Quality of CAD Generation with Large Language Models: A Critical Overview

Kevin Herrmann[1,*], Felix Pusch[1], Elias Berger[2,3], Jan Mehlstäubl[3], Kristin Paetzold-Byhain[2], Roland Lachmayer[1]

[1] Institute of Product Development, Leibniz University Hanover, Germany
[2] Chair of Virtual Product Development, Dresden University of Technology, Germany
[3] MAN Truck & Bus SE, Germany

** Korrespondierender Autor:*
Kevin Herrmann
An der Universität 1
30823 Garbsen
Germany
☎ *+49 (0)511 762 2552*
🖂 *herrmann@ipeg.uni-hannover.de*

Abstract

Large language models increasingly generate CAD models from text, images, or sketches, but prior evaluations emphasize geometric similarity while overlooking design-relevant properties like parametricity, feature structure, and process integration. We conduct a systematic review of LLM-based CAD generation across three academic databases using LLM-assisted screening and propose a six-level maturity model grounded in engineering design theory (VDI 2209) to assess design reusability. Results show most approaches cluster at maturity level 3 (parametric CAD scripts), with none reaching feature-based model history or assembly generation. We discuss training data, modality, and fine-tuning strategies needed to reach higher maturity and enable industrial adoption.

Keywords

digital engineering, generative design, computer-aided design, generative artificial intelligence, large language models

© 2026 die Autoren | DOI: 10.35199/dfx2026.8

1. Motivation

Large language models (LLMs) are increasingly being applied to generate CAD models from textual descriptions, images, sketches, and other structured inputs [1]. From an industrial perspective, however, geometric similarity to a reference model is of limited relevance. The critical criterion is whether generated models are editable, verifiable, and reusable throughout downstream engineering processes [2]. Existing evaluation approaches largely neglect these aspects. Current studies predominantly assess performance using geometric metrics, whereas engineering-related characteristics such as parametric behavior, feature structure, model history, and process integration receive comparatively little attention [3]. Consequently, it remains unclear to what extent generated models can support subsequent product development activities or how far current approaches are from producing industrially deployable CAD models.

A further limitation concerns methodological transparency. Existing approaches differ substantially with respect to training data, input modalities, output representations, and training strategies. However, there is little systematic evidence regarding which combinations of these factors contribute most strongly to model quality [3]. As long as these relationships remain insufficiently understood, neither the current state of the art nor the most promising directions for future research can be reliably identified. In particular, it remains unclear which developments are required to achieve the level of reusability necessary for industrial adoption.

To address this gap, this study conducts a systematic analysis of contemporary LLM-based CAD generation approaches and introduces a maturity model for assessing engineering reusability based on VDI 2209 [4]. The study is guided by the following research questions.

- **RQ1**: Which maturity levels are suitable for evaluating geometry models generated by large language models?
- **RQ2**: Which combinations of training data, input modalities, output representations, and training methods are associated with the highest model quality in current LLM-based CAD generation approaches?
- **RQ3**: Which research activities are required to achieve higher maturity levels in LLM-based CAD generation?

2. Background

Generative design comprises computational methods that derive design alternatives algorithmically from a problem specification rather than through direct manual modeling [5, 6]. Instead of defining geometry explicitly, designers specify the conditions that characterize an admissible solution [7].

2.1. Rule-based Methods for Generative Design

Topology optimization discretizes the design domain and treats material distribution as the design variable [8]. The admissible solution space is constrained by requirements such as mass and volume limits, stress and deformation thresholds, and manufacturing constraints [9]. Level-set methods represent component boundaries implicitly as level sets of scalar fields [10], enabling boundary evolution while preserving geometric sharpness, although the formulation remains closely coupled to the underlying physical model [11]. Parametric and knowledge-based approaches define the solution space through parameters, domains, and constraints [6]. Engineering knowledge, standards, and manufacturing requirements are formalized as explicit rules [12]. Constraint solvers and evolutionary algorithms explore the resulting design space without modifying its predefined boundaries [6]. Grammar-based approaches generate alternatives through recursive rule application to a predefined vocabulary, thereby defining the solution space as the set of all derivable expressions [13]. These approaches share a common trait: domains, constraints, and objectives are represented explicitly, offering transparency but at the cost of modeling effort that grows with dependency count, while larger problems expand the solution space and make search the main bottleneck [14]. Since such models are application-specific, transferring them to new use cases typically requires extensive remodeling [6].

