Proceedings in MANUFACTURING SYSTEMS Proceedings in Manufacturing Systems, Volume 21, Issue 1, 2026, pp. 17-34 ISSN 2067-9238

FROM 2D DRAWINGS TO 3D SOLIDS IN FREECAD: AN LLM-BASED PILOT STUDY ON PERFORMANCE AND PROMPTING STRATEGIES

Diana POPESCU[1,*], **Robert IACOB**[2]

[1)] Prof., PhD, Robots and Manufacturing Systems Dep., National University of Science and Technology POLITEHNICA Bucharest, Romania
[2)] Lecturer, PhD), Robots and Manufacturing Systems Dep., National University of Science and Technology POLITEHNICA Bucharest, Romania

https://doi.org/10.5281/zenodo.21818956

Abstract: *The paper presents a pilot study investigating how prompting strategies affect a multimodal large language model (LLM) 's ability to interpret dimensioned 2D technical drawings and generate executable FreeCAD Python scripts for direct 3D solid reconstruction. The test set included nine prismatic mechanical parts with different geometrical features and difficulty levels to evaluate the interpretation of orthographic views and sections, positioning of holes, difference between through, blind, and counterbore holes, reconstruction of pockets, slots, cut-outs, inclined faces, and local edge features, as well as the valid and correct generation of subsequent 3D solids. Two prompting strategies were tested for each part: a simple prompt with minimal constraints and a moderately structured prompt with general rules for multi-view interpretation, feature separation and modelling order. The evaluation considered script execution, valid solid generation, structural and dimensional correctness, and run-to-run stability. Three runs were conducted for each independent prompt-part configuration. The results showed that, for both prompts, the generated scripts were all executed successfully, producing valid FreeCAD solids. The minimal prompt generated 5/9 fully correct parts (55.6%), whereas the moderately structured prompt generated 6/9 fully correct parts (66.7%). All low-complexity parts were reconstructed correctly with both prompts, the structured prompt producing a small improvement in the results for medium-complexity parts. None of the complex parts was correctly reconstructed. These results indicate that the main limitation is not FreeCAD script generation by the LLM, but the interpretation of complex 2D drawings into a correct 3D feature structure.*

Key words: *2D drawings, CAD, Large Language Models, FreeCAD, prompting strategies.*

1. INTRODUCTION

The use of Artificial Intelligence (AI) in mechanical engineering is not new [1]. AI-based methods can support the development of engineering applications [2–3], improve work performance [4], support the decision-making processes [5–6] and assist the optimization process of engineering tasks [7–8]. However, the use of Large Language Models (LLMs) in engineering, especially in the mechanical design domain, is more recent. Reviews in the field showed that this research direction is still at an early stage, but it is rapidly evolving [9–11]. Current applications include code generation, technical information interpretation and the automation of design-related workflows.

Computer-aided design (CAD) is a central tool in mechanical engineering, 3D CAD models being the standard in product development and manufacturing [12–13]. Transforming dimensioned 2D technical drawings into a 3D CAD model is part of an engineer's job and requires technical drawing knowledge, spatial reasoning, as well 2D and 3D CAD skills in operating different 3D CAD environments. The engineer must combine the visual and dimensional information from several views and sections, identify the features and their reciprocal spatial position, interpret hole axes and depths, hidden lines, and then must translate the result into a valid modelling sequence. This is a time-consuming and repetitive task, especially when similar parts need to be converted from 2D into 3D. Therefore, AI-assisted tools could support the partial automation of this process under the supervision of an engineer, reducing the modelling workload and providing an initial 3D CAD geometry for further verification and correction.

Literature survey showed that recent LLM based studies are focused on text-to-CAD generation, CAD-code synthesis or completion of structured CAD-like representations. Applications such as Text2CAD [14], CAD-Llama [15] and CAD-Coder [16] showed that parametric CAD sequences or executable CAD scripts can be generated from textual or structured inputs. CAD2Program [17] investigated the reconstruction of 3D parametric models from 2D CAD drawings treated as raster images, while Drawing2CAD [18] focused on generating CAD commands from vectorized engineering drawings by framing the task as a sequence-to-sequence problem. Other approaches, such as CAD-LLM [19] and CadVLM [20], investigated sketch or CAD primitive completion using symbolic or CAD-derived representations. However, these approaches are, in general, based on trained frameworks, vectorized representations, structured CAD inputs or iterative

* Corresponding author: Splaiul Independenței 313, district 6, 060042, Bucharest, Romania,
Tel.: 0040 21 402 9420,
E-mail address: *dian_popescu@yahoo.com* (D. Popescu).

refinement. In comparison, this study's purpose is to evaluate a commercially available multimodal LLM, namely ChatGPT Model Thinking 5.5 (OpenAI), using dimensioned 2D technical drawings as input to directly generate Python scripts for open-source FreeCAD 1.1.1. This task is different from Text-to-CAD generation as the design intent is not expressed only in words/language. Knowledge must be inferred from views, sections, dimensions, hidden lines, center lines and local annotations. As a result, a generated CAD script may be executed successfully while still producing a geometrically incorrect solid if the intermediate 3D feature structure was wrongly inferred.

The current research enrolls in a new trend of AI-assisted CAD. There is also an industrial interest in such tools, which can be seen in the development of recent applications integrating AI agents or copilots directly into CAD environments, such as Adam for Onshape/Fusion [21], Camfer for SolidWorks [22], MecAgent for mechanical CAD workflows [23], and the more recent integration of Claude with Autodesk Fusion [24]. Their aim is toward conversational and agent-based CAD interaction in engineering design, at the same time reinforcing the need to evaluate whether generated CAD operations are not only executable, but geometrically correct, and to have a documented perspective over the current limitations of such approaches.

In this context, the present paper explores how effective a multimodal LLM is in reconstructing 3D solids from 2D drawings through FreeCAD Python scripting. Besides checking whether the generated scripts can be executed, the study examines how well the LLM transforms 2D drawing information into a correct 3D feature structure by elaborating modeling plans. Thus, the analysis focuses on two main aspects: (1) reliability of the reconstructed geometry, and (2) types of errors that appear during the interpretation process. Another objective of the research is to evaluate if the structured prompting improves the results compared with minimal prompting. The study is conducted on nine prismatic mechanical parts with different levels of difficulty, allowing the results to be discussed in relation to part complexity, prompt structure and run-to-run stability.

2. MATERIALS AND METHOD

As mentioned, ChatGPT Thinking 5.5 was used as the LLM in this research. CATIA V5R21 (Dassault Systèmes) was used to generate the reference 3D parts and the corresponding dimensioned 2D drawings, while the Python scripts generated by the LLM were executed in FreeCAD for 3D part reconstruction. The generated scripts were executed without manual correction.

The LLM-based process was considered as a sequential visual-to-code transformation, rather than as a deterministic CAD-recognition pipeline. This distinction was important, as a generated script can be executed successfully, but still produces an incorrect solid. Therefore, the evaluation process separated the script execution from the structural and dimensional correctness of the parts.

In the study, the reconstruction process included the following steps: *visual drawing interpretation → inferred 3D feature structure → coordinate assignment → FreeCAD operation sequence → Python script synthesis → FreeCAD script execution*.

This conceptual workflow is summarized below:

Input:
D = dimensioned 2D technical drawing
P = prompt
LLM output:
S = generated FreeCAD Python script
Experimental output:
M = generated FreeCAD solid after script execution
Conceptual workflow:
1. *Interpret D under the constraints of P.*
2. *Identify views, visible contours, hidden lines, center lines and dimensions.*
3. *Assign a modelling coordinate system.*
4. *Infer the main prismatic body and its global dimensions.*
5. *Identify added volumes and their positions.*
6. *Identify removed features and classify them by type, position, axis and depth.*
7. *Associate numerical dimensions with the inferred features.*
8. *Convert the inferred 3D feature structure into a FreeCAD operation sequence.*
9. *Generate the complete Python script S.*
10. *Execute S externally in FreeCAD to obtain M.*
11. *Evaluate M against the reference model for script execution, solid validity, structural correctness, dimensional correctness and run-to-run stability.*

2.1. Test set

The test set consisted of nine dimensioned prismatic mechanical parts, P1–P9 (Table 1). They were designed to include different levels of geometric complexity and various features usually found in mechanical drawings, such as superposed prismatic solids, vertical and horizontal holes, through holes, blind holes, counterbored holes, rectangular pockets, elongated slots, bottom cut-outs, inclined faces, chamfered or rounded edges. The first three parts were considered simple parts, the next three were medium-complexity parts, while the last three parts were classified as complex parts. This classification was based on the number of distinct geometric features, the number of required modelling operations, the need to combine information from several orthographic views, and the presence of features with different orientations or depths.

Each part was introduced in ChatGPT as a separate PDF file containing the dimensioned technical drawing (Annex 1). The PDF format was used instead of a raster image format to improve the readability of small dimensions, leader lines, hidden lines, and local feature annotations. It is important to note that, as seen in previous papers [17–18], the accuracy of the generated model is much higher when high quality detailed drawings are used (PDF files), instead of raster images. No additional verbal description of the geometry was provided during the tests.

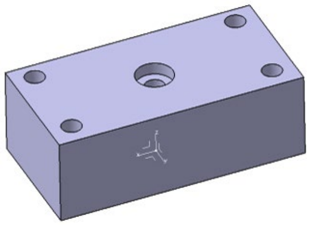
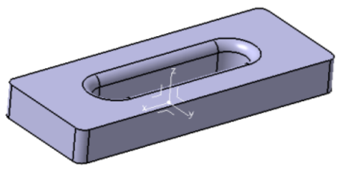
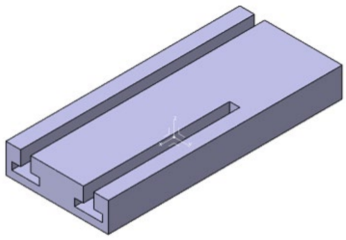
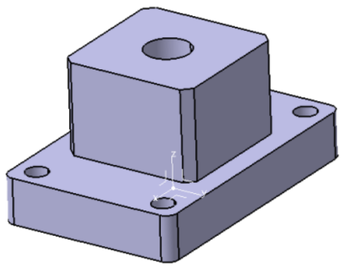
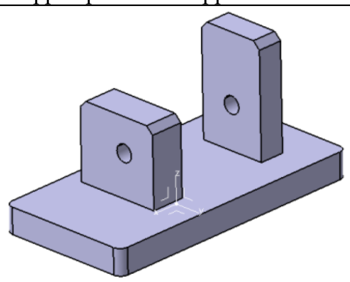
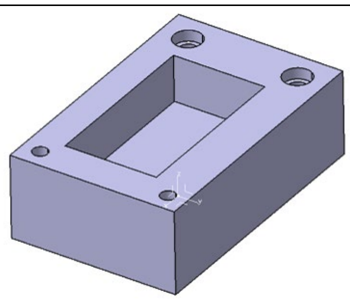
2.2. Prompting strategies

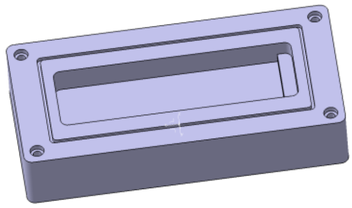
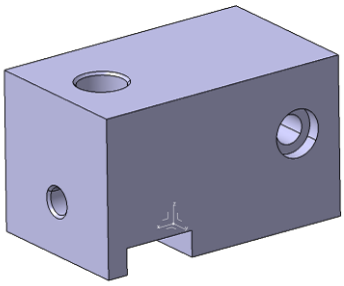
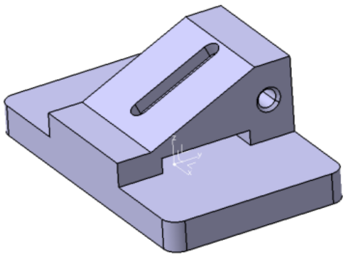
Two prompting strategies were evaluated. The full text of both prompts is presented in Annex 2.

