Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10577

From geometry to function: towards context-aware generative AI for engineering design

Elias Berger[1,2,✉], Kevin Herrmann[3], Felix Pusch[3], Tobias Kriesell[3], Paul Gembarski[3], Jan Mehlstäubl[2], Roland Lachmayer[3] and Kristin Paetzold-Byhain[1]

[1] Dresden University of Technology, Germany, [2] MAN Truck & Bus SE, Germany, [3] Leibniz University Hannover, Germany

✉ elias.berger@tu-dresden.de

ABSTRACT: Current generative artificial intelligence for Computer-Aided Design (CAD) optimizes for geometric similarity, neglecting engineering criteria like functionality, manufacturability, and sustainability. This paper addresses this gap and proposes a conceptual framework to reorient generative CAD from replicating shapes to achieving function. We introduce two hybrid training strategies: a pre-learning approach using synthetically labeled datasets (evaluated via FEA, CAM, LCA) and a self-learning approach where GenAI uses these knowledge-based tools as a reinforcement feedback loop.

KEYWORDS: artificial intelligence (AI), computer-aided design (CAD), design automation

1. Introduction

Generative Artificial Intelligence (GenAI) is advancing rapidly and increasingly shaping engineering disciplines. Within the domain of Computer-Aided Design (CAD), large language models (LLMs) open new opportunities. They can generate CAD from textual or visual descriptions in a data-driven manner (Wang et al., 2025; Wu et al., 2021).

However, current generative CAD methods evaluate their outputs using geometric similarity metrics such as Chamfer Distance (CD), Intersection-over-Union (IoU), or Normal Consistency (NC) (Alam & Ahmed, 2025; Wang et al., 2025; Wu et al., 2021). These metrics quantify how closely a generated geometry resembles a reference shape, but in engineering design form is not as important as function (Pahl et al., 2007; Ullman, 2009). Consequently, existing models optimize for geometric rather than functional criteria, thereby neglecting the fundamental objective of the engineering design process.

Engineering design, however, is driven by design intent and fulfilling functional requirements. Designs emerge not in isolation, but in response to specific requirements, constraints, and objectives such as load cases, manufacturing technologies, or environmental considerations (Roth, 2000). Thus, form is the result rather than the starting point of the design process.

To create genuine value for engineering design, GenAI systems must therefore learn to follow this function-oriented process. However, current LLM architectures exhibit fundamental limitations in structured, logical reasoning and the application of domain-specific engineering knowledge (Shojaee et al., 2025). The resulting research gap lies in extending generative CAD models to incorporate engineering-oriented value criteria and context-sensitive learning mechanisms. From this, the central research questions (RQ) of this work arise:

1. Which engineering design aspects enable context-aware generative CAD?
2. How can training strategies integrate multi-objective engineering constraints?
3. What are the trade-offs between pre-learning and self-learning strategies?

To address these questions, we introduce a framework for evaluating and training generative CAD models beyond geometric criteria, using the function-oriented design process as a reference model for

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press

context-aware AI systems. Unlike existing approaches that optimize for geometric similarity, our framework optimizes for functional requirements through two complementary strategies: pre-learning on synthetically generated and deterministically evaluated CAD datasets, and self-learning via iterative optimization through functional evaluation feedback loops. This hybrid approach aligns generative models with engineering design practice by incorporating engineering-relevant evaluation dimensions as learning signals.