2.2. Data-based methods for Generative Design

Recent advances in LLMs have established an alternative paradigm for CAD generation that does not rely on explicitly formalized design rules. Their generative capabilities originate from large-scale pretraining on text and source-code corpora and adaption to unseen tasks [15]. Simple methods use zero-shot settings and may be combined with iterative feedback mechanisms to repair invalid outputs [16–19]. Prompt-based approaches guide generation through task descriptions, examples, or retrieved contextual information without modifying model parameters [17]. Adaption to CAD modeling is achieved through fine-tuning on domain-specific datasets [20]. Natural language represents the dominant input modality [21]. Images and sketches are increasingly incorporated, either independently or within multimodal architectures [18, 22]. Generated outputs typically fall into three categories. First, LLMs may control external CAD systems through application programming interfaces [20, 23]. Second, they may generate explicit modeling sequences consisting of operations such as sketching and extrusion [18, 24]. Third, they may produce executable CAD code [25].

These characteristics provide several advantages. The absence of an explicit rule base reduces modeling effort and lowers adoption barriers. Pretraining enables a single model to leverage broad knowledge related to design, manufacturing, standards, and programming across multiple engineering domains [20, 26]. Natural-language interaction eliminates the need for specialized modeling languages [21] and supports iterative refinement through conversational feedback. Furthermore, symbolic outputs remain interpretable and machine-verifiable [17, 18]. Such capabilities are particularly valuable during early design stages, where rapid generation of alternative concepts can support conceptual design [7, 27].

LLM-based CAD generation differs, because the design space is implicitly defined by the LLM parameters. Consequently, admissible design regions and the rationale underlying generated solutions are generally inaccessible and cannot be constrained explicitly [20]. Practical limitations include hallucinated modeling operations, geometrically inconsistent outputs, and limited reproducibility under identical input conditions [17, 18, 28]. In addition, dimensional and functional accuracy often remain insufficient for industrial product development because existing training data are dominated by simple geometric primitives and rarely capture design intent or manufacturing knowledge explicitly [18]. These limitations are not adequately reflected by commonly used geometric evaluation metrics such as Chamfer Distance, Intersection-over-Union, and Normal Consistency [20]. Accordingly, the present study introduces an evaluation framework that focuses on engineering reusability as the primary criterion for assessing the maturity of LLM-generated CAD models.

3. Definition of quality metrics for CAD-Models

A maturity model describes a domain through an ordered sequence of levels representing progression from an initial to a target state [29]. Each level is defined by observable characteristics and presupposes the capabilities of preceding levels, establishing an ordinal and cumulative scale. Maturity models enable the classification of a given state, the comparison of different objects on a common scale, and the identification of measures required to reach the next level. This structure is suitable for assessing LLM-generated CAD models for two reasons. First, commonly used geometric similarity metrics evaluate only correspondence with a reference geometry and neglect engineering-relevant characteristics [3, 20]. Second, engineering reusability is inherently hierarchical because it depends on the underlying representation and increases with the information contained in the model [2]. VDI 2209 provides the foundation for the proposed maturity model because it systematically distinguishes representation forms in three-dimensional product modeling and relates them to their applicability in downstream engineering processes [4].

Wireframe and surface models provide incomplete geometric descriptions and do not distinguish unambiguously between interior and exterior regions. Voxel models consist of independent volumetric elements and are primarily used for digital mock-ups [4]. These representations support geometric inspection and collision detection but not the derivation of physical properties. They therefore define **Maturity Level 1**. Representations used solely for visualization and lacking a closed geometric description are classified as **Maturity Level 0**.

Solid models are based on topologically closed boundary representations and capture geometric, volumetric, and, where density is assigned, material information [4]. This enables the derivation of physical properties such as mass and center of gravity and supports the use of models as rigid bodies in assemblies. These capabilities define **Maturity Level 2**.

Parametric modeling introduces variable dimensions and explicit dependencies within and between models while maintaining consistency through constraint management. VDI 2209 identifies the documentation of design intent through parameter relationships as a key benefit of this representation [4]. This capability defines **Maturity Level 3**, at which dimensions can be modified within predefined limits and geometric elements can be added, removed, or adapted without reconstructing the model.

Feature-based modeling extends parametric representations by combining geometric entities with semantic information. Features represent non-geometric product characteristics and serve as information and integration objects across CAD, CAPP, CAM, and FEM systems [4, 30]. In addition, modeling history records the sequence and dependencies of modeling operations and determines subsequent editability and reuse [4]. Together, these characteristics define **Maturity Level 4**, enabling structured access to product information across engineering disciplines.