The first prompt (Prompt 1) was a minimal prompt. It included the role of the model, input type and required task, i.e. to inspect the drawing, prepare a short modelling plan and generate a complete executable

Table 1

Set of parts used in testing

Code	Part type	Geometrical features	Difficulty	Tests / modeling challenges
P1	 Rectangular plate with mixed vertical holes	Rectangular prismatic plate; four vertical corner holes; one central vertical counterbored/stepped hole; rounded or chamfered external edges	Low	Reading global dimensions; positioning multiple vertical holes; distinguishing simple through holes from a central counterbored/stepped hole; recognizing basic edge rounding/chamfers
P2	 Prismatic part with through slot	Rectangular plate; central elongated through slot with rounded ends; rounded or chamfered external edges	Low	Recognizing an elongated slot; distinguishing a through slot from a blind recess; modelling a slot with straight sides and semicircular ends; preserving plate thickness and external dimensions
P3	 Prismatic part with two T channel of different extensions	Rectangular prismatic block; longitudinal stepped or T-shaped channel geometry; open groove features; rounded or chamfered external edges	Low	Recognizing the channel as a removed feature; distinguishing a stepped/T-shaped groove from a simple rectangular slot; interpreting the channel profile from front/section information; maintaining correct longitudinal continuity
P4	 Stepped prismatic support	Rectangular base; superposed rectangular upper block; four vertical fixing holes in the base; one central vertical hole in the upper block; chamfered or rounded edges	Medium	Creating successive extrusions; correctly placing the upper block on the base; positioning multiple vertical holes; distinguishing fixing holes from the central hole; separating global dimensions from local feature dimensions
P5	 Support with two vertical lugs and transverse hole	Rectangular base; two separate vertical lugs; open gap between lugs; one horizontal hole through each lug; chamfered or rounded lug edges	Medium	Modelling separate raised lugs on a common base; interpreting horizontal holes; maintaining hole alignment/coaxiality when dimensioned on a common axis; distinguishing an open gap between lugs from a machined slot; avoiding interpretation as a single U-shaped block unless supported by the drawing
P6	 Block with pocket and holes	Rectangular prismatic body; blind rectangular top pocket; four vertical fixing holes; two counterbored holes and two simple through holes	Medium	Creating a blind pocket; distinguishing through, blind and counterbored holes; assigning different hole depths and diameters; positioning holes relative to the outer body and the pocket; avoiding confusion between the external top contour and the internal cavity

P7	 Simplified prismatic housing / frame-like block	Rectangular housing-type body; large central pocket/cavity; peripheral rim; internal shoulder/stepped ledge; four counterbored fixing holes; chamfered or rounded outer edges	Complex	Creating nested pocket/cavity cuts; maintaining correct wall thicknesses; relating external dimensions, rim dimensions and cavity dimensions; modelling the internal shoulder/ledge; creating counterbored fixing holes; preserving solid validity after multiple Boolean cuts
P8	 Block with holes on different axes and bottom cut-out	Rectangular block; vertical top hole; horizontal/lateral holes on side faces; counterbore on one lateral face; bottom rectangular notch/cut-out	Complex	Interpreting hole-axis orientation; positioning holes from multiple views; creating horizontal cuts through side faces; recognizing a lateral counterbore; handling possible intersections or proximity between vertical and horizontal holes; modelling the bottom cut-out without confusing it with a through cavity
P9	 Support with inclined face, elongated slot and lateral hole	Rectangular base; raised wedge/inclined support; inclined top face; elongated slot on the inclined face; lateral/end horizontal hole with counterbore; side notch/groove; rounded or chamfered edges	Complex	Interpreting the oblique plane; constructing the wedge by profile extrusion or oblique cut; correctly placing and orienting the elongated slot on the inclined face; creating a horizontal hole on an end face; recognizing the counterbore; modelling the side notch/groove; avoiding orientation errors between vertical, horizontal and inclined features

FreeCAD Python script. This prompt did not include explicit rules for how to interpret drawings with multiple views and/or sections, nor asked for modeling order, uncertainties handling or how to classify the holes.

Prompt 2 was developed using the meta-prompting technique, being aimed to be more structured by including specific rules for using the views/sections, rules on identifying the main body and separating added and removed volumes and distinguishing between different feature types. It also explicitly asked for a preferred modelling order.

Although the current trend in the LLM-based applications is to use complex prompting strategies, it is important to mention that, for practical engineering uses, LLMs should provide the ability to obtain correct or at least partially useful outcomes, even with simple prompts. Hence the interest of testing prompts with different complexities.

2.3. Experimental protocol and evaluation criteria

Each part-prompt combination was tested in three independent runs, 54 runs in total (9 parts × 3 runs × 2 prompts). Each run was performed in a separate temporary chat to reduce the influence of prior conversations, memory or context carry-over between runs. For each run, the following aspects were checked:

- if the script executed successfully;
- if a valid solid was generated in FreeCAD;
- if the generated geometry was visually similar to the reference part;
- if the computed volume and surface area matched or differed from run to run and from the reference model.
- if the main dimensions matched the reference part.

Surface area and volume were obtained from the FreeCAD shape properties:

obj = Gui.Selection.getSelection()[0]
obj.Shape.Area
obj.Shape.Volume

For each model, the parts' dimensions, feature positions, hole types, hole-axis orientations, pocket depths, slot geometry and major added or removed volumes were compared with the reference model. Where needed, Boolean comparison operations were also conducted in CATIA V5 to identify the differences between the generated model and the reference model. These comparisons were especially useful when the visual appearance was similar but small local differences were present.

The following main criteria were used in the evaluation process:

- Script execution – for assessing if the generated FreeCAD Python script could be executed without runtime errors;
- Valid solid generation – for assessing whether the executed script produced a valid solid model;
- Structural correctness – for assessing if the generated model produced a part which topologically correspond with the reference;
- Dimensional correctness – for assessing if the generated model has the dimensions and feature positions

of the reference part: global dimensions, hole diameters, hole positions, pocket dimensions, slot dimensions, depths and offsets;

- Run-to-run stability – for assessing if the same prompt generated consistent results across the three independent runs for the same part. A prompt was considered stable for a given part when the three runs produced the same or equivalent geometry, with no relevant differences in topology, feature interpretation or main dimensions.

In addition to these criteria, volume and surface area were recorded to numerically check the geometric similarity. However, these values could not be used alone to determine correctness as small differences in chamfers, fillets, drill tips or counterbore interpretation can modify the surface area and volume without necessarily changing the main structural reconstruction.

The following reconstruction accuracy classes were used:

- E0 – script failed or no valid solid was generated,
- E1 – valid solid, but with incorrect topology,
- E2 – main topology correct, but one or more features are missing or misplaced,
- E3 – fully correct reconstruction, including dimensions, feature positions and local details.

At part level, E3 was assigned when all the independent runs produced fully correct reconstructions. If there were differences between the generated models in different runs or if at least one relevant feature was missing or incorrectly interpreted, the part was assigned a lower class.

3. RESULTS AND DISCUSSION

The results presented further are divided per prompting strategies. First, in section 3.1, the outputs of the simple prompt are discussed part by part, including details on modeling plan generated by the LLM, images of reconstructed FreeCAD model and comparison with the reference part in terms of volume and surface area. The purpose is to document the baseline behavior of the LLM when only minimal instructions are provided. Second, the results obtained with the moderately structured prompt are presented synthetically in Section 3.2, using comparative tables focused on whether the additional interpretation rules improved structural correctness, dimensional correctness and run-to-run stability.

3.1. Results of the simple prompt – Prompt 1

3.1.1. Part P1: modeling plan, 3D model and evaluation

ChatGPT answer: "

- Build the base plate as a rectangular solid: **100 × 50 × 35 mm**.
- Use the top view as the XY layout and extrude height along Z.
- Drill **4 × Ø8 through-holes** at the corner positions:
  - X positions: **9 mm** and **91 mm**
  - Y positions: **41 mm** and **7 mm**
- Drill the central **Ø8 through-hole** at the plate center: **(50, 25)**.
- Add the central top counterbore:
  - Diameter: **Ø15**
  - Depth: **5 mm**
  - Concentric with the Ø8 central through-hole."

All runs produced similar parts and identical with the P1 reference.

In Figure 1 are compared P1 volume and area for the reference part and FreeCAD part, these being identical, while figure 2 shows the P1 in FreeCAD environment.

To illustrate how the reconstruction workflow conducted by ChatGPT operates, this simple part, P1 (see Annex 1), was used as an example for showing how the

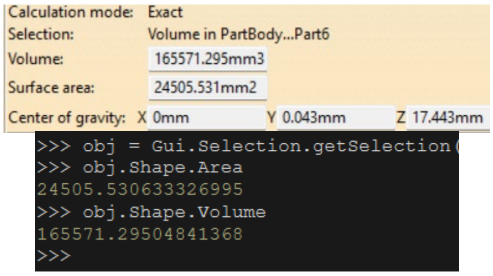


Fig. 1. Volume and area comparison for P1.

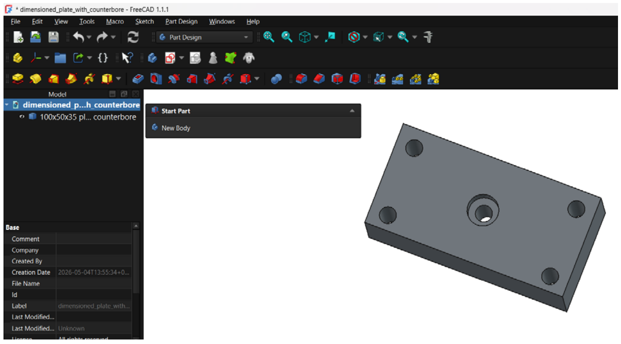


Fig. 2. 3D model of P1 in FreeCAD.