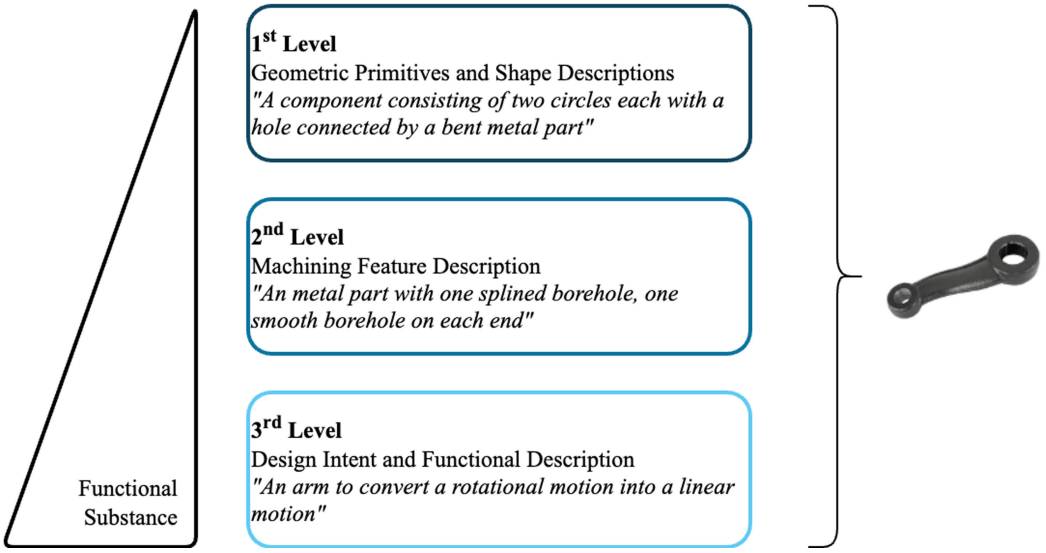


Figure 1. Current research into generative methods focuses on replicating shapes given a literal text description or image. The text description describes the outer shape of the CAD object and corresponds to the first level. For AI to be truly useful in engineering it must advance from 1st level to 2nd or 3rd to level and translate features or design intent into CAD. In this paper we propose a concept to achieve 2nd level

2. Background

2.1. Data-driven and generative methods for engineering design

Recently, the application of generative methods based on LLMs for CAD has gained momentum (Alrashedy et al., 2025; Li et al., 2024; Usama et al., 2025). Several studies demonstrate the ability of LLMs to generate parametric CAD models from textual or multimodal instructions. For instance, (He et al., 2025) describes a text-based system for generating editable CAD sketches in DXF format, while (Wang et al., 2025) presents robust CAD reconstructions from text prompts.

However, the optimization of these generative CAD models has so far been primarily based on geometric similarity metrics such as CD, IoU, or NC. These metrics quantify the shape similarity between the generated and reference models but fail to capture functional properties or manufacturability. Studies (Preintner et al., 2025; Xu et al., 2025) explicitly confirm this limitation. This training strategy leads models to primarily memorize shape patterns rather than follow underlying design intents or functionalities.

The result is a geometric bias in current systems: generated geometries appear plausible but often leave open whether they are manufacturable or functionally suitable. This gap between generative performance and engineering objectives forms the starting point for the research questions addressed in this paper.

2.2. Knowledge-based engineering design

To address the described limitations of data-driven systems, it is useful to consider knowledge-based generative design approaches. Knowledge-based generative systems use explicitly formulated engineering knowledge, physical models, and process-specific boundary conditions to automatically generate and evaluate new designs. Instead of learning from example data, they operate on formalized representations and derive valid geometries from them (Pahl et al., 2007). These approaches therefore differ fundamentally from data-driven models, which identify patterns in existing data but often lack reference to physical plausibility or technical feasibility (Morales, 2025; Srivastava & Kawakami, 2023).

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press

A central approach within this class is the optimization of objective functions, in which geometric and functional targets (e.g., stiffness, mass, temperature distribution) are mathematically formulated and optimized under given constraints.

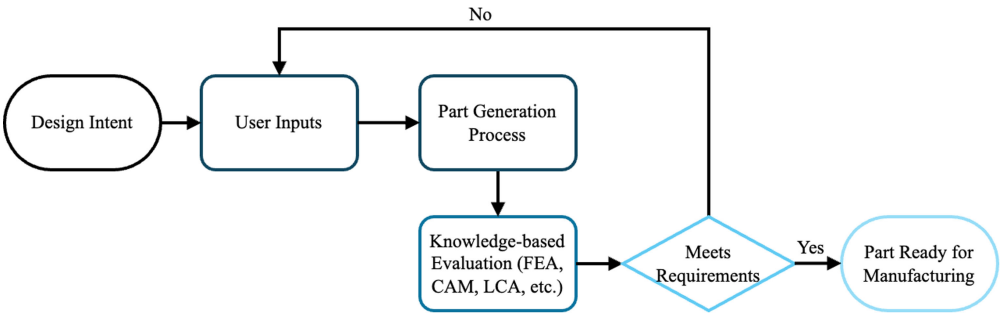


Figure 2. The function-oriented design process (Morales, 2025; Pahl et al., 2007)

A prominent example is topology optimization, where the material distribution within a design space is varied to achieve a defined performance criterion (e.g., maximum stiffness-to-weight ratio) (Bendsøe & Sigmund, 2013). Recent studies extend topology optimization to metaphysical problems in order to simultaneously account for heat conduction, fluid flow, vibration behaviour, or emission output (Hederberg & Thore, 2025; Herrmann et al., 2022). This results in geometries that meet multiple physical requirements simultaneously, such as components with integrated cooling that are both thermally efficient and mechanically robust (Bode et al., 2023).

Another well-established approach is parametric generative design, in which CAD models are described by predefined parameters and dependency rules. These parameters are varied using optimization methods to automatically generate functionally and manufacturability-optimized design variants (Albrecht & Anderl, 2016; Herrmann et al., 2021). Modern systems combine parametric modelling with AI-assisted search optimization to efficiently explore large design spaces (Barbieri & Muzzupappa, 2022). These systems represent engineering knowledge in the form of rules, constraints, or semantic networks and can automatically derive valid design alternatives from them (Herrmann et al., 2023).