The highest level is defined by the explicit representation of design intent. According to VDI 2209, design intent is reflected in model structure and dependency relationships [4]. The guideline emphasizes that only information embedded directly in the product model can be reused efficiently and largely automatically in downstream processes, whereas information contained in derived documentation generally requires manual interpretation [4]. However, detailed reasoning is easier to implement in a separate document. When doing that, it is important to adequately map the text to the geometric entities of the CAD model. Consequently, **Maturity Level 5** requires a feature history enriched with semantic and functional rationale, such as sizing models, design rules, engineering constraints, or formally defined interfaces and tolerances. At this level, the model captures not only geometry and structure but also the rationale underlying engineering decisions.

Table 1 summarizes the six maturity levels and relates each level to model quality, representation form, and engineering reusability.

Table 1: Definition of the maturity levels of geometry representation.

Maturity level	**Quality of results**	**Typical representation**	**Design usability**
Level 0 (ML0)	Visual plausibility	Rendering or 2D mesh model	Concept illustration, visualization, and model reduction
Level 1 (ML1)	Geometric shape	3D mesh, static B-rep geometry, STL	Measurement of dimensions, collision checks
Level 2 (ML2)	Valid CAD geometry	STEP or a neutral CAD format	Mechanical properties (center of gravity and weight), material, rigid models for assemblies

Level 3 (ML3)	Parametric CAD model	CAD script or design sequence	Adjust dimensions within limits; add and remove design elements
Level 4 (ML4)	Feature-based model history	Editable feature tree with parameters and dependencies	Structured access to design elements from the perspective of different disciplines (CAE, CAM, etc.)
Level 5 (ML5)	CAD model with design intent	Feature history with semantic and functional justification	Additional annotations regarding the design intent (analytical design models, design guidelines, structural interfaces with tolerance specifications)

4. Literature Review Method

To answer RQ2, we conducted a systematic literature review of LLM-based CAD generation approaches.

4.1. Academic Databases

To collect a body of literature, we searched three academic databases (Scopus, IEEE Xplore, and Web of Science) for papers matching the following query, adapted to each database's native search syntax:

```
("generative design*" ∨ "algorithmic design*" ∨ "CAD" ∨ "computer aided design" ∨ "computer-aided design") ∧ ("LLM*" ∨ "large language model*" ∨ "foundation model*" ∨ "transformer model*")
```

The query combines terms related to generative and computer-aided design with terms covering large language models and related architectures, using wildcards (*) to capture morphological variants (e.g., "generative design" and "generative designs"). This initial search returned a total of 1577 papers (574 from IEEE Xplore, 731 from Scopus, and 272 from Web of Science) matching at least one term from each group, which were then passed to LLM-based screening against the inclusion and exclusion criteria described below.

4.2. LLM-based Inclusion and Exclusion Screening

A search query in scholarly databases only returns exact word matches. Therefore, we leverage an LLM for second screening because they can also identify semantic matches without intended search objective. To this end we translated the search string into three prompts consisting of a question, inclusion, and exclusion criteria. The result of the LLM is a binary decision. We purposely made the decision binary to reduce risk of hallucination often occurring when LLMs need to fulfil a task beyond their capabilities. The prompts containing the question, inclusion, and exclusion criteria are listed in the worksheet “Exclusion” of the Excel file that can be accessed through [31].

We employ a multi-stage screening process applied to the literature corpus including the use of LLMs for semantic filtering of large literature corpora. The original 1577 records were reduced through three successive rounds of LLM-based inclusion and exclusion screening against the criteria defined in Section 4.2, narrowing the corpus to 502, then 219, and finally 81 candidate papers (see Figure 1). Currently a growing number of literature review publications rely on LLMs [1]. First studies show that literature reviews integrating LLM-based screening are actually more efficient and accurate compared to review based entirely on human judgement [3,4]. Combining human capabilities to LLMs, a final round of human verification confirmed 51 papers as the basis for the concept-matrix analysis presented in the following sections.