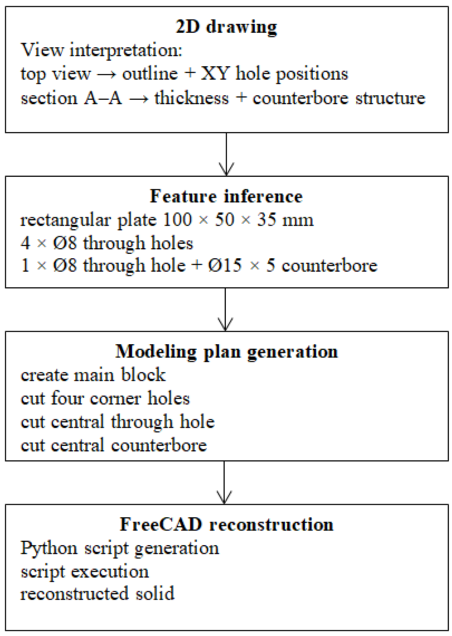


Fig. 3. Example of interpretation workflow for part P1.

modeling plane is derived from the 2D images of the part. The example illustrates the intermediate stage between 2D drawing interpretation and FreeCAD Python script generation (Fig.3).

The top view was first used to identify the outer rectangular contour and the XY positions of the holes, while section A-A was used to determine the part thickness and the internal structure of the central hole. Based on these two views, the part was interpreted as a rectangular plate of 100 × 50 × 35 mm containing four Ø8 through holes near the corners and one central Ø8 through hole with a Ø15 counterbore of 5 mm depth. This interpretation was then converted into a short modeling plan consisting of four main operations: creation of the main block, cutting of the four corner holes, cutting of the central through hole, and creation of the concentric counterbore.

3.1.2. Part P2: modeling plan, 3D model and evaluation

ChatGPT answer: "

- Model the base as a **130 × 50 × 15 mm** plate.
- Round the **4 outer plan-view corners** with **R4**.
- Cut a central through-slot:
  - Slot end radius: **R8**, so slot width is **16 mm**.
  - Straight slot length between tangent points: **60 mm**.
  - Right tangent point is **35 mm** from the right edge, so the slot centers are at **X = 35 mm** and **X = 95 mm**.
  - Slot centerline is at **Y = 25 mm**.
- Apply a **R4 fillet to the upper edge of the slot**, as shown in section A-A. "

The generated parts were geometrically identical in all runs and matched the reference, for P2 as well. Figures 4 and 5 presents the results from P2 evaluation.

3.1.3. Part P3: modeling plan, 3D model and evaluation

ChatGPT answer: "

- Create the base block: **70 × 160 × 20 mm**.

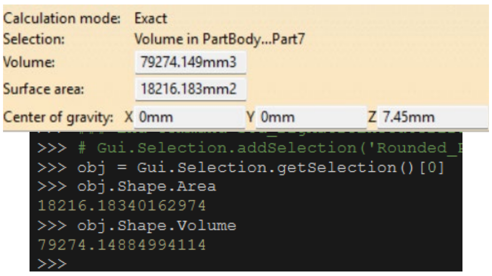


Fig. 4. Volume and area comparison for P2.

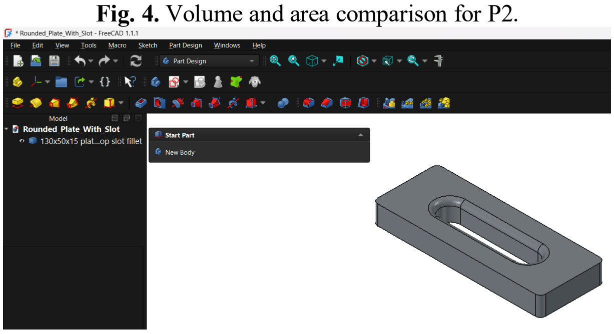


Fig. 5. 3D model of P2 in FreeCAD.

- Cut two longitudinal **T-slots** using the front-view profile:
  - Upper slot opening width: **7 mm**
  - Undercut width: **19 mm**
  - Side undercut: **6 mm each side**
  - Bottom remaining wall: **5 mm**
  - Undercut height assumed as **5 mm**, so the upper narrow slot runs from **Z = 10 to Z = 15**.
- Left T-slot runs the full **160 mm** length.
- Right T-slot runs **100 mm** from the lower end.
- The **10 mm** dimensions in the top view are interpreted as offsets from the side edge to the outer edge of the upper 7 mm slot opening."

All runs for P3 were identical with the reference based on volume, area and dimensions checking (Figs. 6 and 7).

3.1.4. Part P4: modeling plan, 3D model and evaluation.

ChatGPT answer: "Use millimeters. Create a base plate **120 × 80 × 20** with **R5 rounded corners**. Add a centered raised boss on top, **60 × 60 × 40**, with **4 mm chamfers on all four corners**. Cut four through mounting holes **Ø10** at **100 × 60** center spacing, and cut the centered **Ø20** through-hole through the whole part."

In all three runs, identical P4 parts were generated, and they were identical with the reference part (Figs. 8 and 9). Only the modeling plan differed between runs as

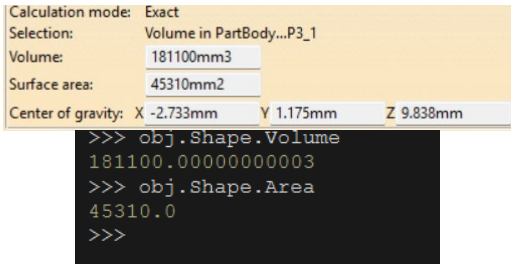


Fig. 6. Volume and area comparison for P3.

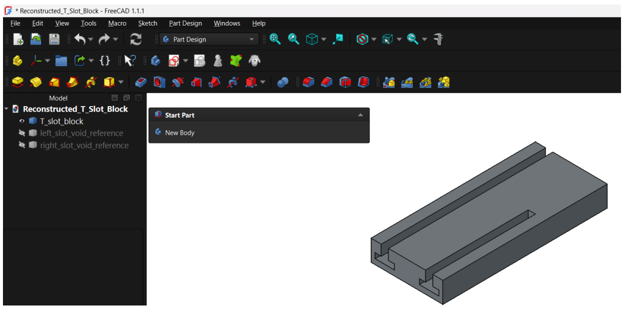


Fig. 7. 3D model of P3 in FreeCAD.

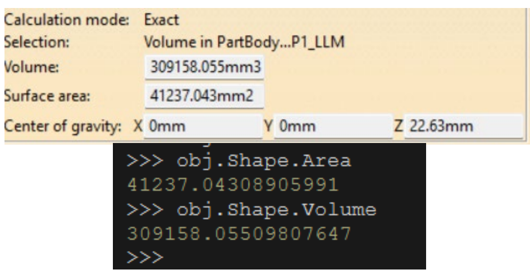


Fig. 8. Volume and area comparison for P4.

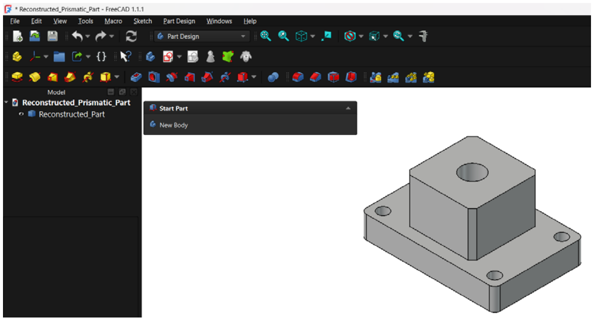


Fig. 9. 3D model of P4 in FreeCAD.

no constraints regarding its format were included in the prompt. As an example, in run 2 of the simple prompt, the following response was generated, which can be compared with another answer presented at the beginning:

"Interpreted geometry:

- Base plate: **120 × 80 × 20 mm**
- Outer plate corner fillets: **R5**
- Upper raised boss: **60 × 60 × 40 mm**, centered on the base
- Upper boss plan shape: square with **4 × 4 mm chamfers** on the four corners
- Four vertical corner holes: **Ø10**, through the 20 mm base plate
- Hole centre spacing: **100 mm × 60 mm**, therefore centres at 10 mm from each outside edge
- Central vertical hole: **Ø20**, through the complete part
- Total height: **60 mm**"

3.1.5. Part P5: modeling plan, 3D model and evaluation. The modeling plan generated by the LLM was the following:"

- Units: **mm**.
- Base plate: **160 × 80 × 15**, with **R5 rounded plan corners**.
- Left upright: **15 × 40 footprint**, located **25 mm from left edge**, overall height **80**, top chamfered **4 × 45°**.
- Right upright: **20 × 50 footprint**, located with its left face **44 mm from the right edge**, overall height **60**, top chamfered **4 × 45°**.
- Both upright holes: **Ø10**, through the upright thickness along the **X direction**, centered at **Z = 40** and **Y = 0**."

In all three runs executed individually in temporary chat using the same Prompt 1, the parts were identical and they had the same volume, total area and dimensions with the reference part (Figs. 10 and 11).

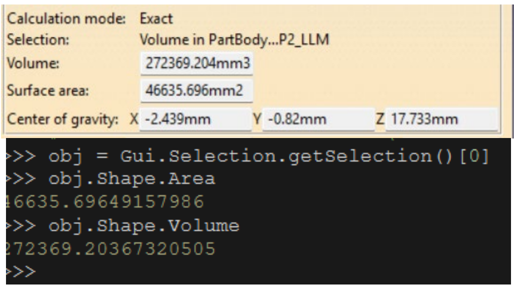


Fig. 10. Volume and area comparison for P5.

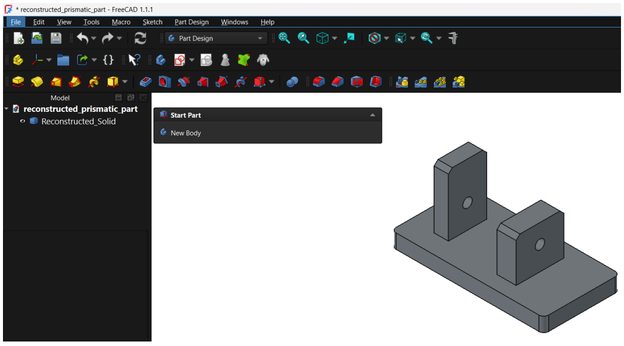


Fig. 11. 3D model of P5 in FreeCAD.