These knowledge-based methods have in common is that they build on formalized or experience-based engineering knowledge. They are particularly advantageous when data is scarce or inconsistent for example, in early development phases or for novel product classes. However, their main drawback lies in the significant effort required for knowledge acquisition, modeling, and system maintenance. The knowledge must be formalized, validated, and continuously updated, which limits the applicability of such systems to well-defined domains (Stokes, 2001; Verhagen et al., 2012).

2.3. Hybrid approach to engineering design

The comparison of both paradigms reveals a complementary tension: while data-driven models excel in scalability and creative variability, they often lack physical and functional grounding. Knowledge-based approaches, in contrast, ensure technical validity but are resource-intensive to build and adapt. This leads to the central research problem: how can data-driven generative CAD systems be extended to incorporate functional and context-sensitive knowledge? A promising perspective lies in hybrid approaches that combine both paradigms. This can be achieved, on the one hand, through highly annotated, semantically rich training datasets that explicitly represent the design knowledge required for learning. On the other hand, data-driven models can be coupled with knowledge-based tools—such as simulation models, rule systems, or optimization functions—allowing them to acquire the necessary knowledge autonomously. In this way, the performance of data-driven AI can be combined with the validity of knowledge-based methods.

3. Method

This work follows the Design Research Methodology (DRM) framework (Blessing & Chakrabarti, 2009), which structures engineering design research into four stages: Research Clarification, Descriptive Study I, Prescriptive Study, and Descriptive Study II. Our research addresses the three research questions through a combination of descriptive and prescriptive studies, as outlined in Table 1.

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press

Table 1. Mapping of DRM stages to our research

Stage	Activities	RQs Addressed
Research Clarification	Literature review of generative CAD methods and knowledge-based engineering; Identification of geometric bias in current approaches.	RQ 1-3
Descriptive Study I	Analysis of current generative CAD evaluation practices; Characterization of engineering design aspects. Review of knowledge-based evaluation tools	RQ 1
Prescriptive Study	Development of two-approach framework and integration strategy for hybrid systems	RQ 2
Descriptive Study II	Analysis and discussion of trade-offs between approaches including a proof-of-concept validation	RQ 3

4. Required contextual information and design knowledge for context-sensitive generative AI systems

Generative CAD systems process two kinds of information: **project-specific context** and **transferable engineering knowledge**. Context changes frequently (machine fleet, supply chains, regulations), whereas knowledge is comparatively stable (working principles, design and tolerance rules). If both are mixed, project specifics creep into the rules—hindering reuse, maintenance, and explainability. **Context** defines the application envelope (e.g., "3-axis milling, min. tool radius 2 mm, material 42CrMo4"). **Knowledge** describes, independent of any project, *how* functions are realized and validated geometrically (e.g., "sealing → O-ring groove," "torque transmission → keyway/polygon/spline," "ribs increase stiffness at constant package space").

Many knowledge elements become effective only once concrete context values are bound. **Examples:** From a knowledge perspective, an internal corner "must be filleted"; whether 1.5 mm or 3 mm is permissible is determined by the tool catalog of the current milling center. Lattice structures are sound as a working principle but may create unacceptable overhangs in an SLM context without support strategies. A polygonal shaft can transmit the required torque, but under a pure 3-axis route with available reamers it may be impractical—then a keyway becomes preferable. The knowledge remains general; context supplies parameters. This keeps the system flexible: when context changes (e.g., from 3-axis milling to hybrid additive–subtractive, or from Al6061 to Inconel), parameters are rebound and the same knowledge base yields appropriate alternatives.

For **training and benchmarking**, the separation enables robust comparisons: identical functional tasks can be evaluated under varying contexts without altering the knowledge module. This reveals whether a model truly generalizes with context sensitivity rather than merely reproducing geometry patterns from a specific dataset. **Example:** A "bracket" is assessed once with a 3-axis milling context (tool list, minimum radii, fixturing) and once with an SLM context (overhang limits, support effort)—the knowledge module remains identical.

For **AI training**, the implication is: strictly separate knowledge and context, and couple them only where decisions depend on the environment. **Datasets and labels** should represent general function–geometry knowledge in a context-neutral way; **context parameters** (process, machine limits, tools, material/quality targets, cost/CO_2) are bound **late** and transparently. This includes:

- Separate sources & documentation: Maintain knowledge rules (e.g., minimum wall thickness as process-neutral guidance) separately from context tables (e.g., "milling tool Ø6 mm available → min. hole 6.6 mm"); document assumptions clearly.
- Scenario diversity & context coverage: Vary tasks across contexts (milling ↔ AM, 3- ↔ 5-axis, material and tolerance variants) to promote generalization. *Example:* The same sealing function once with an NBR O-ring and IT7 tolerance band, once with FKM and tighter IT6.
- Negative examples & failure signals: Include explicit "fail" labels such as unmillable internal corners, non-demoldable undercuts, standards violations, or missing tool access—and capture *why* they fail (e.g., "CAM check: collision in pocket").

