Vibe-CADing: Conditional CAD Generation and Retrieval for a Text-to-CAD Design Pipeline

Masoud Jafaripour
Edmonton, Canada

Abstract

Vibe-CADing is an agentic, language-driven generative design pipeline for producing, refining, and retrieving CAD parts from natural language descriptions. The system unifies multiple modeling paradigms: (1) voxel-based 3D generation using diffusion models and VAE-based latent compression, and (2) autoregressive PixelCNN-style generators that produce 2D CAD token grids which can be extruded into 3D geometry. To align generated geometry with designer intent, Vibe-CADing incorporates a feedback-guided refinement module based on symmetry, center-of-mass, structural heuristics, and manufacturability constraints. A RAG-like retrieval component enables geometric search over open-source CAD libraries to guide generation and provide structural priors. At the orchestration level, we employ a multi-agent architecture built with LangChain and LangGraph, consisting of a planner, retriever, generator, and critic. These agents collaborate over shared state to decompose natural language prompts, retrieve similar CAD exemplars, synthesize candidate geometries, and iteratively verify or refine outputs. Together, these components form a unified, modular pipeline that bridges intuitive human expression with precise, context-aware, and manufacturable CAD modeling.

1 Introduction

Computer-Aided Design (CAD) has long been central to mechanical and product design, yet it remains inaccessible to non-experts due to the steep learning curve of parametric modeling interfaces and symbolic workflows. Natural language, by contrast, offers an intuitive and expressive modality for communicating design intent. Inspired by the notion of “vibe-coding”—generative code creation guided by high-level intent—we introduce *Vibe-CADing*: a generative and retrieval-augmented pipeline that transforms free-form language prompts into manufacturable 3D part designs.

Vibe-CADing integrates two complementary model families for geometry synthesis. The first family focuses on voxel-based 3D generation using generic conditional generative models—including Variational Autoencoders (VAEs), VQ-VAEs van den Oord et al. [2017], autoregressive models, and diffusion models Ho et al. [2020], Sanghi et al. [2022]—which enable coarse-to-fine synthesis and refinement of CAD-like shapes. The second family consists of a PixelCNN-style autoregressive model van den Oord et al. [2016b] for generating 2D CAD layouts as token grids, which can be exported as SVG/DXF or extruded into 3D voxel structures. Both families operate within a feedback-driven refinement loop that evaluates geometric properties such as symmetry, center-of-mass, and structural balance to guide iterative improvement.

Complementing the generative models, the system incorporates Retrieval-Augmented Generation (RAG) capabilities Chase [2022], enabling reuse or adaptation of components from open-source CAD repositories. Retrieval is performed using a combination of text embeddings and visual embeddings (e.g., CLIP Radford et al. [2021]) to capture user intent from both language and GUI interactions, as well as geometry-aware embeddings of 3D shapes (e.g., PointNet++ Qi et al. [2017]) to support structure-sensitive similarity search over large part libraries.

To orchestrate these components, Vibe-CADing adopts an agentic architecture built with LangChain and LangGraph Team [2023], comprising specialized agents for planning, retrieval, geometry generation, and constraint-based critique. This multi-agent loop integrates retrieval, generative synthesis, and feedback evaluation into a unified, modular system that iteratively aligns generated geometry with user intent and manufacturability requirements.

2 Related Work

Text-to-3D Generation. Early work such as Text2Shape Chen et al. [2018] explored learning joint embeddings between text and voxelized shapes for retrieval and GAN-based synthesis. CLIP-Forge Sanghi et al. [2022] extended this idea by leveraging CLIP embeddings with implicit neural fields to enable zero-shot text-to-shape generation. Subsequent optimization-based approaches such as DreamFusion Poole et al. [2022] and Magic3D harness 2D diffusion priors to iteratively optimize NeRFs, while Shap-E and Point-E Nichol et al. [2022] provide fast autoregressive or diffusion architectures for mesh and point cloud generation. Despite impressive visual results, these approaches lack geometric precision, constraint reasoning, or manufacturability awareness.

Programmatic and Parametric CAD Generation. Language-to-CAD Wang et al. [2023] introduced autoregressive modeling of CAD construction sequences to generate parametric B-Rep programs from text. DeepCAD Liu et al. [2020] and related works focused on learning distributions over CAD sketch and extrusion operations. Although programmatic approaches improve symbolic fidelity, they remain limited in language grounding, iterative refinement, or integration with retrieval and constraint verification.