3.1.6. Part P6: modeling plan, 3D model and evaluation. ChatGPT answer: "Assumptions used from the drawing:

- Main block: **140 × 90 × 45 mm**.
- Central rectangular blind pocket: **90 × 50 mm**, depth **25 mm** from the top face.
- Pocket position: **30 mm** from the left edge, **20 mm** from the right edge, centered in width with **20 mm** margins.
- Two left holes: vertical **Ø10 through-holes** with **Ø15 × 6 mm counterbores** from the top face.
- Two right holes: vertical **Ø8 through-holes**.
- Hole positions:
    - Left pair: x = 15, y = 15 and y = 75.
    - Right pair: x = 130, y = 10 and y = 80."

For P6, the generated models were visually close to the reference (Fig. 12) and reproduced the main body, the top pocket, and the hole types. However, differences were recorded between runs in volume and surface area values (Fig. 13), and comparison with the reference showed that the simple-hole positions were not correctly reconstructed. This error is important because volume and surface area are not sufficient indicators of correctness: holes with the same diameter and depth may remove the same amount of material even when placed in the wrong position. Therefore, feature-position checking was necessary in addition to volume and surface-area comparison.

3.1.7. Part P7: modeling plan, 3D model and evaluation. ChatGPT answer: "I treated the uploaded drawing as a prismatic flange/frame part in millimetres. The main interpreted dimensions are: overall body

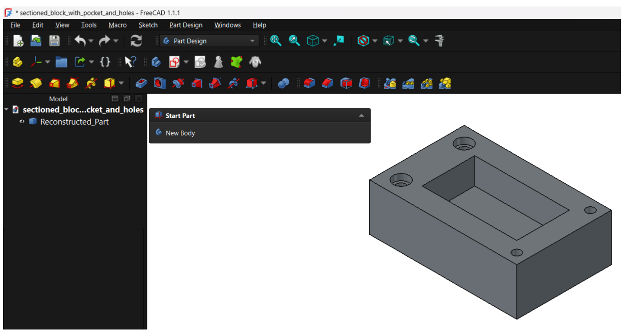


Fig. 12. 3D model of P6 in FreeCAD.

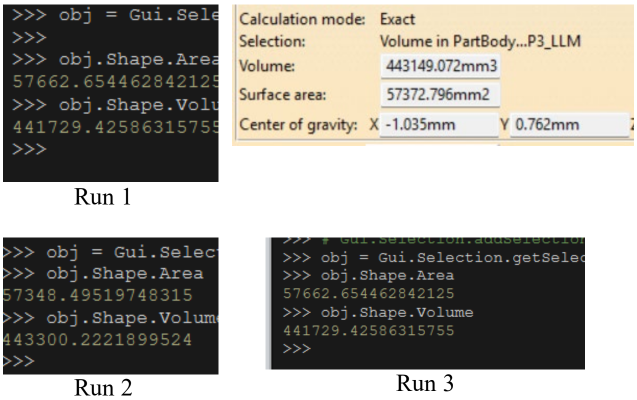


Fig. 13. Volume and area comparison for P6 in different runs and in comparison to reference.

230 × 110 × 20, outer corner radius R6, four mounting holes on a 210 × 90 bolt pattern, Ø6 through holes with Ø10 counterbores, a central rounded rectangular through-window 170 × 50, R5, and a shallow top rectangular groove/rabbet around it using the shown 187.2 × 79.4, 2.2 offset, R3, and 3 mm depth features. The drawing is a single-page orthographic sketch, so the exact internal step interpretation is mildly ambiguous; the script below uses a standard manufacturable interpretation of the visible and hidden-line geometry."

Figure 14 presents the results obtained by running the generated scripts for P7. One can note the differences between the three runs, which showed LLM run-to-run instability in interpreting the same drawing.

In run 1, the central cavity was considered a through pocket. In run 2, the part's height was smaller than in the reference model and no stair-like interior feature was modelled, whereas in run 3, the external dimensions and the main pocket were dimensionally similar to the reference, but the internal stair-like feature was omitted. Therefore, even if all scripts executed successfully, Prompt 1 did not produce a stable or structurally correct reconstruction for this part. The errors were not related to the script execution, but to the unstable interpretation of the pocket and internal shoulder.

3.1.8. Part P8: modeling plan, 3D model and evaluation. ChatGPT answer: "I interpreted the page-1 orthographic views as a 100 × 60 × 60 mm block with a bottom rectangular notch, a vertical Ø18 through-bore on the left, a Ø10 blind drilled side hole with 120° drill point, and a front Ø18/Ø11 stepped blind hole with 120° drill point.

Modeling plan:

1. Create the main rectangular solid: 100 mm length × 60 mm depth × 60 mm height.
2. Cut the bottom rectangular notch: 28 mm wide, 10 mm high, starting 8 mm from the left face, through the full depth.
3. Cut the left horizontal Ø10 blind drilled hole, centered at y = 30 mm, z = 20 mm, with a 120° conical drill tip and a small 1 × 1 chamfer at the entrance.
4. Cut the right/front stepped blind hole: center x = 82 mm, z = 35 mm, outer Ø18 counterbore depth 5 mm, inner Ø11 drill depth 40 mm, with a 120° drill tip. "

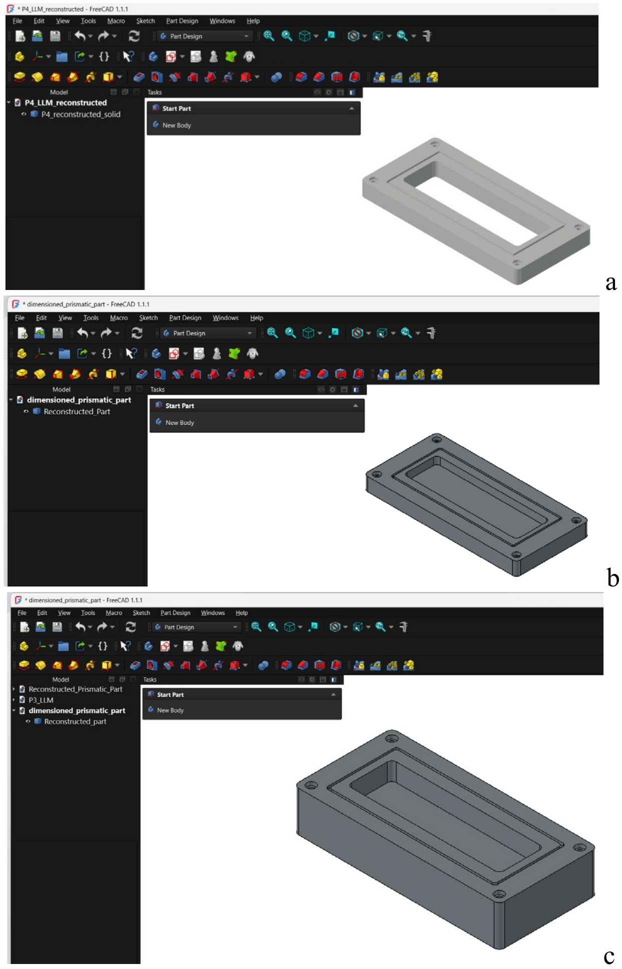


Fig. 14. P7 generated in three independent runs: *a* – run 1; *b* – run 2; *c* – run 3.

Figure 15 shows the part generated in run 1, as an example. For all runs, generated parts were measured, and their volume and surface area were compared (Fig. 16). Small deviations were observed, caused by the missing chamfer at the top through hole.

For P8, the main topology was correctly reconstructed with the exception of a local feature. The generated model included the main prismatic body, the vertical top hole, the lateral hole and the bottom notch. The observed differences in volume and surface area were caused by a missing chamfer at the top through hole. The case shows that small edge features, such as chamfers, are less reliably reconstructed than the main volumetric features.

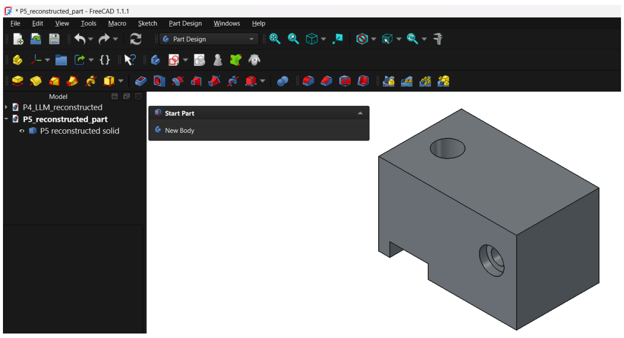


Fig. 15. 3D model of P8 generated in FreeCAD.

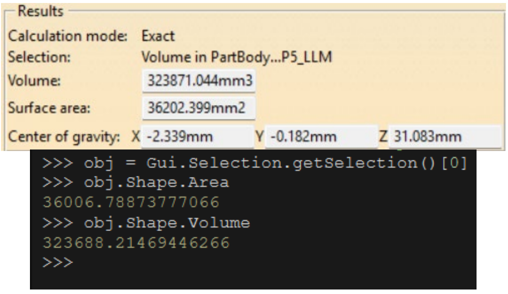


Fig. 16. Volume and area comparison for P8 in different runs and in comparison to reference.

3.1.9. Part P9: modeling plan, 3D model and evaluation. ChatGPT answer: "From the page-1 drawing, I modeled the part as a 100 × 60 mm rounded rectangular base, with a central 40 mm wide raised ramp block, a Ø6 through-hole, and an obround slot shown in the top/front/auxiliary views. Some slot placement is not fully dimensioned, so I used the visible 27/30 and 8 mm indications from the auxiliary/section views as the governing values.

Modeling plan:

Use millimeters.

1. Create the base plate: 100 × 60 × 10 mm, with R5 rounded plan corners.
2. Add the central raised wedge: X = 30 to 70 mm, full 60 mm width, high side 35 mm, low side 18 mm, with a 155° included ramp angle.
3. Cut a Ø6 through-hole along X through the raised block, centered at Y = 8 mm and Z = 23 mm.
4. Cut an obround slot through X in the raised block, approximately 27 × 8 mm, centered on the part.
5. Fuse/cut all features into one checked FreeCAD solid."

P9 was the most difficult part in the set (Fig. 17). Although all scripts executed successfully, none of the generated models reproduced the reference topology. The failures were not identical across runs.

In one run, the inclined support and the slot were interpreted as an incorrect open cut through the raised body. In another run, the raised wedge and the lateral hole were partially reconstructed, but the slot on the inclined face and the local side features were missing or incorrectly placed. The third run was visually closer to the reference, but the orientation and integration of the slot, the lateral hole and the inclined support were still incorrect. This indicates that the LLM had difficulty combining information from multiple views when the part included inclined geometry and features not aligned with the main Cartesian axes.

