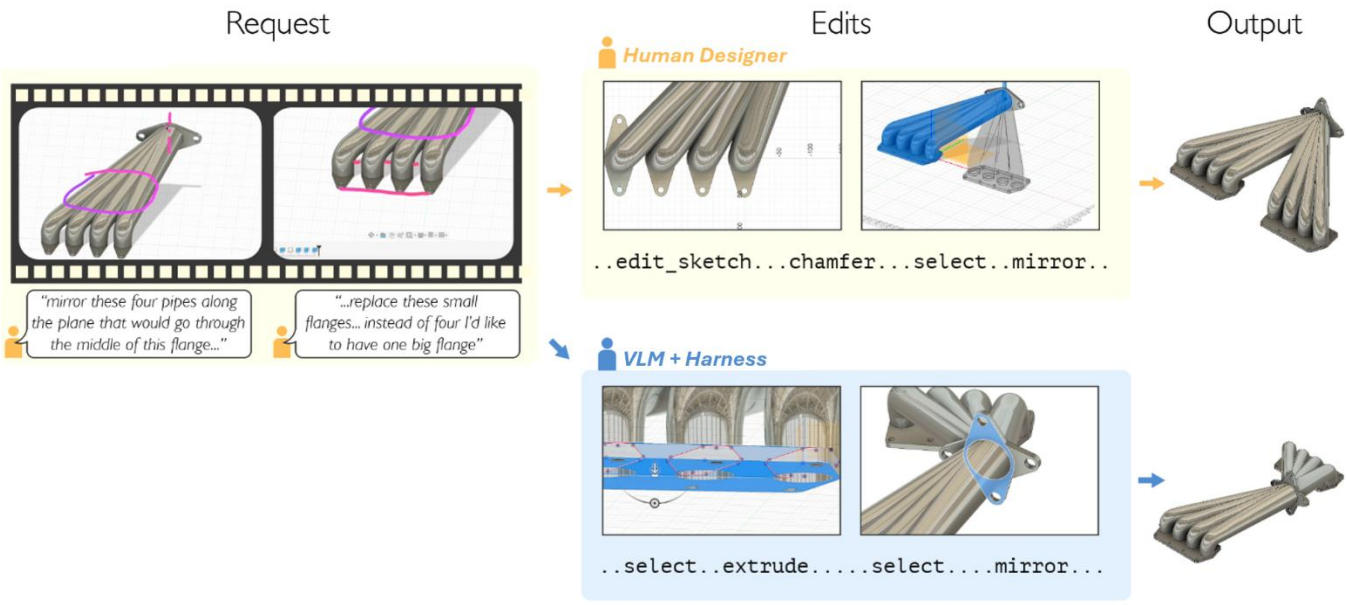
3D CAD Model Editing with Visual Language Models

Autodesk



Problem Background

AI has drastically changed how programmers write code. While improved model performance has driven some of these changes, the biggest leaps in recent years have come from engineering advances: agentic workflows, reasoning steps, test harnesses, and tool integration. As AI systems begin to tackle CAD and mechanical design, there is a real opportunity to apply the same kind of systems-level thinking to drastically extend what these models can do in engineering domains.

Within CAD, most AI research has focused on *generation* (producing models from scratch given text or image inputs). In practice, engineers spend far more time *editing* than generating: refining models to meet tolerances, incorporating collaborator feedback, updating designs as specifications change, and repairing errors. This editing workflow is largely unaddressed by current AI systems, and the benchmarks that do exist are limited to text-only, single-component, or synthetically generated tasks. CAD-Edit-3D addresses this gap with 192 expert-collected multimodal editing requests across varying difficulties, modalities, and model types, and reveals that even the best foundation model (GPT 5.2) performs far worse than human CAD experts. Critically, the results also show that performance depends heavily on the *harness* (how the model interacts with the CAD environment, inspects its own outputs, and iterates on scripts) rather than on raw model capability alone. Models that persisted longer and generated more reasoning tokens performed meaningfully better, suggesting that the engineering around the model (tool use, feedback loops, visual self-inspection) may matter as much as the model itself.

Objective

Design an agentic harness to allow a VLM to edit CAD models in response to text instructions.

Task Description

The task is to design a harness that allows a VLM to edit CAD files. The objective is to maximize performance on automatic metrics.

As a starting point, participants may wish to play around with the following:

- Passing additional metadata about the CAD model and its current state at different times during the edit.
- Using multiple agents to perform and guide different parts of the edit process.
- Providing tools to the VLM to help it in parts of the editing process it struggles with.
- Focus on improving the token efficiency.

Data description

Data will consist of input .step files and a parquet file containing the necessary metadata. We will provide example code that loads the data and runs the task using a sample harness, along with example outputs to clarify the required submission format. Tools for visualizing the data will also be made available to help participants inspect the models and outputs.

Deliverables:

- Code
    - Submit your final code as a pull request to our repository.
- Output CAD files (.step and .stl). Must be in standard format.
    - Include in the repository.
- Scores (computed by code we provide):
    - 3D chamfer distance, Dino feature similarity vs human edit.
- Final presentation
    - Include a PDF of the presentation in your repository
    - Include overview of your method, final scores, images of the 5 selected test examples for qualitative evaluation on 1 slide.

Evaluation criteria:

Category	Criteria	Score
Quantitative Evaluation (40%)	On all text-conditioned edits (48), computed with provided codebase: • 3D chamfer distance • Dino feature similarity	0-10 points
Qualitative Evaluation (40%)	Method evaluation: • Novelty • Cost efficiency • Scalability	0-10 points
Other (20%)	Presentation quality, structure, and clarity. Future work and broader impact discussion.	0-10 points

Points of Contact

Daniele Grandi
daniele.grandi@autodesk.com

Toby Perret
toby.perrett@autodesk.com

William McCarthy
william.mccarthy@autodesk.com