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press

- Conservative defaults & alternative scenarios: When context is incomplete, start with standard assumptions (e.g., generic 3-axis capabilities) and evaluate alternative contexts in parallel. *Example:* If the supplier is unknown, test variants for aluminium and steel portfolios.
- Evaluation with a fixed knowledge core: Set up benchmarks so that the knowledge module is fixed and only the context changes; keep metrics and checks transparent. *Example:* Same load case and functional targets—tools/parameters drawn from two different machine pools.

This yields a training regime that does not merely produce shape-similar outputs but achieves decision-capable performance within the given environment: the model learns to apply general engineering knowledge cleanly and to translate it—through late, transparent context binding—into project-compliant, manufacturable, and standards-conformant solutions.

5. Proposed data workflow

As established in the background, current generative models for CAD do not optimize towards engineering-relevant criteria. The absence of training datasets annotating CAD models with multi-objective engineering criteria represents a fundamental barrier to advancing generative design methods. To enable models that optimize for functional performance, manufacturability, cost, and sustainability, each CAD object must be paired with quantifiable metrics across these dimensions.

First, we propose a self-learning workflow that leverages established knowledge-based engineering method such as finite element analysis (FEA), computer-aided manufacturing (CAM) simulation, and lifecycle assessment (LCA) in a hybrid AI system. The idea is that the AI will autonomously use knowledge-based method to iteratively refine its output.

Second, we propose a direct, pre-training approach. Given the inputs and final outputs of the prior approach we could collect an annotated datasets with CAD objects and engineering evaluation metrics. This section present both approaches in detail.

5.1. Approach 1: self-learning approach

First, we propose a self-learning framework for generative engineering based on reinforcement learning (see Figure 3). This approach is based on the function-oriented design process in Figure 1 but adapted to the capabilities of GenAI. The concept integrates engineering value criteria introduced in Chapter 4 and combines knowledge-based systems with generative AI models. Traditional knowledge-based engineering (KBE) systems encode decades of domain expertise in areas such as manufacturability, cost estimation, and material selection. Instead of reproducing these capabilities through data-driven learning, our approach leverages them as a feedback mechanism that guides model optimization.

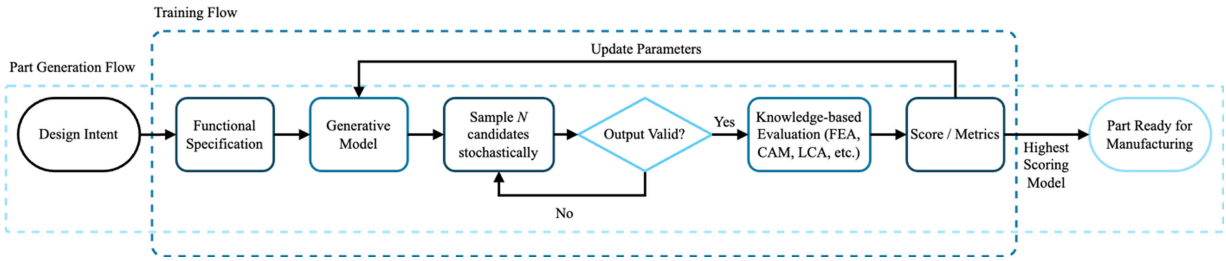


Figure 3. Self-learning framework for generative engineering. The framework couples a generative model with knowledge-based evaluation. Candidate designs are generated stochastically and iteratively assessed against engineering value criteria. The feedback loop enables the model to update its parameters based on validated performance metrics, converging toward functionally sound, manufacturable designs without requiring large, labelled datasets

Technically, the workflow follows a reinforcement learning scheme: for a given functional specification (e.g., a load case), the generative model produces N candidate CAD sequences using a probabilistic decoding strategy. Invalid or incomplete designs are removed by a quality filter, while the remaining candidates are subjected to deterministic evaluation through simulation or analysis tools (e.g., finite element analysis, CAM, or life-cycle assessment). For a stable reinforcement learning process, based on the design intent, select a small, defined set of properties to avoid unforeseen inconsistencies between

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press

properties. Each evaluated design receives a score reflecting its performance with respect to the targeted engineering criteria. High-scoring candidates reinforce desirable model behaviour, whereas low-scoring ones induce negative updates. This closed feedback loop enables the model to gradually learn to generate geometries that are not only geometrically valid but also functionally and economically optimal.

