International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

CADGENIUS: An AI-Based Assistant for AutoCAD

Tanvi Sonawane (Zeal Polytechnic, Pune)
Shravani Phalke (Zeal Polytechnic, Pune)
Santoshi Kale (Zeal Polytechnic, Pune)
Shruti Darwatkar (Zeal Polytechnic, Pune)

Abstract – Computer-aided design (CAD) tools are powerful but demanding: faults are found late, hand-drawn intent is ambiguous, teamwork is awkward, and design history is rarely captured in reusable form. This paper surveys the techniques that underpin CADGENIUS, a proposed AI-assisted CAD companion built on five features: DesignGuard AI, Smart Sketch Clarification, CAD Sync Talk, Voice and Visual Commands, and AI Record Tracker. We describe the review approach, present a layered architecture and workflow, review the literature behind each feature, and compare the features on maturity, data needs, real-time suitability and explainability. The synthesis shows that deterministic geometry checks, learned models and language-model orchestration are complementary, and that provenance logging is the feature that ties the others together. Limitations and open problems are also discussed.

Keywords – CAD, generative AI, design validation, sketch understanding, collaborative design, voice interface, provenance.

I. INTRODUCTION

Parametric CAD systems let engineers describe a part as an ordered sequence of sketches and feature operations such as extrusions and fillets. Large datasets of such human design sequences [1], [2] have made it possible to learn from CAD data, and sequence models based on the Transformer [3] now treat CAD construction as a language-modeling problem [4]. At the same time, large language models [5] can follow natural-language instructions, opening the way to assistants that sit inside the design environment rather than beside it.

Day-to-day CAD work still has four recurring pain points. First, design faults such as thin walls, unreachable features or violated standards are often found only at the manufacturing or review stage. Second, freehand sketches are ambiguous, and an under- or over-constrained sketch can fail silently. Third, distributed teams must coordinate edits and discussion across separate tools. Fourth, hands-busy or shop-floor settings make menu-driven interaction slow, and the reasoning behind each design decision is rarely recorded.

CADGENIUS is proposed to address these points through five features: (1) DesignGuard AI, (2) Smart Sketch Clarification, (3) CAD Sync Talk, (4) Voice and Visual Commands, and (5) AI Record Tracker. The contributions of this survey are a consolidated architecture and workflow for the five features, a feature-wise review of supporting research, a comparative synthesis, and a discussion of limitations. Section II explains the review approach, Section III presents the architecture, Section IV reviews each feature, Section V compares them, Section VI lists limitations, and Section VII concludes.

II. REVIEW APPROACH

A narrative, feature-driven review was adopted. Literature was gathered from IEEE Xplore, the ACM Digital Library, arXiv and Google Scholar using keyword combinations such as "CAD generation," "design for manufacturability," "sketch constraint inference," "collaborative editing," "speech recognition," "multimodal language model" and "provenance." Each candidate was screened by title and abstract and retained if it (i) addresses a task that maps directly to one of the five features, (ii) presents a reproducible method, dataset or formal model, and (iii) is a peer-reviewed paper, a widely used preprint or a recognized specification. Foundational works outside CAD, for example in speech and vision-language modeling, were included where CADGENIUS reuses their techniques.

Retained works were mapped to features and compared on four criteria: maturity (research prototype versus deployed practice), data requirements, suitability for real-time use, and explainability of outputs. The comparison is qualitative; no new experiments were run, and CADGENIUS is treated as a design under study rather than a validated product.

III. ARCHITECTURE AND WORKFLOW

Fig. 1 shows the layered architecture. The Voice and Visual Command layer captures speech, pointer selections and sketches. The Intent Orchestrator, a language model with tool-calling access to the CAD application programming interface (API), converts these inputs into structured actions. Three services then act on the shared model: DesignGuard AI validates it, Smart Sketch Clarification resolves sketch ambiguity, and CAD Sync Talk synchronizes edits and discussion between users. The CAD kernel holds the authoritative model, and the AI Record Tracker observes every layer.

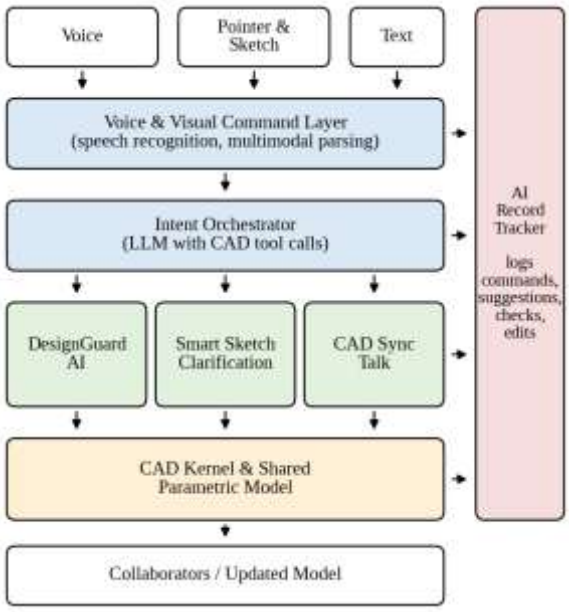


Fig. 1. Layered architecture and workflow of CADGENIUS.