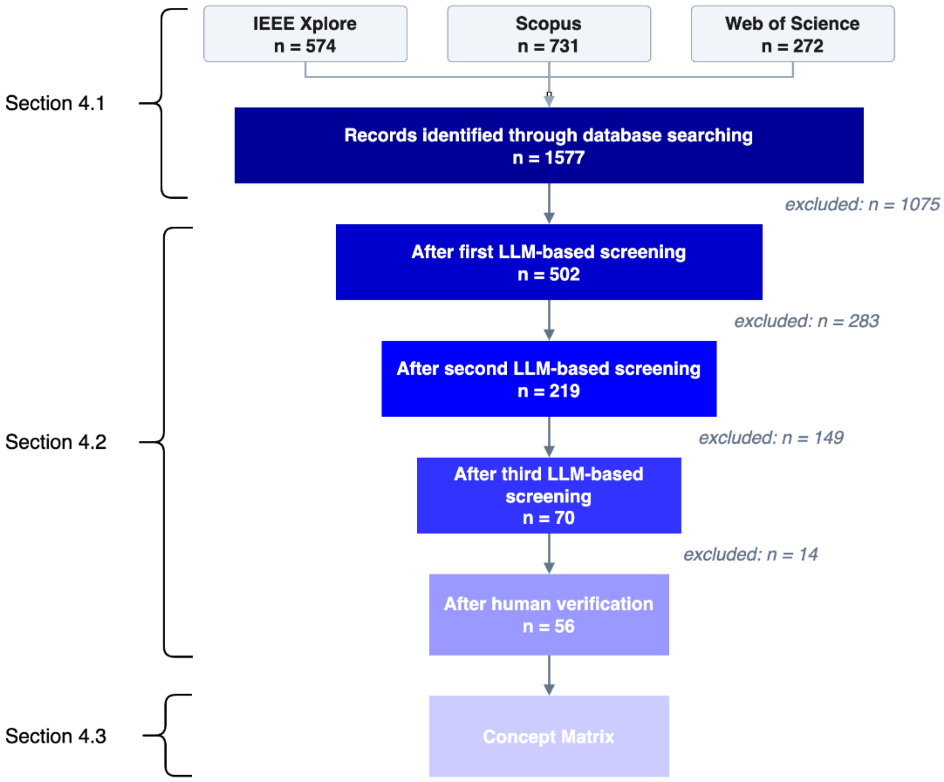

Figure 1: Literature screening funnel, from initial database search (n = 1577) to the final set of included studies (n = 51) after three rounds of LLM-based screening and human verification.

4.3. Concepts

To address RQ2, we extract each paper's methodological choices along eight dimensions grounded on relevant concept defined in the Background chapter: paper type, adaptation strategy, learning method, training data, input modality, application scope, output representation, and target CAD software or format. Each maps onto a factor our maturity model (Section 3) identifies as shaping design reusability. Output representation, for example, sets the ceiling on the maturity level a given approach can reach, while adaptation strategy and training data determine whether domain-specific design knowledge enters the model at all. For each paper we mark the applicable category in a concept matrix. This concept-centric structure allows us to compare across papers along methodologically relevant axes rather than summarizing each paper in isolation.

There is also always the option that each category is not relevant or mentioned in a paper. This lets us compare which combinations of data, inputs, outputs, and training methods correlate with higher maturity levels, rather than describing each paper individually. We rate the maturity level on the output of the LLM only. When the LLM only changes parameter values in an existing CAD model, but the existing CAD model has maturity level 4, the maturity level of the CAD-generation is still just maturity level 3, since the LLM only provided a change on maturity level 3 by changing parameter values.

Arriving at a small and relevant body of research, we follow the concept matrix approach [32, 33], which organizes a literature review around concepts rather than individual papers. We have used both two human evaluators and an LLM to review the content of the discovered literature against our review criteria defined above to fill the concept matrix. For reproducibility the prompt for the LLM on this task is provided in the worksheet “Criteria Prompt” of the Excel file, that can be accessed through [31].

5. Results

The results of the literature review can be accessed through [31] using the worksheet "Concept Matrix". The IDs correlate with the worksheet "References". Our rating of the papers is based on the maturity level reached by the CAD model output of the LLM, and whether the LLM creates an entirely new CAD model, creates an entirely new CAD model by combining predefined templates or whether it modifies an existing CAD model. For modifying existing CAD models, a distinction must be made as to whether only the parameters are changed, whether the CAD model itself is changed (e.g., through additional extrusions), or whether this change is made using predefined templates. Some approaches can also generate new CAD models and change existing ones. Often, they change the newly generated CAD model as part of a feedback loop to improve the result. A clear concentration can be observed at maturity level 3, with 89% of the approaches reaching this level. The combination of the achieved maturity level and the type of generation is shown in Figure 2. In that figure, only the most impressive achievement of each approach is shown.