5.2. Approach 2: pre-learning approach

Instead of iteratively refining a CAD object, in practice it is preferred to have the GenAI learn to directly output an ideal geometry. We follow this paradigm in our second approach by assembling a dataset of function specifications and ideal geometries for a specification, training the GenAI to replicate this ideal geometry. This approach resembles the machine learning process of replicating data samples from a training dataset. The pre-training approach uses the established setup of labelled data that a deep learning model can replicate. To this end, we require a large, labelled corpus that links CAD objects to functional specifications. Such a resource does not exist, so we propose a synthetic-data workflow to create it. The process is depicted in Figure 4.

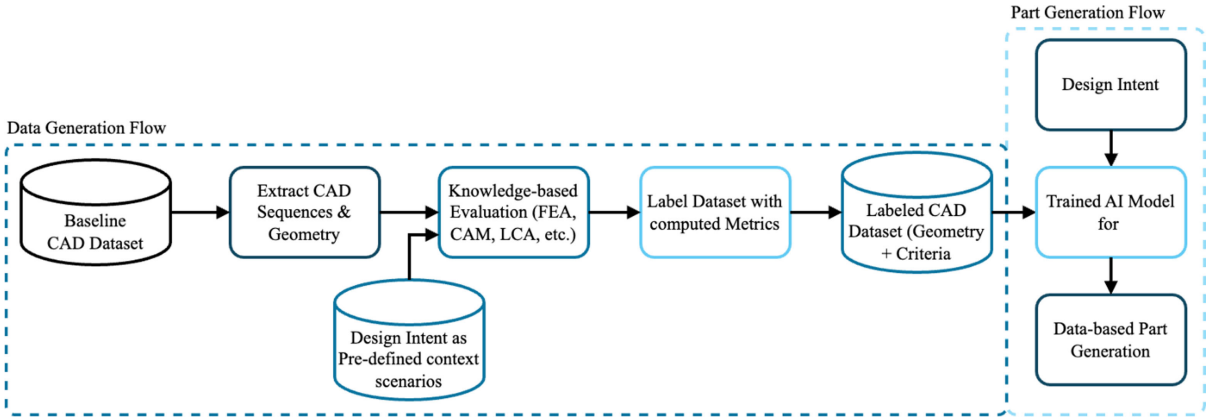


Figure 4. Approach 2: Synthetic data generation workflow for function-conditioned CAD modeling. The process constructs a labeled dataset linking functional specifications to ideal geometries. Starting from a baseline CAD dataset, sequences and geometries are extracted and evaluated using knowledge-based to compute engineering metrics. These evaluations label each sample with quantifiable performance criteria, forming a dataset that trains a generative AI model to directly produce geometry fulfilling given design intents eliminating the need for iterative refinement

First, we select a baseline dataset that exposes parametric construction history or B-Reps (e.g., DeepCAD, Fusion360 Gallery, or the shaft dataset). These sources provide sketches, operations, and topology needed for downstream analyses. Second, we assemble functional specifications for the CAD models such as specific load cases, weight constraints, etc. This allows us to perform deterministic evaluation per CAD object using knowledge-based engineering tools to compute criterion-aligned scores: functionality via feature recognition or lightweight FEA against simple load cases; manufacturability via CAM toolpath feasibility and feature machinability checks; sustainability via material usage, scrap, and estimated machining energy; and cost via process-based estimates combining cycle time with machine and material rates. Third, label the dataset by storing each CAD instance together with its computed metrics, producing geometry–attribute pairs that reflect engineering value rather than mere geometric similarity. Finally, use the labeled data in two ways: train surrogate models to approximate expensive evaluators, and fine-tune generative CAD models so their output distribution shifts toward higher-scoring designs. This pipeline operationalizes engineering value as learnable signals and provides a practical path to optimization beyond geometry.

6. Discussion

6.1. Approach 1 – self-learning

The reinforcement learning approach addresses several limitations of pre-learning by enabling direct optimization for engineering objectives. This approach represents a true hybrid methodology that

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press

integrates generative AI with established knowledge-based engineering systems—coupling the exploratory power of large language models with the validated analysis capabilities of FEA, CAM, and LCA tools. By formulating functionality, manufacturability, cost, and sustainability as components of a reward function, the model can learn to balance competing design criteria through iterative exploration. This approach naturally supports multi-objective optimization through reward shaping, enabling trade-off discovery via Pareto-optimal solutions or preference learning. Critically, reinforcement learning can target design criteria absent from any existing dataset, offering greater generalization potential beyond memorized patterns. The model learns through interaction with deterministic knowledge-based evaluation oracles (FEA, CAM, LCA), potentially discovering design strategies not represented in static training data.