After the parts were generated, they were analyzed based on various criteria.

Table 2 summarizes the evaluation of the models generated with Prompt 1. All 27 scripts executed successfully and produced valid FreeCAD solids. Thus, reported differences refer to geometric correctness and run-to-run stability, not to execution failure.

3.2. Results of the structured prompt – Prompt 2

The same approach of running three independently generated codes and analyzing the results, i.e. the FreeCAD 3D solids, was applied for Prompt 2.

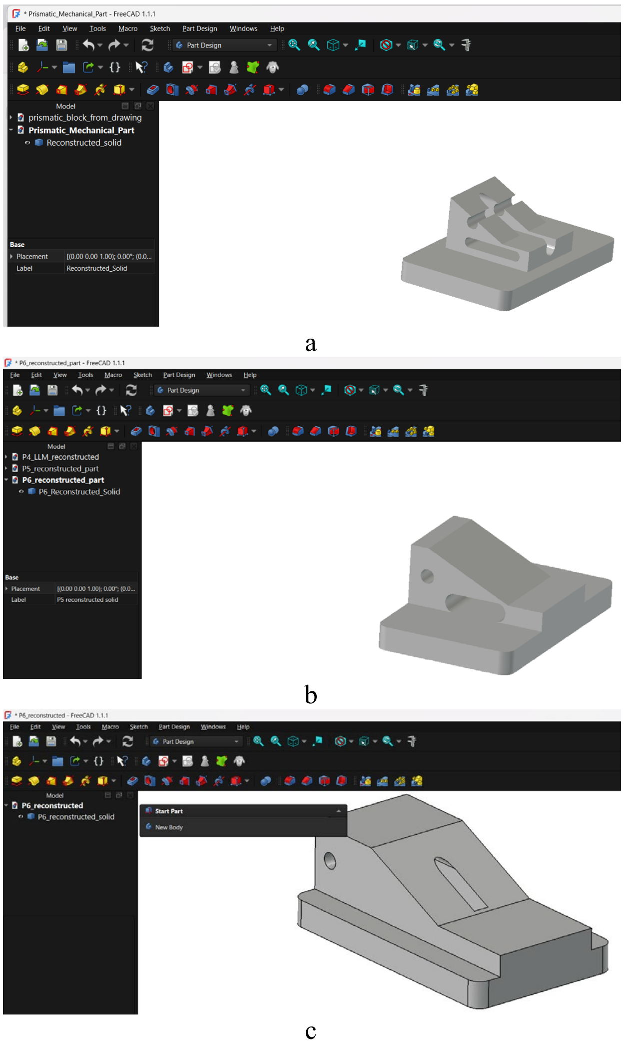


Fig. 17. P9 part generated in the three runs by means of the simple prompt: *a* – run 1; *b* – run 2; *c* – run 3.

Table 3 presents the results with this structured prompt. As in the case of Prompt 1, all Python scripts generated by ChatGPT executed successfully and produced valid FreeCAD solids.

Parts P1-P5 were correctly reconstructed in all three runs, also matching the corresponding reference models.

For P6, all the generated scripts were also executed successfully, producing valid solid parts. However, in one of the runs, a small numerical difference, at the third decimal place, was recorded between the reference and reported surface area and volume values.

A Boolean check was conducted in CATIA V5R21, which showed that the generated parts were geometrically identical to the reference model. The possible explanation was that there was no modelling error or any missing feature, the difference being determined by the internal B-rep representation or a display rounding in FreeCAD. Although in other research the evaluation of results is conducted automatically, some cases, like this one, show that the best evaluation method is manual, because it is possible to identify and analyze complex errors.

For P7, the internal step-like feature was again missed in all three runs, despite improved prompt aimed at specifically analyzing each view. The rest of the dimensions matched the reference model, which

Table 2

Evaluation of parts generated with Prompt 1

Part	Volume / surface area compared with reference	Dimensional correctness	Structural correctness	Run-to-run stability	E-level	Final evaluation	Main observations
P1	Matching	Correct	Correct	Stable	E3	Correct	All three runs produced identical models and matched the reference geometry.
P2	Matching	Correct	Correct	Stable	E3	Correct	All three runs produced identical models and matched the reference geometry.
P3	Matching	Correct	Correct	Stable	E3	Correct	All three runs produced identical models and matched the reference geometry.
P4	Matching	Correct	Correct	Stable	E3	Correct	All three runs generated identical models matching the reference. Only the wording of the modelling plan differed between runs.
P5	Matching	Correct	Correct	Stable	E3	Correct	All three runs generated models with the same volume, surface area and main dimensions as the reference model.
P6	Different	Partially correct	Partially correct	Unstable	E2	Partially correct	The main body, top pocket and hole types were visually close to the reference, but the simple-hole positions were not correctly reconstructed. Volume and surface area also differed between runs.
P7	Not meaningful after structural mismatch	Incorrect	Incorrect	Unstable	E1	Incorrect	All scripts executed and generated valid solids, but the models were structurally incorrect. The internal stepped feature inside the pocket was missing, and the interpretation of the pocket differed between runs.
P8	Slightly different	Partially correct	Structurally acceptable	Stable	E2	Partially correct	The main body and main features were correctly reconstructed. The remaining deviation was caused by a missing chamfer at the top through hole.
P9	Not meaningful after structural mismatch	Incorrect	Incorrect	Unstable	E1	Incorrect	All scripts executed, but none of the generated models reproduced the reference topology. The main errors concerned the hole, the through pocket/slot and the inclined-feature interpretation.

Table 3

Evaluation of parts generated with moderately structured Prompt 2

Part	Volume / surface area compared with reference	Dimensional correctness	Structural correctness	Run-to-run stability	E-level	Final evaluation	Main observations
P1	Matching	Correct	Correct	Stable	E3	Correct	All three runs produced correct models matching the reference model.
P2	Matching	Correct	Correct	Stable	E3	Correct	All three runs produced correct models matching the reference model.
P3	Matching	Correct	Correct	Stable	E3	Correct	All three runs produced correct models matching the reference model.
P4	Matching	Correct	Correct	Stable	E3	Correct	All three runs produced correct models matching the reference model.

P5	Matching	Correct	Correct	Stable	E3	Correct	All three runs produced correct models matching the reference model.
P6	Matching after Boolean comparison	Correct	Correct	Stable	E3	Correct	A small difference in reported surface area and volume was observed at the third decimal place in one run. Boolean comparison showed that the model was geometrically identical to the reference.
P7	Different / incomplete	Partially correct	Partially correct	Stable but incomplete	E2	Partially correct	The main dimensions matched the reference model, but the internal step-like feature was omitted in all runs.
P8	Slightly different	Correct for main dimensions	Structurally acceptable with minor local omission	Stable but incomplete	E2	Partially correct	The main geometry and features were reconstructed correctly, but the chamfer at the top through hole was omitted.
P9	Not meaningful after structural mismatch	Incorrect	Incorrect	Unstable / incorrect in all runs	E1	Incorrect	All scripts executed, but none of the generated models reproduced the reference topology. The main errors concerned the inclined feature, the hole and the through pocket/slot.

Table 4

Comparison between the results for Prompt 1 and Prompt 2

Criteria	Prompt 1 result	Prompt 2 result
Script execution success	27/27 = 100%	27/27 = 100%
Valid solid generation	27/27 = 100%	27/27 = 100%
E3 — fully correct parts	5/9 = 55.6%	6/9 = 66.7%
E2 — partially correct parts	2/9 = 22.2%	2/9 = 22.2%
E1 — valid solids but structurally incorrect	2/9 = 22.2%	1/9 = 11.1%
E0 — execution failure / invalid solid	0/9 = 0%	0/9 = 0%
Stable fully correct parts	5/9 = 55.6%	6/9 = 66.7%

Table 5

Fully correct reconstructions by difficulty level

Difficulty level	Parts	Prompt 1 fully correct	Prompt 2 fully correct
Low	P1–P3	3/3	3/3
Medium	P4–P6	2/3	3/3
Complex	P7–P9	0/3	0/3
Total	**P1–P9**	**5/9**	**6/9**

indicates a localized, but structurally relevant error. For P8, the main geometry was correctly reconstructed, but in all runs ChatGPT missed the chamfer of the top hole. So, the result was omitting the chamfer in a stable manner.

P9 was geometrically incorrect in all runs.

Thus, Prompt 2 achieved 6/9 fully correct parts, one additional part with only a minor local-detail omission, one partially correct part and one incorrect part.

The final results are gathered in Tables 4 and 5. Table 4 summarizes the general performance of the two analyzed prompting strategies. Both prompts achieved 100% script execution success and 100% valid solid generation. This result shows that, for the tested cases, FreeCAD Python code generation was not the limiting stage. The difference, even a very small one, between the prompts appeared at the level of geometrical correctness.

The number of fully correct parts increased from 5/9 for the minimal prompt to 6/9 for the moderately structured prompt. The same values were obtained when analyzing the number of stable fully correct parts, this showing a consistency across the three independent runs.

Table 5 shows that reconstruction success was strongly dependent on part complexity. The low-complexity parts were correctly reconstructed regardless of the prompting strategy. This confirms that the LLM can reliably reconstruct simple and clearly dimensioned prismatic parts when the relevant features are aligned with the main Cartesian axes and are directly represented in the drawing views.

For the medium complexity parts, Prompt 1 produced two out of three fully correct reconstructions, while Prompt 2 produced three out of three fully correct reconstructions. This shows that the additional interpretation rules were useful for medium complexity geometry. In particular, P6 was correctly reconstructed with Prompt 2, while the use of Prompt 1 produced an unstable reconstruction with incorrect hole positions.

For the complex parts, neither prompt produced a fully correct reconstruction. Prompt 2 improved the result for P7 by producing a stable, but still incomplete model, the internal step-like feature being omitted. P8 was just partially correct because of a missing chamfer, while P9 was incorrect for both prompting strategies. These results indicate that prompt structuring improved reconstruction accuracy only up to a certain complexity level. It did not solve the interpretation of complex features requiring the integration of several views, internal steps, inclined geometry, local details or non-axis-aligned features.

The same results distribution as in Table 5 was obtained for stable fully correct reconstructions, because all fully correct parts were reproduced consistently in the three runs.

4. CONCLUSIONS

The results of this pilot research showed that ChatGPT Thinking Model 5.5 can reliably generate executable FreeCAD scripts for simple to medium-complex prismatic parts based on their 2D drawings. However, currently, it cannot reliably and autonomously reconstruct 3D parts from complex mechanical drawings.