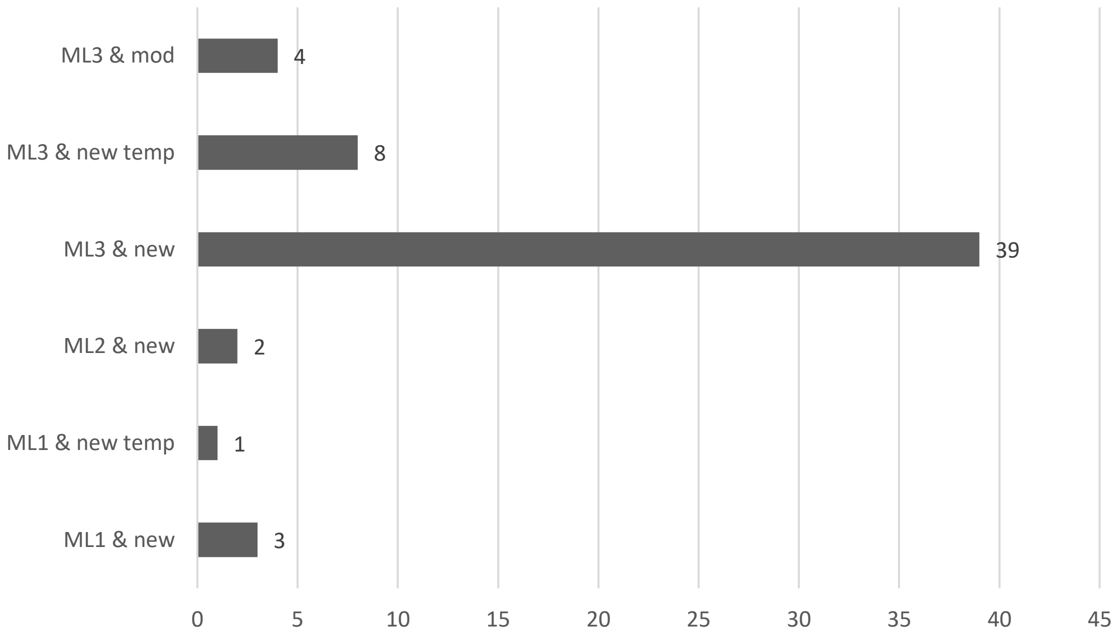

Figure 2: Bar chart of the number of approaches which generate different combinations of maturity levels and applications. "new" means that new CAD models are generated from scratch. "temp" means that predefined templates are used. "mod" means that CAD models are only modified, not newly generated.

As can be seen in Figure 2, the majority of the approaches (70%) reach maturity level 3 and create new CAD models. Full training of a new LLM from scratch is used in one case (2.5% of these approaches). In 41% of these approaches, fine-tuning is used to train the LLM for geometry generation. In the remaining 56.5% of the approaches, only prompt conditioning is used. This shows that prompt conditioning is sufficient to generate CAD models of this kind. However, training the LLM appears promising for achieving higher quality.

14.3% of the approaches also create new CAD models at maturity level 3, but use predefined templates to do so. In that case, an important part of design intelligence is provided by the predefined templates rather than by the LLM itself. While this makes it easier to achieve a high maturity level of the generated CAD models, it also limits its usability and generalizability because it shifts the CAD creation logic from the LLM to predefined and therefore limited templates. In only one (14.3%) of those approaches, fine-tuning is used to train the LLM. All other approaches use prompt conditioning. When predefined templates are used, fine-tuning can be applied, but does not appear to provide sufficient benefits to justify the effort.

7.1% of the approaches reach maturity level 3 but do not create new CAD models. Instead, they only modify existing models. Since the CAD models already exist, reaching maturity level 3 is less impressive. One of these approaches (25%) uses fine-tuning. The remaining approaches use prompt conditioning. Since the CAD model already exists, modifying it is easier than creating a new CAD model, and prompt conditioning therefore appears to be sufficient.

3.6% of the approaches reach maturity level 2 and generate new CAD models. Here, the LLMs are fine-tuned for this task using CAD datasets. Although this approach is relatively advanced, the achievable maturity level is limited by the CAD training data: boundary representations (B-Reps) and NURBS-based representations from STEP files are used for training and consequently determine the representation of the output.