The workflow has six steps. (1) The designer speaks, points or draws. (2) The interaction layer transcribes speech and resolves references such as "this edge" using the current selection. (3) The orchestrator maps the request to CAD operations or to a clarification question. (4) The sketch and validation services check constraints and design rules before the operation is committed. (5) The committed change is broadcast to collaborators together with its discussion thread. (6) The tracker stores the command, its outcome and any accepted or rejected AI suggestion.

IV. FEATURE-WISE REVIEW

A. DesignGuard AI

DesignGuard AI is a proactive validation assistant. Classical work recognizes machining features in solid models through graph-based heuristics [6], and this remains the basis of deterministic design-for-manufacturing (DFM) checks such as minimum wall thickness, hole depth-to-diameter ratio and draft angle. Learning-based methods complement these rules: datasets of real construction sequences [1], [2] allow models to learn what typical, valid designs look like and to flag outliers. For explanation, retrieval-augmented generation [7] can ground a language model's warning in the relevant standard or company guideline instead of free-form text. The literature therefore points to a hybrid design: rules for hard constraints, learned models for soft warnings, and retrieval for explanation.

B. Smart Sketch Clarification

A sketch is a set of primitives plus geometric constraints, which can be represented as a graph. SketchGraphs provides millions of real CAD sketches in this form and supports tasks such as constraint prediction [8]. Generative sequence models of freehand strokes [9] show that drawing intent can be modeled statistically. For image input, vision transformers [10] and vision-language pre-training [11] offer features for recognizing hand-drawn shapes and linking them to text. In CADGENIUS, clarification means detecting that a sketch is under-constrained, over-constrained or visually ambiguous (for example, whether two lines are meant to be parallel) and asking the designer a short question rather than guessing. Reviewed work covers prediction well, but clarification dialogs remain little studied.

C. CAD Sync Talk

CAD Sync Talk couples real-time co-editing with contextual conversation. Operational transformation (OT) [12] and conflict-free replicated data types (CRDTs) [13] are the two mainstream families for merging concurrent edits without locking. Both were designed for text-like data. Parametric CAD is harder because feature operations depend on earlier features and on persistent naming of faces and edges, so a remote edit can invalidate a local one. Practical designs therefore synchronize at the feature-tree level, attach chat messages to specific features, and use the AI layer to summarize discussions and suggest conflict resolutions.

D. Voice and Visual Commands

Speech recognition has improved markedly with large-scale weakly supervised training [14], giving usable accuracy across accents and noisy audio without per-user training. Multimodal language models trained with visual instruction tuning [15] can interpret screenshots and annotated views, which supports deictic commands such as "make this hole larger." Mapping recognized intent to precise CAD operations is a tool-use problem, in which language models learn to call external APIs [16]. Key design choices are to require confirmation for destructive operations, to echo the interpreted command back to the user, and to fall back gracefully to menus.

E. AI Record Tracker

The AI Record Tracker maintains a structured history of what was done, by whom or by which AI component, and why. Provenance models such as the Open Provenance Model [17] describe artifacts, processes and agents, and map naturally to CAD, where parts are artifacts and feature operations are processes. Recording voice commands, AI suggestions, user decisions and design-rule results supports audit, onboarding, rollback and design-rationale retrieval. Recorded human design sequences are also the training data on which learning-based CAD methods depend [1], so a well-structured tracker can improve the other four features over time, subject to privacy and consent.

V. COMPARATIVE SYNTHESIS

Table I compares the five features. DesignGuard AI and Voice and Visual Commands rest on the most mature building blocks, whereas Smart Sketch Clarification and CAD Sync Talk need CAD-specific adaptation of general methods. All features benefit from a shared representation of the design, and the AI Record Tracker is the common data source.