The study also analyzed the influence of the prompting technique over the performance, the outcomes showing that the moderately structured prompt did not perform significantly better than the basic prompt. It increased from 5/9 to 6/9 the number of fully correct reconstruction just for the medium-complexity group, not solving the reconstruction of complex parts. This suggests that prompt can help the LLM to organize the interpretation of clear or moderately complex drawings, but it cannot compensate for incorrect inference of complex 3D feature structures.

Another important conclusion of the research is that, in LLM-based reconstruction of 3D parts from 2D drawings, the main difficulty is not the generation of executable FreeCAD Python code, but the interpretation of the 2D drawing so that to produce a modeling plan and a correct 3D feature structure. In the analyzed parts, most errors appeared in local details, such as chamfers, internal steps and slots, or in features involving inclined geometry. These results indicate that LLM-based 2D-to-3D reconstruction can be useful as an assisted modelling approach, but it is not currently reliable as a fully autonomous CAD reconstruction method for complex mechanical parts.

The study has the limitation of a small dataset as it included only nine prismatic mechanical parts. The results were obtained with one multimodal LLM, two prompting strategies and one CAD scripting environment. Therefore, the results should be considered exploratory and mainly applicable to LLM-assisted reconstruction of relatively simple prismatic parts. They should not be generalized to all types of technical drawings, all CAD systems or all 2D→3D reconstruction tasks.

REFERENCES

[1] Brown, D.C. *Artificial intelligence for design process improvement.* In: Clarkson, J., Eckert, C. (eds) Design process improvement. 2005. Springer, London.

[2] Chen, H., Gao, C., Wu, J., Ren, K., Zhang, W. *Study on optimization design of airfoil transonic buffet with reinforcement learning method*. Aerospace, 10(5), 2023, 486.

[3] Eser, A., Aşkar Ayyıldız, E., Ayyıldız, M., Kara, F. *Artificial intelligence-based surface roughness estimation modelling for milling of AA6061 alloy.* Adv. Mater. Sci. Eng., 2021(1), 2021, 5576600.

[4] Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., Quillen, D. *Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection.* Int. J. Robot Res., 37(4–5), 2018, 421–436.

[5] Puttegowda, M., Nagaraju, S.B. *Artificial intelligence and machine learning in mechanical engineering: Current trends and future prospects.* Engineering Applications of Artificial Intelligence, 142, 2025, 109910.

[6] Li, Z. *Application of artificial intelligence technology in decision-making of mechanical manufacturing process.* Journal of Physics: Conference Series, 2143(1), 2021, 012008.

[7] Jenis, J., Ondriga, J., Hrcek, S., Brumercik, F., Cuchor, M., Sadovsky, E. *Engineering applications of artificial intelligence in mechanical design and optimization.* Machines, 11(6), 2023, 577.

[8] Ciccone, F., Bacciaglia, A., Ceruti, A. *Optimization with artificial intelligence in additive manufacturing: A systematic review.* Journal of the Brazilian Society of Mechanical Sciences and Engineering, 45(6), 2023, 303.

[9] Peckham, O., Raines, J., Bulsink, E., Goudswaard, M., Gopsill, J., Barton, D., Nassehi, A., Hicks, B. *Artificial intelligence in generative design: A structured review of trends and opportunities in techniques and applications.* Designs, 9, 2025, 79.

[10] Kretzschmar, M., Dammann, M.P., Schwoch, S., Braun, F., Saske, B., Paetzold-Byhain, K. *Evaluating the Role of Generative AI in Product Development and Design - A Systematic Review.* In: NordDesign 2024, Reykjavik, Iceland, 12–14 August 2024.

[11] Khanolkar, P.M., Vrolijk, A., Olechowski, A. A case study of the decision-making behind the automation of a composites-based design process. Proceedings of the Design Society, 3, 2023, pp. 49–58.

[12] Rix, J., Haas, S., Teixeira, J. (eds) *Virtual Prototyping: Virtual environments and the product design process.* Springer, New York, 2016

[13] Ortner-Pichler, A., Landschutzer, C. *Integration of parametric modelling in web-based knowledge-based engineering applications*. Advanced Engineering Informatics, 51, 2022, 101492.

[14] Khan, M.S., Sinha, S., Sheikh, T.U., Stricker, D., Ali, S.A., Afzal, M.Z. *Text2CAD: Generating sequential CAD models from beginner-to-expert level text prompts*. 2024, *arXiv*. https://doi.org/10.48550/arXiv.2409.17106

[15] Li, J., Ma, W., Li, X., Lou, Y., Zhou, G., Zhou, X. *CAD-Llama: Leveraging large language models for computer-aided design parametric 3D model generation.* 2025, *arXiv*. https://doi.org/10.48550/arXiv.2505.04481

[16] Guan Y, Wang X, Xing X, Zhang J, Xu D, Yu Q. *CAD-Coder: text-to-CAD generation with chain-of-thought and geometric reward*. 2025. arXiv. https://doi.org/10.48550/arXiv.2505.19713

[17] Wang, X., Zheng, J., Hu, Y., Zhu, H., Yu, Q., Zhou, Z. *From 2D CAD drawings to 3D parametric models: A vision-language approach*. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2025, 39(8), pp. 7961–7969

[18] Qin, F., Lu, S., Hou, J., Wang, C., Fang, M., Liu, L. *Drawing2CAD: Sequence-to-sequence learning for CAD generation from vectorized drawings*. 2025 *arXiv*, https://doi.org/10.48550/arXiv.2508.18733

[19] Wu, S., Khasahmadi, A., Katz, M., Jayaraman, P.K., Pu, Y., Willis, K., Liu, B. *CAD-LLM: Large language model for CAD generation*. In: NeurIPS Conference, San Diego, California, USA, 2023

[20] Wu, S., Khasahmadi, A., Katz, M., Jayaraman, P.K., Pu, Y., Willis, K., Liu, B. *CadVLM: Bridging language and vision in the generation of parametric CAD sketches*. 2024, *arXiv*. https://doi.org/10.48550/arXiv.2409.17457

[21] Adam. Adam - AI Powered CAD. Available online: https://www.adam.new/ (accessed on 7 May 2026).

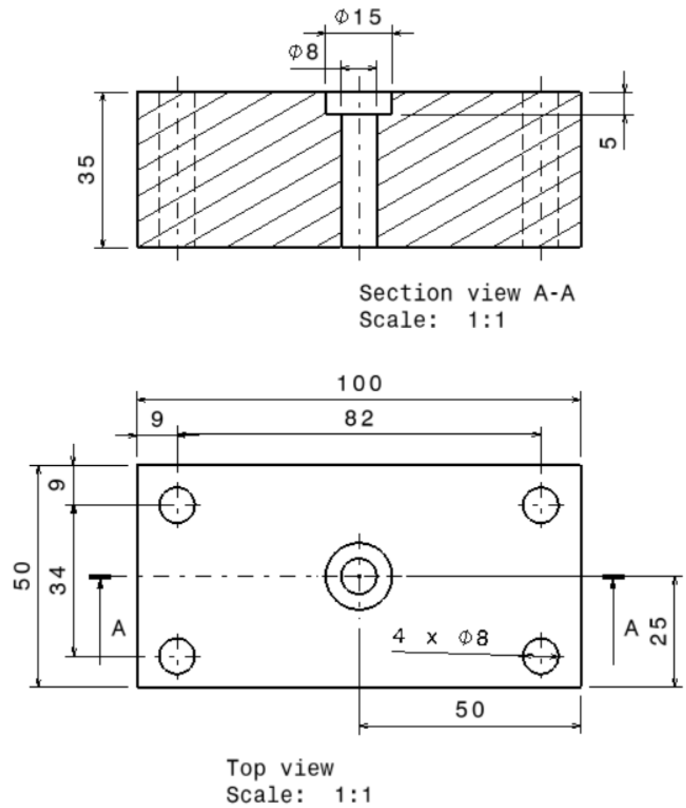
[22] Camfer. Camfer - The AI CAD Tool. Available online: https://camfer.dev/ (accessed on 7 May 2026).

[23] MecAgent Inc. MecAgent - AI CAD Copilot. Available online: https://mecagent.com/ (accessed on 7 May 2026).

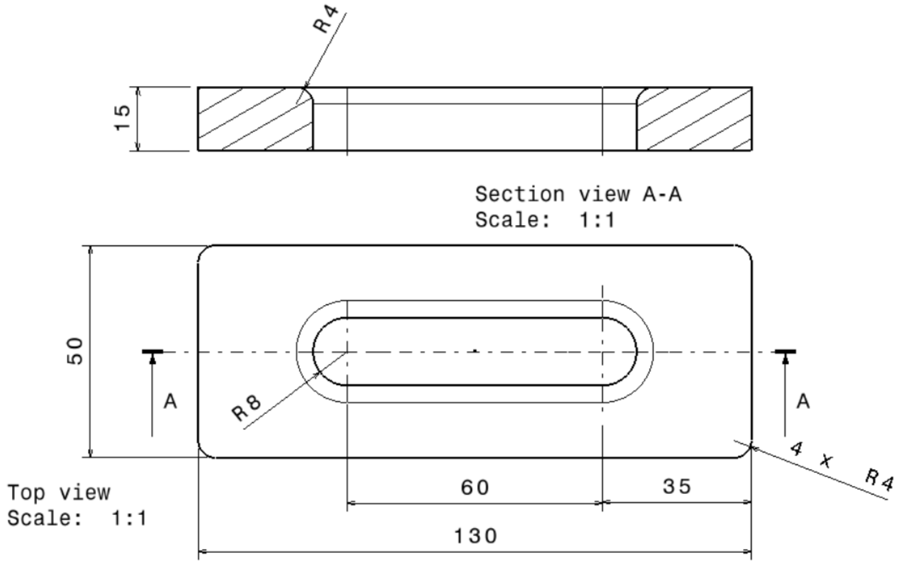
[24] Autodesk. *Bringing Fusion onto Claude for Creative Work*. Autodesk Platform Services Blog, 28 April 2026. Available online: https://aps.autodesk.com/blog/bringing-fusion-claude-creative-work (accessed on 11 May 2026).

