari & Iosifidis, 2024). Sheet metal design, which heavily relies on bending operations, is also notably absent from existing work despite its prevalence in automotive manufacturing (Trzepieciński, 2020).

For sketch primitives current state-of-the-art models such as Wu et al., 2021; Xu et al., 2022, 2023 only support line, arc and circle primitives in both their datasets and GenAI models (Sarkar, 2014). The accumulation of a large dataset as proposed in chapter 6.2. could include new commands besides sketch and extrude, which would be essential for developing more versatile and industry-relevant GenAI CAD systems.

6.4. Vendor-neutral file format

A vendor-neutral CAD file format would foster innovation by eliminating barriers created by proprietary formats and incompatibilities between CAD systems (Lee et al., 2019). A neutral format would enable broader interoperability, making aggregating data from diverse sources easier. Achieving this could be accomplished either through a collaborative standardization process or by the widespread adoption of a single open-source format. Developing a standardized format upon existing JSON formats or YAML could provide significant benefits since both are flexible, human readable and version control compatible make it a promising foundation for CAD data exchange. Its widespread support across programming languages could facilitate broader adoption and collaboration in the CAD community.

7. Discussion

The underdevelopment of GenAI for CAD compared to GenAI for other media stems from several interrelated challenges. Existing CAD datasets suffer from inadequate complexity and sparse annotations, while the intrinsic characteristics of CAD data pose unique machine learning representation challenges, and proprietary file formats in commercial systems create substantial barriers to accessibility and interoperability. In the future we would like to overcome these challenges and investigate the practical implementation of each use case. The presented use cases of GenAI for CAD are also applicable to design in team projects, with the AI potentially acting as an enabler for the entire team. GenAI tools can foster interdisciplinary collaboration, allowing team members from different backgrounds to leverage the AI's capabilities to better understand and manipulate CAD data and effectively work with complex CAD software.

8. Conclusion

Advancing GenAI in CAD requires addressing several fundamental challenges. Key requirements include building comprehensive datasets that better represent industrial applications, developing a vendor-neutral file format that preserves all essential information, creating AI architectures capable of handling diverse data types, and expanding generative methods to encompass complex CAD operations. A significant obstacle remains the disconnect between companies possessing valuable training data but lacking GenAI development resources, and organizations with AI expertise but limited access to real-world CAD data. Future work will focus on implementing the proposed AI architecture using open file formats to bridge these gaps and advance the field of generative AI for CAD.

References

Aytac, B., & Wu, S. (2013). Characterization of demand for short life-cycle technology products. *Annals of Operations Research*, 203, 1–23. https://doi.org/10.1007/s10479-010-0771-5

Borisov, V., Leemann, T., Seßler, K., Haug, J., Pawelczyk, M., & Kasneci, G. (2024). Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(6), 7499–7519. https://doi.org/10.1109/TNNLS.2022.3229161

Brynjolfsson, E., Li, D., & Raymond, L. R. (2023). Generative AI at work (Working Paper No. 31161; Working Paper Series). National Bureau of Economic Research. https://doi.org/10.3386/w31161

Cardoso Ermel, A. P., Lacerda, D. P., Morandi, M. I. W. M., & Gauss, L. (2021). Literature grounded theory (LGT). In *Literature reviews: Modern methods for investigating scientific and technological knowledge* (pp. 85–145). Springer International Publishing. https://doi.org/10.1007/978-3-030-75722-9_6

Ding, B. (2014). 3D CAD Model Representation and Retrieval based on Hierarchical Graph. *Journal of Software*, 9(10), 2499–2506. https://doi.org/10.4304/jsw.9.10.2499-2506

Dunne, C. (2011). The place of the literature review in grounded theory research. *International Journal of Social Research Methodology*, 14(2), 111–124. https://doi.org/10.1080/13645579.2010.494930

Ebert, C., Louridas, P., & Ebert, C. (2023). Generative AI for software practitioners. *IEEE Software*, 40(4), 30–38. https://doi.org/10.1109/MS.2023.3265877

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.

Heidari, N., & Iosifidis, A. (2024). Geometric Deep Learning for Computer-Aided Design: A Survey (No. arXiv:2402.17695). *arXiv*. http://arxiv.org/abs/2402.17695

Hirz, M., Harrich, A., & Rossbacher, P. (2011). Advanced computer aided design methods for integrated virtual product development processes. *Computer-Aided Design and Applications*, 8, 901–913. https://doi.org/10.3722/CADAPS.2011.901-913

Jayaraman, P. K., Lambourne, J. G., Desai, N., Willis, K. D. D., Sanghi, A., & Morris, N. J. W. (2023). SolidGen: An Autoregressive Model for Direct B-rep Synthesis (No. arXiv:2203.13944). *arXiv*. http://arxiv.org/abs/2203.13944

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., & Amodei, D. (2020). Scaling laws for neural language models. https://arxiv.org/abs/2001.08361

Kim, E. (2017). Workshop design for enhancing the appropriateness of idea generation using analogical thinking. *International Journal of Innovation Studies*, 1(2), 134–143. https://doi.org/10.1016/j.ijis.2017.10.002

Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., & Panozzo, D. (2019). ABC: A Big CAD Model Dataset For Geometric Deep Learning (No. arXiv:1812.06216). *arXiv*. http://arxiv.org/abs/1812.06216