Autoregressive Models for Layout and Token Grids. PixelCNN and masked autoregressive models van den Oord et al. [2016a] have been used for 2D layout and structured grid generation. Recent extensions in 2D-to-3D extrusion pipelines demonstrate how grid-based CAD sketches can be transformed into usable geometry. Vibe-CADing builds on this by producing text-conditioned CAD token grids that can be extruded and further refined via 3D diffusion or VAE decoders.

Agentic LLM Systems. Agent-based frameworks such as LangChain Chase [2022], LangGraph Team [2023], AutoGen, and MCP provide mechanisms for multi-agent coordination, shared state management, and modular reasoning. While widely explored in software automation and knowledge workflows, their application to multimodal CAD reasoning remains underdeveloped. Vibe-CADing extends these frameworks to coordinate retrieval, generation, and constraint checking across geometry-aware agents.

Geometric Retrieval. Most retrieval-augmented systems rely on text or image embeddings. Geometric retrieval—matching CAD parts based on 3D structure—is considerably more challenging. Prior work has explored shape-based descriptors and CLIP-adapted embeddings Sanghi et al. [2022], but scalable, CAD-specific geometric RAG pipelines remain largely unaddressed. Our system incorporates geometric retrieval as a core step in the design loop.

3 Our Approach

Vibe-CADing unifies generative modeling, geometric retrieval, and agentic reasoning into a modular text-to-CAD pipeline. The system is orchestrated using LangChain and LangGraph, which manage tool invocation, state transitions, and multi-agent collaboration. At the core lies a multi-agent refinement loop composed of four specialized agents:

- **Planner**: Interprets natural language prompts, extracts part specifications, and decomposes the request into actionable subgoals. It also determines whether generation, retrieval, or hybrid synthesis is required.
- **Retriever**: Performs geometric retrieval over open-source CAD libraries, using shape embeddings to locate structurally or functionally similar parts. Retrieved examples serve as priors, constraints, or refinement targets.

- **Generator**: Synthesizes candidate geometries using one of two model families: (i) voxel-based generators such as VAEs or diffusion models for coarse-to-fine 3D synthesis, and (ii) autoregressive PixelCNN-style models that produce 2D CAD token grids which can be exported or extruded into 3D. Retrieved components may optionally condition or guide these models.
- **Critic**: Evaluates the generated or retrieved geometry using symmetry metrics, center-of-mass and balance scoring, structural heuristics, and manufacturability constraints. When violations occur, it provides structured feedback to the planner for the next refinement iteration.

Agents exchange information through a shared, persistent state managed by LangGraph, with the option of using the Model Context Protocol (MCP) for structured multi-modal memory. This architecture supports iterative improvement cycles, tight coupling between text and geometry, and flexible integration of retrieval, generative models, and constraint checking. Together, these components form a scalable pathway toward interactive, GUI-driven, assembly-aware, and manufacturability-informed CAD generation directly from natural language.

References

Harrison Chase. Langchain, 2022. GitHub: `https://github.com/hwchase17/langchain`.

Kevin Chen, Christopher B. Xu, Rohit Girdhar, William T. Freeman, Joshua B. Tenenbaum, and Wojciech Matusik. Text2shape: Generating shapes from natural language by learning joint embeddings. *arXiv preprint arXiv:1803.08495*, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Yutong Liu, Yewen Zhang, Panos Achlioptas, Leonidas Guibas, Bipul Deka, and Niloy J. Mitra. Deepcad: A deep generative network for computer-aided design models. *arXiv preprint arXiv:2005.11090*, 2020.

Alex Nichol, Joshua Achiam, et al. Point-e: A system for generating 3d point clouds from complex prompts, 2022. OpenAI.

Ben Poole, Ajay Jain, Ben Mildenhall, Matthew Tancik, Jonathan T. Liu, and Jonathan T. Barron. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.

Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, et al. Learning transferable visual models from natural language supervision. *ICML*, 2021.

Aditya Sanghi, Mikhail Tchapmi, Chenfanfu Xu, Silvio Savarese, and Li Fei-Fei. Clip-forge: Towards zero-shot text-to-shape generation. In *CVPR*, 2022.

LangChain Team. Langgraph: State machines for llm applications, 2023. GitHub: `https://github.com/langchain-ai/langgraph`.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *ICML*, 2016a.

Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pages 1747–1756, 2016b.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *Advances in Neural Information Processing Systems*, 2017.

Kevin Wang, Tao Gao, Jonathan Tremblay, and Rohit Raina. Language-to-cad: Programmatic cad model generation from natural language instructions. *Autodesk Research*, 2023.