ANNEX 1 – 2D drawings of P1-P9 parts included in the dataset

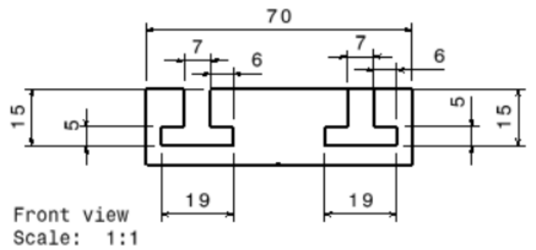
P1

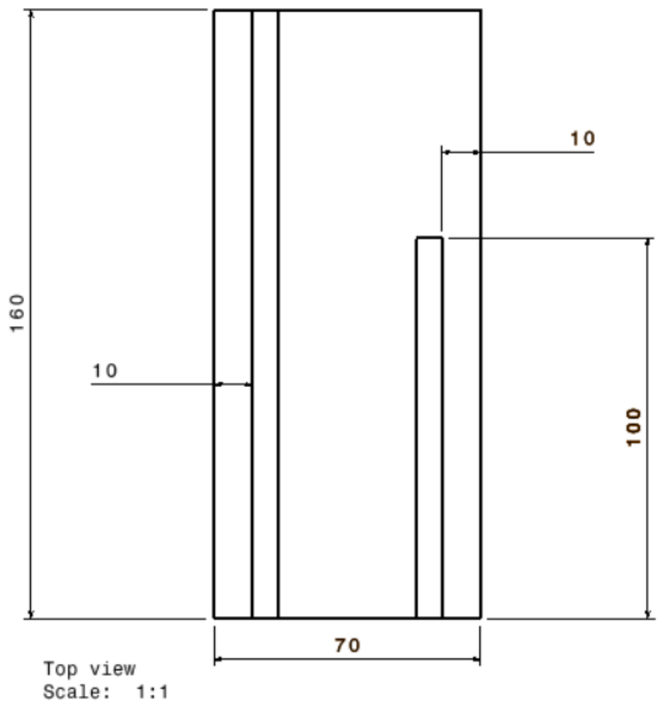


P2

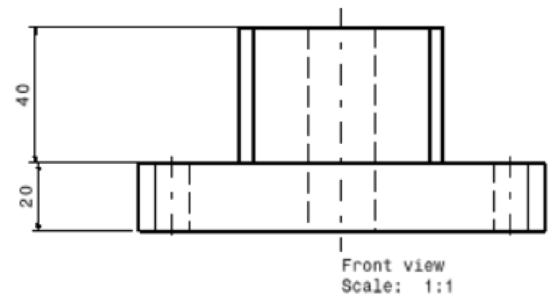


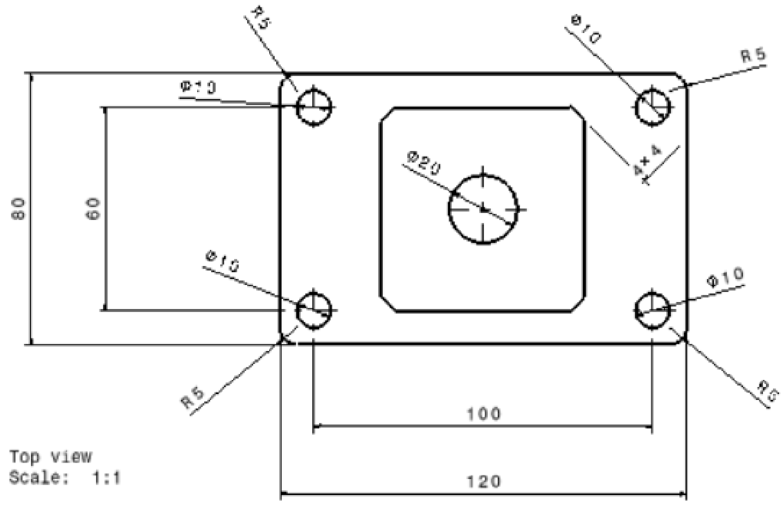
P3


70
7
6
7
6
15
5
5
15
19
19
Front view
Scale: 1:1

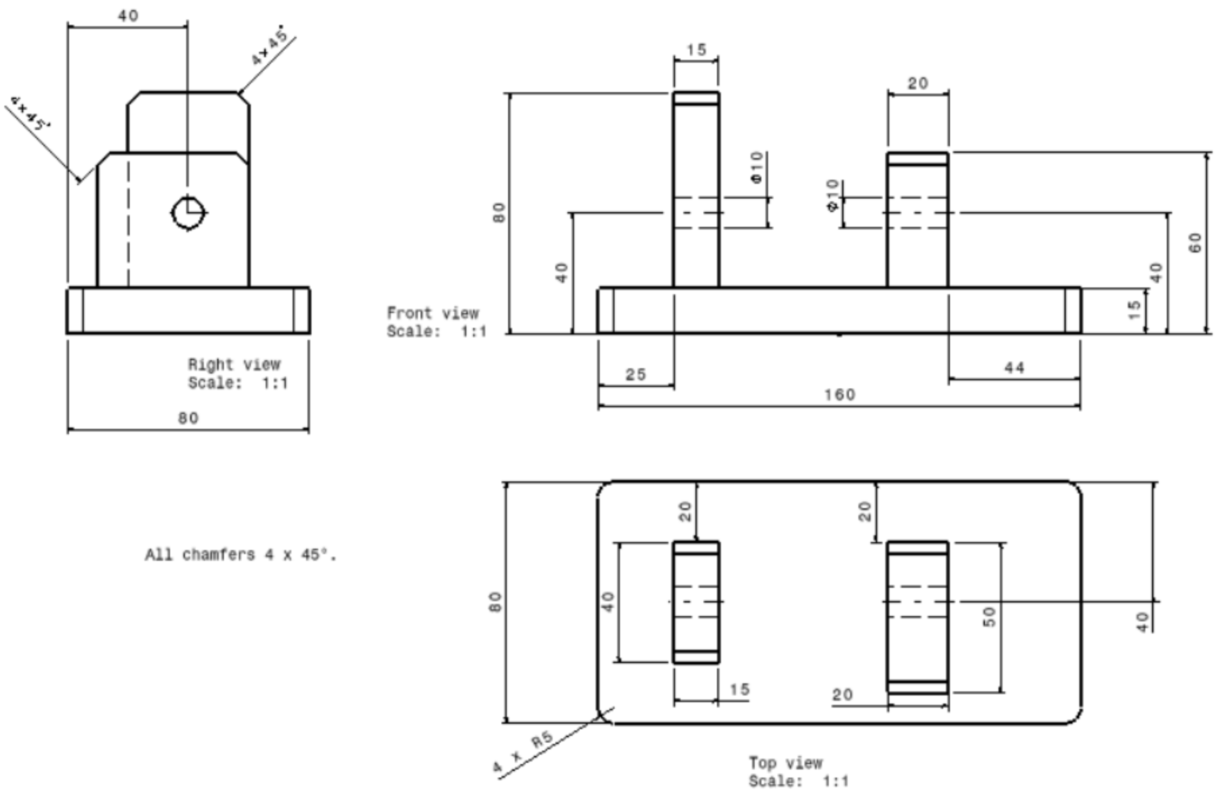

160
10
10
100
70
Top view
Scale: 1:1

P4

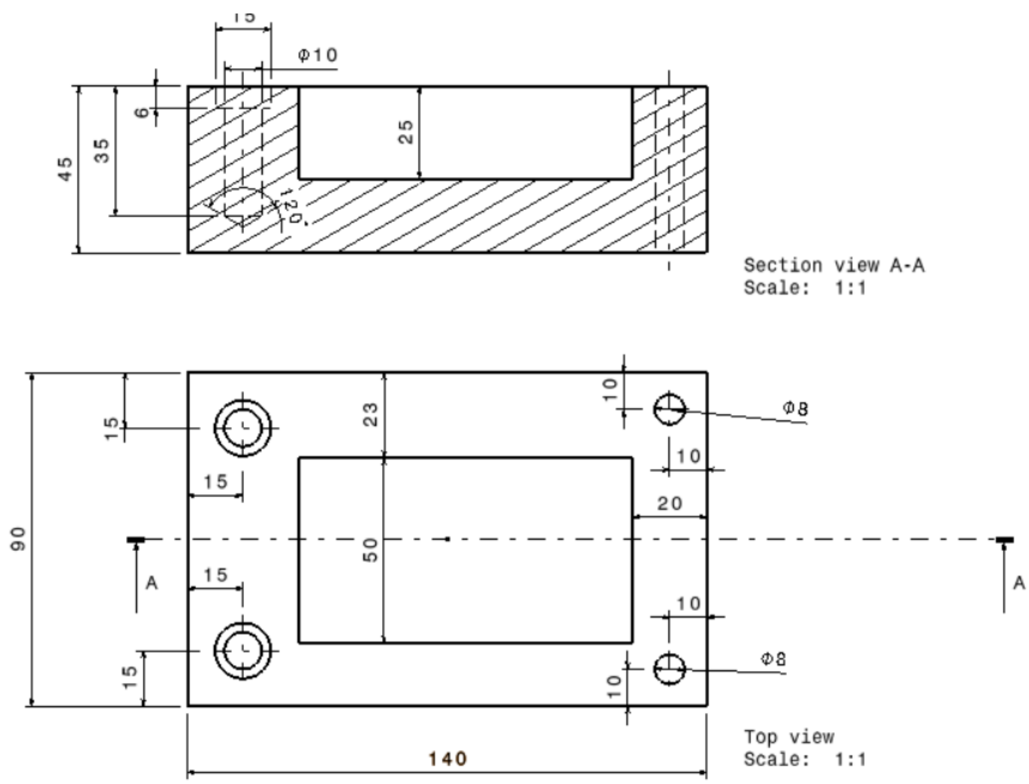

40
20
Front view
Scale: 1:1


R5
Φ10
R5
Φ10
Φ20
4×4
80
60
Φ10
Φ10
R5
R5
100
120
Top view
Scale: 1:1

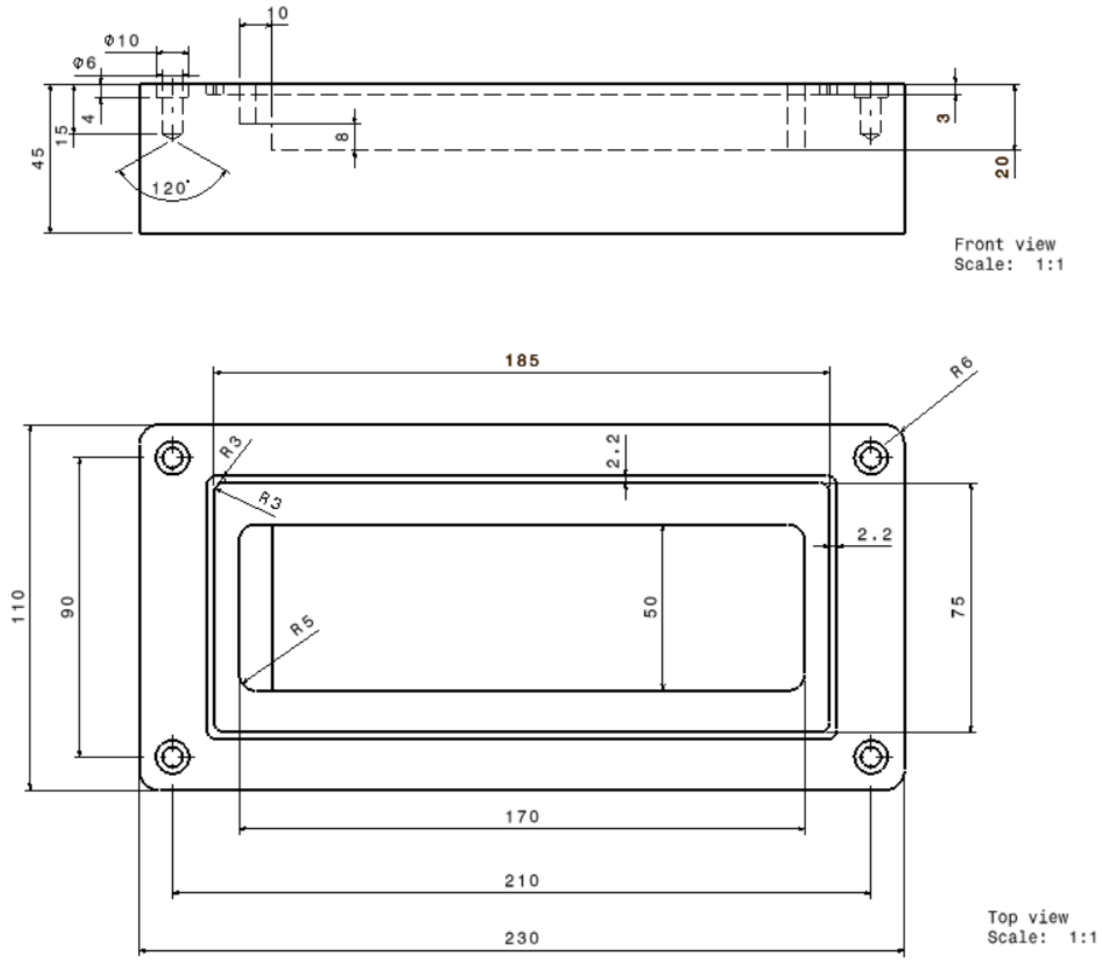
P5


Right view
Scale: 1:1
Front view
Scale: 1:1
Top view
Scale: 1:1
All chamfers 4 x 45°.
4 x R5
4x45°
Φ10
40
80
15
20
60
25
44
160
50

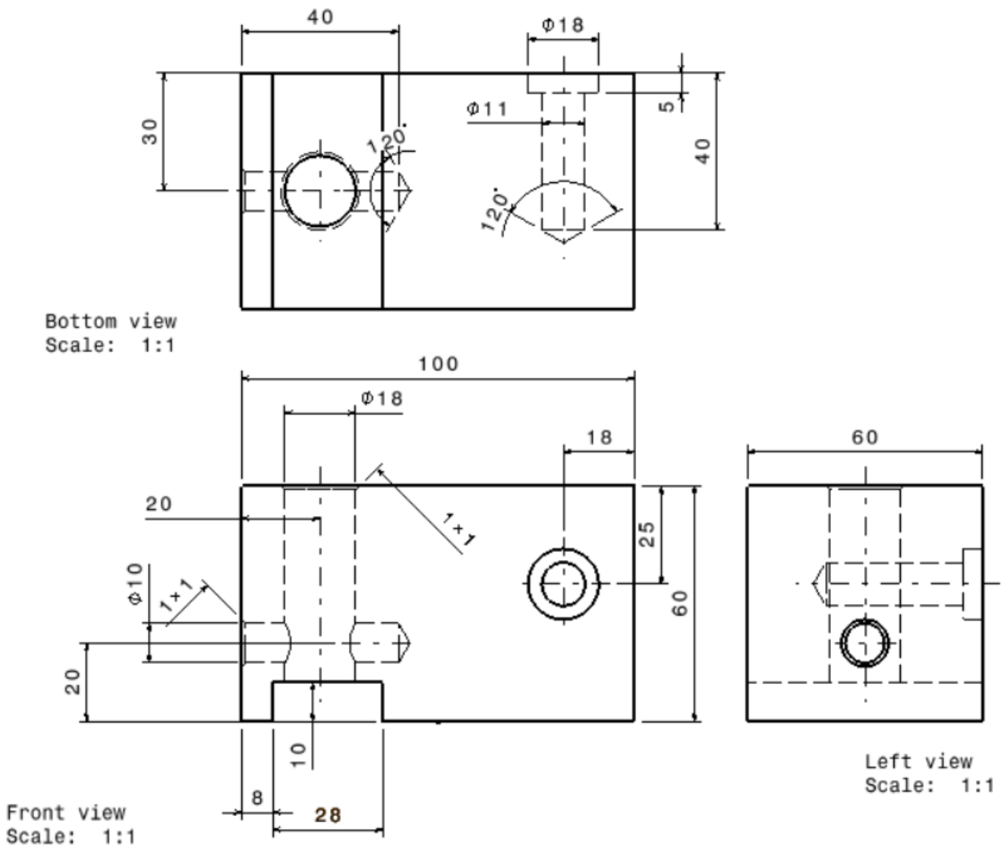
P6


Section view A-A
Scale: 1:1
Top view
Scale: 1:1
Φ10
Φ8
120°
45
35
6
25
15
23
50
90
10
20
140
A

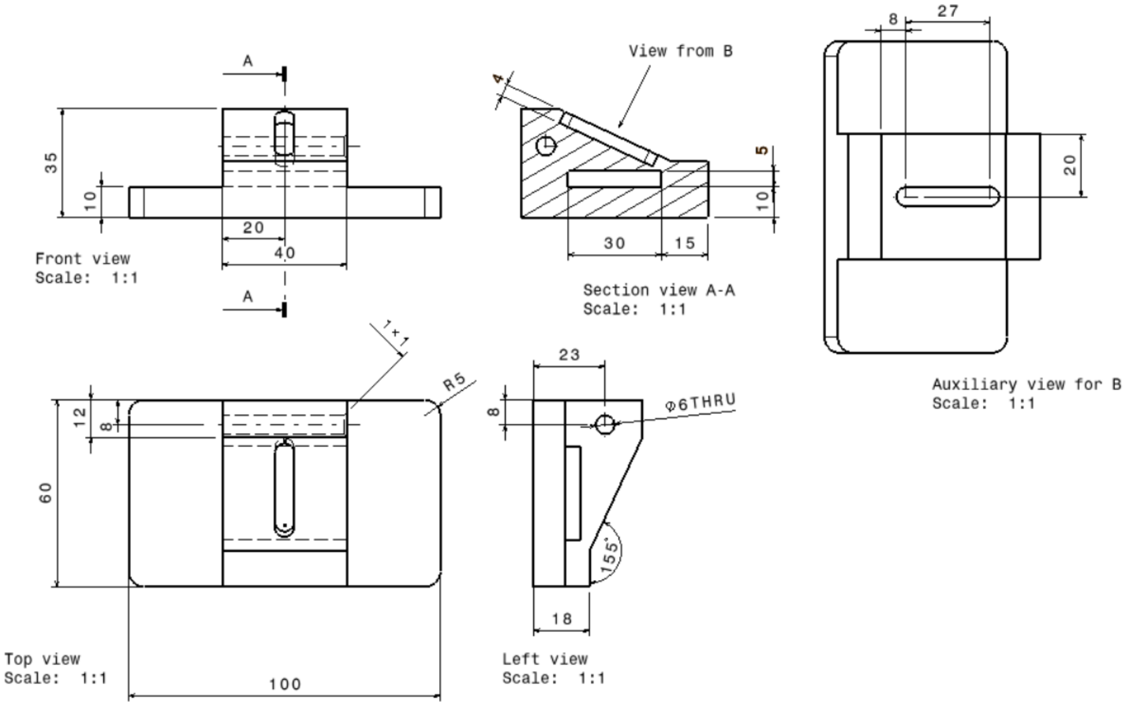
P7


Front view
Scale: 1:1
Top view
Scale: 1:1

P8


Bottom view
Scale: 1:1
Front view
Scale: 1:1
Left view
Scale: 1:1

P9



ANNEX 2 – Simple and structured prompts used in the study

Prompt 1:

You are a FreeCAD Python scripting assistant for prismatic mechanical parts.
Input:
- One uploaded image containing dimensioned drawings of a simple mechanical part.
Task: Inspect the drawings and generate:
1. a short modeling plan, and
2. one complete, executable FreeCAD Python script that reconstructs the part as a valid 3D solid.

Prompt 2:

You are a FreeCAD Python scripting assistant for dimensioned prismatic mechanical parts.

Input:
- One uploaded PDF containing dimensioned orthographic drawings, sections, detail views and/or auxiliary views of one mechanical part.

Task:
Inspect the drawing and generate:
1. a short modeling plan, and
2. one complete, executable FreeCAD Python script that reconstructs the part as a valid 3D solid.

Use a direct best-effort approach. Do not ask questions and do not wait for user validation.

Interpretation rules:
- Use all available views. Use top views mainly for XY layout and feature positions, and use front, side and section views mainly for heights, depths, hole axes and internal features.
- First model the main prismatic body, then added volumes such as blocks, lugs or wedges, then removed features such as holes, pockets, slots, notches and cut-outs.
- Keep feature types separate. Do not merge through holes, blind holes, counterbored holes, pockets, slots and notches.
- Pay special attention to hole axis direction. Vertical holes, lateral holes and end-face holes must be modeled along the axis indicated by the drawing.

- For pockets, slots and cut-outs, distinguish between through cuts and blind recesses when this is indicated by the views or sections.
- If a dimension is unreadable or ambiguous, use a clearly named editable variable near the beginning of the script and add a short comment.
- If a small detail such as a chamfer, fillet or drill tip is difficult to model robustly, simplify it and add a short code comment.
- Prioritize a valid, executable and structurally correct solid over excessive local detail.

Output rules:
- Do not display a dimension-extraction table.
- Do not display view-by-view reasoning.
- Final answer must contain only:
  1. A very short modeling plan, maximum 5 bullet points.
  2. One complete executable FreeCAD Python script.
  3. The FreeCAD commands needed to obtain surface area and volume.

The FreeCAD script must:
- use millimetres;
- use named variables for all important dimensions;
- create a single valid solid;
- use simple and robust FreeCAD Part operations whenever possible;
- build the model in this order:
  1. main base/body;
  2. added prismatic volumes;
  3. pockets, slots and notches;
  4. holes and counterbores;
  5. chamfers/fillets only if robust;
- recompute the document;
- print the computed surface area and volume at the end.

Use this coordinate convention unless the drawing clearly suggests another:
- X = main length;
- Y = depth/width;
- Z = height;
- origin at the lower-left-front corner of the main body.