TABLE I
COMPARISON OF CADGENIUS FEATURES

Feature	Core techniques	Maturity	Key gap
DesignGuard AI	Rule-based DFM checks, feature recognition, retrieval	High (rules); medium (learned)	Verified rule base; false alarms
Smart Sketch Clarification	Constraint graphs, vision transformers	Medium	Few studies of clarification dialogs
CAD Sync Talk	OT / CRDT merging, feature-tree sync	Medium	Conflicts in parametric models
Voice and Visual Commands	ASR, multimodal LLM, tool use	High (ASR); medium (CAD mapping)	Ambiguity; command safety
AI Record Tracker	Provenance models, event logs	High	Privacy; schema design

Two observations follow. First, deterministic components (rule checks, constraint solvers, CRDT merging) give predictable behavior and explainability, whereas learned components give flexibility but need guardrails; CADGENIUS

should place learned models only where a deterministic check can verify or veto their output. Second, latency matters differently per feature: voice and synchronization must respond in near real time, while validation and logging can run asynchronously.

VI. LIMITATIONS

The survey itself is narrative rather than an exhaustive systematic review, so relevant work may be missing, and it treats CADGENIUS as a design proposal: no empirical evaluation of an integrated system is reported. Several technical limitations also emerge from the literature. Labeled CAD data are scarce and skewed toward simple sketch-and-extrude models [1], [2], which limits generalization to complex assemblies. Language models may hallucinate operations or misread ambiguous speech, which is risky when a command changes a model. Real-time collaboration on parametric geometry lacks a standard conflict-resolution model. Cloud-hosted AI raises intellectual-property and privacy concerns for proprietary designs, and detailed record keeping increases that exposure. Finally, no common benchmark evaluates the five features together, so cross-study comparison remains difficult.

VII. CONCLUSION

This paper surveyed the research behind CADGENIUS and its five features, and proposed a layered architecture and workflow that connects them. The review suggests a hybrid strategy: deterministic checks for correctness, learned models for flexibility, language models for orchestration and explanation, and provenance logging as a shared backbone. Future work includes building a prototype, evaluating each feature with designers on realistic parts, creating a benchmark for conversational CAD, and studying on-premise deployment for sensitive designs.

REFERENCES

[1] K. D. D. Willis et al., “Fusion 360 Gallery: A dataset and environment for programmatic CAD construction from human design sequences,” ACM Trans. Graph., vol. 40, no. 4, Art. 54, 2021.

[2] R. Wu, C. Xiao, and C. Zheng, “DeepCAD: A deep generative network for computer-aided design models,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), 2021, pp. 6772-6782.

[3] A. Vaswani et al., “Attention is all you need,” in Adv. Neural Inf. Process. Syst. (NeurIPS), 2017, pp. 5998-6008.

[4] Y. Ganin, S. Bartunov, Y. Li, E. Keller, and S. Saliceti, “Computer-aided design as language,” in Adv. Neural Inf. Process. Syst. (NeurIPS), 2021.

[5] T. B. Brown et al., “Language models are few-shot learners,” in Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.

[6] S. Joshi and T. C. Chang, “Graph-based heuristics for recognition of machined features from a 3D solid model,” Comput.-Aided Des., vol. 20, no. 2, pp. 58-66, 1988.

[7] P. Lewis et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Adv. Neural Inf. Process. Syst. (NeurIPS), 2020.

[8] A. Seff, Y. Ovadia, W. Zhou, and R. P. Adams, “SketchGraphs: A large-scale dataset for modeling relational geometry in computer-aided design,” in ICML Workshop Object-Oriented Learn., 2020.

[9] D. Ha and D. Eck, “A neural representation of sketch drawings,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2018.

[10] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2021.

[11] A. Radford et al., “Learning transferable visual models from natural language supervision,” in Proc. Int. Conf. Mach. Learn. (ICML), 2021, pp. 8748-8763.

[12] C. A. Ellis and S. J. Gibbs, “Concurrency control in groupware systems,” in Proc. ACM SIGMOD Int. Conf. Manage. Data, 1989, pp. 399-407.

[13] M. Shapiro, N. Preguiça, C. Baquero, and M. Zawirski, “Conflict-free replicated data types,” in Proc. Symp. Self-Stabilizing Syst. (SSS), 2011, pp. 386-400.

[14] A. Radford et al., “Robust speech recognition via large-scale weak supervision,” in Proc. Int. Conf. Mach. Learn. (ICML), 2023.

[15] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” in Adv. Neural Inf. Process. Syst. (NeurIPS), 2023.

[16] T. Schick et al., “Toolformer: Language models can teach themselves to use tools,” in Adv. Neural Inf. Process. Syst. (NeurIPS), 2023.

[17] L. Moreau et al., “The Open Provenance Model core specification (v1.1),” Future Gener. Comput. Syst., vol. 27, no. 6, pp. 743-756, 2011.