5.4% of the approaches reach only maturity level 1 and generate new CAD models. In these approaches, the LLMs are not specifically trained for this task. Instead, prompt conditioning is used. For generating simple mesh-based models or STL files, this is sufficient.

1.8% of the approaches reach maturity level 1 and generate new CAD models using predefined templates. Here, prompt conditioning is used instead of additional training. By using templates, this approach is not universally applicable, but is limited to a specific component.

96.4% of all approaches use some form of script-based geometry description as output, which is subsequently converted into a geometric model. This is a more robust way to generate geometry than generating geometry files directly. Many of the models are trained on simple geometries without engineering context taken from the DeepCAD dataset or from adapted or reduced versions of it. The review also reveals a complete lack of assembly generation. All identified approaches either generate only individual parts, use templates of existing simple assemblies, or change existing assemblies. This represents a significant research gap, especially when considering the fact that the dimensions of individual parts often depend on one another. The same CAD geometry can be modelled in different ways. A cylinder, for example, can be created by extruding a circle or by revolving a rectangle around an axis. This aspect is not addressed by any of the approaches.

6. Discussion

The results reveal a gap between the current state of LLM-based CAD generation and the quality criteria required in engineering design practice. Contemporary literature mostly leverages built-in capabilities of LLMs by generating CAD as Python source code, resulting in ML3 CAD objects without much effort. Another direction of research uses LLMs to intelligently parametrize existing CAD templates. While these approaches can reach ML4, the engineering intelligence resides in the pre-built templates, rather than in the LLM merely supplying the parameters. It can also be seen in Section 5 that no approach surpasses ML3. The lack of adaptability and editability, which would be necessary to reach ML4 along with the lack of detailed annotations and explanations, which would be necessary to reach ML5, is a research gap that should be addressed.

However more complex CAD data such as naming conventions, parameters and uniform modelling guidelines are not yet integrated into LLM-based generation. A possible reason for this mismatch is that much of the research is driven by computer science rather than mechanical engineering, resulting in a mismatch between AI research and practical need.

A further gap is the absence of material or manufacturing method selection in the design process. In engineering design combining existing components into new assemblies is more important than creating new designs from scratch. Hence also CAD objects generated by LLMs should match this work.

This positions LLM-based approaches relative to classical generative design methods. The classical methods are positioned between ML2 (topology optimization with volumetric meshes) and ML4 (editable feature tree with design sequence, dependencies and parameters).

As a possible direction for future research, we see integration of engineering design knowledge and normative engineering standards. With deeper integration with CAD software, the LLMs could reach ML4 by using editable feature trees.

7. Conclusion and Outlook

This paper introduced a six-level maturity model for evaluating LLM-generated CAD models beyond geometric similarity, and applied it to a systematic review of current approaches. Results show that most approaches cluster at maturity level 3, limited primarily by output representation, training data, and a research focus driven more by computer science than mechanical engineering. None of the reviewed approaches reach feature-based modelling or assembly generation. Future work should therefore focus on fine-tuning strategies that embed design knowledge and standards into generative models, and on deeper integration with parametric CAD software to enable higher-maturity, industrially reusable outputs.

References

[1] Xu, Jingwei ; Wang, Chenyu ; Zhao, Zibo ; Liu, Wen ; Ma, Yi ; Gao, Shenghua: *CAD-MLLM: Unifying Multimodality-Conditioned CAD Generation With MLLM*. 2024

[2] Camba, Jorge D. ; Contero, Manuel ; Company, Pedro: *Parametric CAD modeling: An analysis of strategies for design reusability*. In: *Computer-Aided Design* 74 (2016), S. 18–31

[3] Zhang, Licheng ; Le, Bach ; Akhtar, Naveed ; Lam, Siew-Kei ; Ngo, Duc: *Large Language Models for Computer-Aided Design: A Survey*. In: *ACM Computing Surveys* 58 (2026), Nr. 9, S. 1–39

[4] VDI 2209. 2009. *3D-Produktmodellierung. Technische und organisatorische Voraussetzungen, Verfahren, Werkzeuge und Anwendungen, wirtschaftlicher Einsatz in der Praxis*

[5] Verhagen, Wim J. C. ; Bermell-Garcia, Pablo ; van Dijk, Reinier E. C. ; Curran, Richard: *A critical review of Knowledge-Based Engineering: An identification of research challenges*. In: *Advanced Engineering Informatics* 26 (2012), Nr. 1, S. 5–15