Kretzschmar, M., Dammann, M. P., Schwoch, S., Braun, F., Saske, B., & Paetzold-Byhain, K. (2024). Evaluating the role of generative AI in product development and design—A systematic review. In J. Malmqvist, M. Candi, R. J. Saemundsson, F. Bystrom, & O. Isaksson (Eds.), *Proceedings of NordDesign 2024* (pp. 21–30). Technische Universitat Dresden, Dresden, Germany and MAN Truck & Bus SE, Munich, Germany. https://doi.org/10.35199/NORDDESIGN2024.3

Lee, S., Baek, H., & Oh, S. (2019). The role of openness in open collaboration: A focus on open-source software development projects. *ETRI Journal*, 41(6), 801–810. https://doi.org/10.4218/etrij.2018-0536

Li, P., Guo, J., Zhang, X., & Yan, D. (2023). SECAD-Net: Self-Supervised CAD Reconstruction by Learning Sketch-Extrude Operations (No. arXiv:2303.10613). *arXiv*. http://arxiv.org/abs/2303.10613

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *ArXiv*, abs/1411.1784. https://doi.org/10.48550/arXiv.1411.1784

Para, W. R., Bhat, S. F., Guerrero, P., Kelly, T., Mitra, N., Guibas, L., & Wonka, P. (2021). SketchGen: Generating Constrained CAD Sketches.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners.

Rankin, D., Black, M. M., Bond, R., Wallace, J., Mulvenna, M., & Epelde, G. (2020). Reliability of supervised machine learning using synthetic data in health care: Model to preserve privacy for data sharing. *JMIR Medical Informatics*, 8. https://doi.org/10.2196/18910

Regenwetter, L., Nobari, A. H., & Ahmed, F. (2022). Deep Generative Models in Engineering Design: A Review (No. arXiv:2110.10863). *arXiv*. http://arxiv.org/abs/2110.10863

Sarkar, J. (2014). Computer aided design: A conceptual approach (1st ed., p. 739). CRC Press. https://doi.org/10.1201/b17741

Seff, A., Ovadia, Y., Zhou, W., & Adams, R. P. (2020). SketchGraphs: A Large-Scale Dataset for Modeling Relational Geometry in Computer-Aided Design (No. arXiv:2007.08506). *arXiv*. http://arxiv.org/abs/2007.08506

Seff, A., Zhou, W., Richardson, N., & Adams, R. P. (2022). Vitruvion: A Generative Model of Parametric CAD Sketches (No. arXiv:2109.14124). *arXiv*. http://arxiv.org/abs/2109.14124

Sharma, V., Sharma, V., & Shukla, O. J. (2023). Principles and practices of CAD/CAM (1st ed., p. 332). Chapman and Hall/CRC. https://doi.org/10.1201/9781003350842

Stroud, I., & Nagy, H. (2011). Solid Modelling and CAD Systems. Springer London. https://doi.org/10.1007/978-0-85729-259-9

Tan, J. (2018). Special issue on innovative design of complex products. *Chinese Journal of Mechanical Engineering*, 31. https://doi.org/10.1186/s10033-018-0232-7

Trzepieciński, T. (2020). Recent Developments and Trends in Sheet Metal Forming. *Metals*, 10(6), 779. https://doi.org/10.3390/met10060779

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, \Lukasz, & Polosukhin, I. (2017). Attention Is All You Need. *Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS)*, 5998–6008.

Vukašinovic, N., & Duhovnik, J. (2019). Advanced CAD modeling: Explicit, parametric, free-form CAD and re-engineering. Springer International Publishing. https://doi.org/10.1007/978-3-030-02399-7

Willis, K. D. D., Jayaraman, P. K., Lambourne, J. G., Chu, H., & Pu, Y. (2021). Engineering Sketch Generation for Computer-Aided Design. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2105–2114. https://doi.org/10.1109/CVPRW53098.2021.00239

Willis, K. D. D., Pu, Y., Luo, J., Chu, H., Du, T., Lambourne, J. G., Solar-Lezama, A., & Matusik, W. (2021). Fusion 360 Gallery: A Dataset and Environment for Programmatic CAD Construction from Human Design Sequences (No. arXiv:2010.02392). *arXiv*. http://arxiv.org/abs/2010.02392

Wu, R., Xiao, C., & Zheng, C. (2021). DeepCAD: A Deep Generative Network for Computer-Aided Design Models. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6752–6762. https://doi.org/10.1109/ICCV48922.2021.00670

Xu, X., Jayaraman, P. K., Lambourne, J. G., Willis, K. D. D., & Furukawa, Y. (2023). Hierarchical Neural Coding for Controllable CAD Model Generation (No. arXiv:2307.00149). *arXiv*. http://arxiv.org/abs/2307.00149

Xu, X., Willis, K. D. D., Lambourne, J. G., Cheng, C.-Y., Jayaraman, P. K., & Furukawa, Y. (2022). SkexGen: Autoregressive Generation of CAD Construction Sequences with Disentangled Codebooks (No. arXiv:2207.04632). *arXiv*. http://arxiv.org/abs/2207.04632

Zhang, S., Guan, Z., Jiang, H., Ning, T., Wang, X., & Tan, P. (2023). Brep2Seq: A dataset and hierarchical deep learning network for reconstruction and generation of computer-aided design models. *Journal of Computational Design and Engineering*, 11(1), 110–134. https://doi.org/10.1093/jcde/qwae005

Zhu, Z., Zhang, Z., Xhonneux, L.-P., & Tang, J. (2022). Neural Bellman-Ford Networks: A General Graph Neural Network Framework for Link Prediction (No. arXiv:2106.06935). *arXiv*. https://doi.org/10.48550/arXiv.2106.06935