Table 2. Summary of the advantages and challenges of each approach. The self-learning approach is more flexible in its application, but more challenging to develop and train. The pre-learning approach can only generalize from training data but is easier to implement

	Approach 1: Self-Learning	Approach 2: Pre-Learning
Advantages	• Direct optimization for engineering objectives • Multi-objective trade-off discovery • Generalizes beyond dataset distribution • Adaptive improvement through interaction	• Leverages existing CAD datasets • Enables surrogate model training • Scalable benchmarking after labeling • Straightforward implementation
Challenges	• Computationally expensive • Training stability challenges • Extensive hyperparameter tuning re- • quired	• Limited generalization beyond source dataset coverage • Tool-specific biases in labels • Static supervision only • Idealized load case assumptions

Nevertheless, significant implementation challenges limit immediate deployability. Computational expense poses the primary barrier—each design candidate requires 10-60 seconds for complete engineering evaluation, and convergence typically demands hundreds to thousands of iterations. Unlike supervised learning where gradients flow directly from labels, reinforcement learning must explore the design space through trial and error, amplifying computational requirements by orders of magnitude. Reward sparsity compounds this challenge: many generated CAD sequences are geometrically invalid, fail meshing procedures, or violate basic manufacturability constraints, yielding no useful gradient signal. Feature recognition errors and simulation failures introduce noise that degrades learning stability. Training stability presents additional complexity—large language models require careful regularization strategies including proximal policy optimization (PPO) with KL divergence constraints, curriculum learning schedules, and periodic surrogate model audits to prevent mode collapse or divergence. Extensive hyperparameter tuning across learning rates, exploration strategies, and reward scaling factors proves necessary for convergence, substantially increasing development effort relative to supervised approaches.

6.1.1. Experimental validation

To demonstrate feasibility as a proof-of-concept, we conducted a simplified validation using a Pitman arm CAD model. This experiment implements the self-learning workflow in simplified form, demonstrating that the proposed iterate-simulate-evaluate cycle is executable in practice, albeit with a constrained design problem. We equipped the Claude 4.5 with tool-calling capabilities to interact with Autodesk Inventor, enabling it to modify geometry, execute FEA, and evaluate results. The design objective was straightforward: minimize weight while maintaining structural integrity under a specified load. The reward signal combined two metrics: low mass and low Von Mises stress with no pre-training required, as the LLM operated in a reinforcement-learning-like paradigm using direct simulation feedback. Claude 4.5 was selected for its reasoning capabilities and visual input processing, but did not perform any weight updates of the model. A sample iteration is shown in Figure 5.

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press

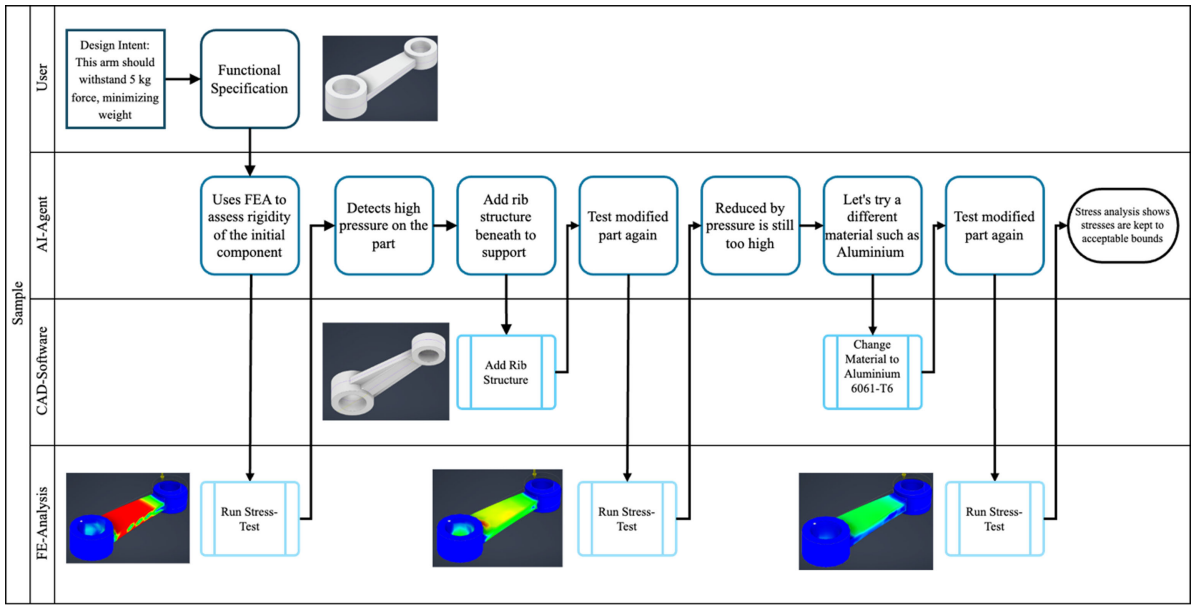


Figure 5. Flowchart summarizing one of our experiments: an interaction between a data-driven method (AI agent) and two knowledge-based systems (CAD software and finite-element analysis software) for designing an arm component. Although simplified, the experiment demonstrates that a data-driven method can effectively utilize and interpret knowledge-based tools

While this experiment is intentionally simplified and does not capture real-world design complexity, it validates our core premise: that generative AI can effectively leverage and respond to knowledge-based engineering tools. The LLM's reasoning in tool selection and design iteration was both comprehensible and goal-directed, demonstrating the viability of coupling data-driven agents with simulation-based evaluation. It only adds appropriate amounts of material, displaying its ability to balance volume and material stress. However, our experiments are not a comprehensive validation, but rather an existence proof that the proposed framework can work for generative CAD.