[6] Herrmann, Kevin ; Altun, Osman ; Wolniak, Pilipp ; Mozgova, Iryna ; Lachmayer, Roland: Methodischer Aufbau von Entwicklungsumgebungen nach dem Generative Parametric Design Approach. In: *DS 111: Proceedings of the 32nd Symposium Design for X (DFX2021)*, 2021, S. 1–10

[7] Krish, Sivam: *A practical generative design method*. In: *Computer-Aided Design* 43 (2011), Nr. 1, S. 88–100

[8] Bendsøe, M. P. ; & Sigmund, O.: *Topology optimization: theory, methods, and applications*, 2004

[9] Sigmund, Ole ; Maute, Kurt: *Topology optimization approaches*. In: *Structural and Multidisciplinary Optimization* 48 (2013), Nr. 6, S. 1031–1055. URL https://link.springer.com/article/10.1007/s00158-013-0978-6

[10] Wang, Michael Yu ; Wang, Xiaoming ; Guo, Dongming: *A level set method for structural topology optimization*. In: *Computer Methods in Applied Mechanics and Engineering* 192 (2003), 1-2, S. 227–246

[11] van Dijk, N. P. ; Maute, K. ; Langelaar, M. ; van Keulen, F.: *Level-set methods for structural topology optimization: a review*. In: *Structural and Multidisciplinary Optimization* 48 (2013), Nr. 3, S. 437–472. URL https://link.springer.com/article/10.1007/s00158-013-0912-y

[12] Herrmann, Kevin ; Plappert, Stefan ; Gembarski, Paul ; Lachmayer, Roland: *Process Chain-Oriented Design Evaluation of Multi-Material Components by Knowledge-Based Engineering*. In: *Algorithms* 16 (2023), Nr. 5, S. 247

[13] Chakrabarti, Amaresh ; Shea, Kristina ; Stone, Robert ; Cagan, Jonathan ; Campbell, Matthew ; Hernandez, Noe Vargas ; Wood, Kristin L.: *Computer-Based Design Synthesis Research: An Overview*. In: *Journal of Computing and Information Science in Engineering* 11 (2011), Nr. 2

[14] Herrmann, Kevin ; Pusch, Felix ; Becker, Christian ; Gembarski, Paul Christoph ; Lachmayer, Roland: *Modeling constraint-based manufacturing networks: theory, application and best practices*. In: *Proceedings of the Design Society* 5 (2025), S. 2461–2470

[15] Radford, A. ; Wu, J. ; Child, R. ; Luan, D. ; Amodei, D. ; Sutskever, I.: *Language models are unsupervised multitask learners* : OpenAI blog, 2019

[16] Berger, Elias ; Usama, Muhammad ; Mehlstäubl, Jan ; Saske, Bernhard ; Paetzold-Byhain, Kristin: *Physics-in-the-Loop: A Hybrid Agentic Architecture for Validated CAD Engineering Design*. 2026

[17] ALRASHEDY, K., TAMBWEKAR, P., ZAIDI, Z. H., LANGWASSER, M., XU, W., GOMBOLAY, M: *Generating cad code with vision-language models for 3d designs*. In: *International Conference on Learning Representations* Vol. 2025 (2025), S. 52236–52262

[18] PUSCH, Felix ; BERGER, Elias ; GEMBARSKI, Paul Christoph ; MEHLSTÄUBL, Jan ; HERRMANN, Kevin ; PAETZOLD-BYHAIN, Kristin ; LACHMAYER, Roland: *Labeling CAD Parts to capture Design Intent for LLMs through Machining Feature Recognition*. In: *Procedia CIRP* 142 (2026), S. 43–48

[19] WANG, Ruiyu ; YUAN, Yu ; SUN, Shizhao ; BIAN, Jiang: *Text-to-CAD Generation Through Infusing Visual Feedback in Large Language Models*, 2025

[20] BERGER, Elias ; HERRMANN, Kevin ; PUSCH, Felix ; KRIESELL, Tobias ; GEMBARSKI, Paul ; MEHLSTÄUBL, Jan ; LACHMAYER, Roland ; PAETZOLD-BYHAIN, Kristin: *From geometry to function: towards context-aware generative AI for engineering design*. In: *Proceedings of the Design Society* 6 (2026), S. 2193–2202