6.2. Approach 2 – pre-learning

The pre-learning approach offers practical advantages as it is straight-forward to implement. By leveraging existing parametric CAD datasets (e.g., DeepCAD) and annotating them with deterministic metrics, this method provides a straightforward path to creating engineering-aware training data. Once labeled, the dataset enables efficient training of surrogate models that can rapidly approximate expensive simulation-based evaluations. After initial annotation costs, models can be benchmarked efficiently without repeated simulation overhead, facilitating systematic comparison across architectures and training configurations

However, fundamental limitations constrain the approach's scope. Dataset coverage is inherently bounded by the diversity of geometries present in source repositories, underrepresenting complex or domain-specific designs (Berger et al., 2025). More critically, labels generated through automated analysis tools reflect the assumptions made during label generation, but do not generalize to new scenarios.

The methodology necessarily relies on idealized load cases and standardized manufacturing scenarios, whereas real-world engineering involves complex and changing constraints. Generated labels capture only a single evaluation snapshot under fixed assumptions, failing to account for the iterative refinement and evolving requirements characteristic of authentic design processes. This approach provides only static supervision—it does not enable the generative model to adaptively improve through interaction with engineering objectives, limiting its ability to discover novel design solutions beyond the training distribution.

7. Limitations

Our proposed method faces constraints. First, it requires functional requirements to be quantifiable through computational metrics. This assumption that holds for load-bearing requirements evaluated via

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press

FEA or cost assessments derived from bills of materials and machining expenses but becomes problematic for less well-defined design criteria such as repairability, serviceability, or user experience. Second, the computational demands are substantial. LLM inference incurs both significant time and compute costs. While these costs are comparable to established computational design methods like topology optimization, they may limit practical deployment in resource-constrained or time-sensitive design contexts.

8. Conclusion

Current generative CAD methods optimize for geometric similarity rather than engineering value, using metrics like Chamfer Distance that ignore functional performance and manufacturability. We proposed a framework that reorients generative CAD toward engineering objectives through two complementary strategies: pre-learning on synthetically labeled data and self-learning via reinforcement signals from simulation feedback. This approach enables benchmarking against functional and manufacturing requirements, supports multi-objective trade-offs, and establishes foundations for hybrid systems combining data-driven learning with knowledge-based evaluation. By shifting focus from shape fidelity to performance and feasibility, we move toward generative systems that better reflect engineering reasoning and design intent.

References

Alam, M. F., & Ahmed, F. (2025). GenCAD: Image-Conditioned Computer-Aided Design Generation with Transformer-Based Contrastive Representation and Diffusion Priors (arXiv:2409.16294). arXiv. https://doi.org/10.48550/arXiv.2409.16294

Albrecht, K., & Anderl, R. (2016). Information model for the integration of manufacturing restrictions into the algorithm based product development process. *Procedia CIRP*, 50, 819–824. https://doi.org/10.1016/j.procir.2016.04.136

Alrashedy, K., Tambwekar, P., Zaidi, Z., Langwasser, M., Xu, W., & Gombolay, M. (2025). Generating CAD Code with Vision-Language Models for 3D Designs (arXiv:2410.05340). arXiv. https://doi.org/10.48550/arXiv.2410.05340

Barbieri, L., & Muzzupappa, M. (2022). Performance-driven engineering design approaches based on generative design and topology optimization tools: A comparative study. *Applied Sciences*, 12(4), 2106. https://doi.org/10.3390/app12042106

Bendsøe, M. P., & Sigmund, O. (2013). *Topology optimization: Theory, methods, and applications* (2nd ed.). Springer.