[21] MAKATURA, L. ; FOSHEY, M. ; WANG, B. ; HÄHNLEIN, F. ; MA, P. ; DENG, B. ; TJANDRASUWITA, M. ; SPIELBERG, A. ; OWENS, C. ; CHEN, P. Y. ; ZHAO, A. ; ZHU, A. ; NORTON, W. ; GU, E. ; JACOB, J. ; LI, Y. ; SCHULZ, A. ; MATUSIK, W.: *How can large language models help humans in design and manufacturing? Part 1: Elements of the LLM-enabled computational design and manufacturing pipeline* : Harvard Data Science Review, 2024

[22] XU, Jingwei ; WANG, Chenyu ; ZHAO, Zibo ; LIU, Wen ; MA, Yi ; GAO, Shenghua: *CAD-MLLM: Unifying Multimodality-Conditioned CAD Generation With MLLM*. In: *arXiv preprint arXiv:2411.04954* (2024)

[23] SCHÜPBACH, Aurel ; SAN MIGUEL, Raul ; FERCHOW, Julian ; MEBOLDT, Mirko: *From text to design: a framework to leverage LLM agents for automated CAD generation*. In: *Proceedings of the Design Society* 5 (2025), S. 1893–1902. URL https://www.cambridge.org/core/product/identifier/S2732527X25102034/type/journal_article

[24] XU, Xiang ; WILLIS, Karl D. D. ; LAMBOURNE, Joseph G. ; CHENG, Chin-Yi ; JAYARAMAN, Pradeep Kumar ; FURUKAWA, Yasutaka: *SkexGen: Autoregressive Generation of CAD Construction Sequences with Disentangled Codebooks*. arXiv, 2022

[25] DORIS, Anna C. ; ALAM, Md Ferdous ; HEYRANI NOBARI, Amin ; AHMED, Faez: CAD-Coder: An Open-Source Vision-Language Model for Computer-Aided Design Code Generation. In: *Volume 3A: 51st Design Automation Conference (DAC)* : American Society of Mechanical Engineers, 2025

[26] GOVINDARAJAN, Prashant ; BALDELLI, Davide ; PATHAK, Jay ; FOURNIER, Quentin ; CHANDAR, Sarath: *CADmium: Fine-Tuning Code Language Models for Text-Driven Sequential CAD Design*. arXiv, 2026/01/16/17:09:50

[27] MAKATURA, L. ; FOSHEY, M. ; WANG, B. ; HÄHNLEIN, F. ; MA, P. ; DENG, B. ; TJANDRASUWITA, M. ; SPIELBERG, A. ; OWENS, C. ; CHEN, P. Y. ; ZHAO, A. ; ZHU, A. ; NORTON, W. ; GU, E. ; JACOB, J. ; LI, Y. ; SCHULZ, A. ; MATUSIK, W.: *How can large language models help humans in design and manufacturing? Part 2: Synthesizing an end-to-end LLM-enabled design and manufacturing …* : Harvard Data Science Review, 2024

[28] OUYANG, Shuyin ; ZHANG, Jie M. ; HARMAN, Mark ; WANG, Meng: *An Empirical Study of the Non-Determinism of ChatGPT in Code Generation*. In: *ACM Transactions on Software Engineering and Methodology* 34 (2025), Nr. 2, S. 1–28

[29] BECKER, Jörg ; KNACKSTEDT, Ralf ; PÖPPELBUß, Jens: *Developing Maturity Models for IT Management*. In: *Business & Information Systems Engineering* 1 (2009), Nr. 3, S. 213–222

[30] VDI 2218. 2003. *Informationsverarbeitung in der Produktentwicklung. Feature-Technologie*

[31] HERRMANN, Kevin ; PUSCH, Felix ; BERGER, Elias ; MEHLSTÄUBL, Jan ; PAETZOLD-BYHAIN, Kristin ; LACHMAYER, Roland: *Supplements for Quality of CAD Generation with Large Language Models: A critical Overview*. LUIS, 2026. URL https://data.uni-hannover.de/dataset/4e75cb36-2840-48d8-8289-72981dd1a682

[32] SALIPANTE, P. ; NOTZ, W. ; BIGELOW, J.: *A matrix approach to literature reviews*. In: *Research in organizational behavior* 4 (1982), S. 321–348

[33] WEBSTER, Jane ; WATSON, Richard T.: *Analyzing the Past to Prepare for the Future: Writing a Literature Review*. In: *MIS Quarterly* 26 (2002), Nr. 2, S. xiii–xxiii