Berger, E., Dammann, M. P., Mehlstäubl, J., Saske, B., Braun, F., & Paetzold-Byhain, K. (2025). Challenges and opportunities in the integration of generative AI with computer-aided design. *Proceedings of the Design Society*, 5, 881–890. https://doi.org/10.1017/pds.2025.10102

Blessing, L. T. M., & Chakrabarti, A. (2009). DRM, a design research methodology (1st ed., p. XVII, 397). Springer London. https://doi.org/10.1007/978-1-84882-587-1

Bode, B., Herrmann, K., Reusch, J., Plappert, S., Ehlers, T., Gembarski, P. C., Hasse, C., & Lachmayer, R. (2023). Thermo-elastic topology optimization for high temperatures gradients using load separation. *Procedia CIRP*, 119, 576–581. https://doi.org/10.1016/j.procir.2023.03.113

He, C., Zhang, S., Zhang, L., & Miao, J. (2025). CAD-Coder:Text-Guided CAD Files Code Generation (arXiv:2505.08686). arXiv. https://doi.org/10.48550/arXiv.2505.08686

Hederberg, H., & Thore, C.-J. (2025). Fluid–structure interaction topology optimization using density jumps for implicit boundary representation. *Structural and Multidisciplinary Optimization*, 68(6), 122. https://doi.org/10.1007/s00158-025-04045-4

Herrmann, K., Altun, O., Wolniak, P., Mozgova, I., & Lachmayer, R. (2021). Methodischer aufbau von entwicklungsumgebungen nach dem generative parametric design approach. *Proceedings of the 32nd Symposium Design for X*, DFX 2021. https://doi.org/10.35199/dfx2021.14

Herrmann, K., Bode, B., Wurst, J., Gembarski, P. C., Mozgova, I., & Lachmayer, R. (2022). Quantification of the material-related environmental impact of topology-optimized multi-material components. *Proceedings of the 33rd Symposium Design for X*, DFX 2022. https://doi.org/10.35199/dfx2022.01

Herrmann, K., Plappert, S., Gembarski, P., & Lachmayer, R. (2023). Process chain-oriented design evaluation of multi-material components by knowledge-based engineering. *Algorithms*, 16(5), 247. https://doi.org/10.3390/a16050247

Li, X., Sun, Y., & Sha, Z. (2024). LLM4CAD: Multimodal Large Language Models for Three-Dimensional Computer-Aided Design Generation. *Journal of Computing and Information Science in Engineering*, 25(021005). https://doi.org/10.1115/1.4067085

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press

Morales, E. D. C. (2025). *Comparative analysis of generative design and topology optimization in mechanical components* [Master, Purdue University]. https://hammer.purdue.edu/ndownloader/files/56139845

Pahl, G., Beitz, W., Feldhusen, J., & Grote, K.-H. (2007). Engineering Design A Systematic Approach. In *Engineering design: A systematic approach* (pp. 125–143). Springer. https://doi.org/10.1007/978-1-84628-319-2

Preintner, T., Yuan, W., König, A., Bäck, T., Raponi, E., & van Stein, N. (2025). EvoCAD: Evolutionary CAD code generation with vision language models. https://arxiv.org/abs/2510.11631

Roth, K. (2000). Konstruieren mit konstruktionskatalogen: Band 1: Konstruktionslehre (3rd ed., p. XVIII+440). wangSpringer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-642-17466-7

Shojaee, P., Mirzadeh, I., Alizadeh, K., Horton, M., Bengio, S., & Farajtabar, M. (2025). The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. https://arxiv.org/abs/2506.06941

Srivastava, J., & Kawakami, H. (2023). Systematic Review of Difference Between Topology Optimization and Generative Design. 22nd IFAC World Congress, 56(2), 6561–6568. https://doi.org/10.1016/j.ifacol.2023.10.307

Stokes, M. (2001). *Managing engineering knowledge: MOKA – methodology for knowledge based engineering applications*. Professional Engineering Publishing.

Ullman, D. G. (2009). *The mechanical design process* (4th ed.). McGraw-Hill.

Usama, M., Khan, M. S., Stricker, D., & Afzal, M. Z. (2025). NURBGen: High-fidelity text-to-CAD generation through LLM-driven NURBS modeling. https://arxiv.org/abs/2511.06194

Verhagen, W. J. C., Bermell-Garcia, P., van Dijk, R. E. C., & Curran, R. (2012). A critical review of Knowledge-Based Engineering: An identification of research challenges. *Advanced Engineering Informatics*, 26(1), 5–15. https://doi.org/10.1016/j.aei.2011.06.004

Wang, S., Chen, C., Le, X., Xu, Q., Xu, L., Zhang, Y., & Yang, J. (2025). CAD-GPT: Synthesising CAD Construction Sequence with Spatial Reasoning-Enhanced Multimodal LLMs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(8), 7880–7888. https://doi.org/10.1609/aaai.v39i8.32849

Wu, R., Xiao, C., & Zheng, C. (2021). DeepCAD: A Deep Generative Network for Computer-Aided Design Models (arXiv:2105.09492). arXiv. https://doi.org/10.48550/arXiv.2105.09492

Xu, J., Zhao, Z., Wang, C., Liu, W., Ma, Y., & Gao, S. (2025). CAD-MLLM: Unifying Multimodality-Conditioned CAD Generation With MLLM (arXiv:2411.04954). arXiv. https://doi.org/10.48550/arXiv.2411.04954

https://doi.org/10.1017/pds.2026.10577 Published online by Cambridge University Press