

Machinable AI

Two Spatial-Reasoning Failure Modes in Frontier VLM Extraction of Z-Axis Dimensions from 2D Engineering Drawings

August 2026

by Garron Ware

ABSTRACT

Frontier vision-language models (VLMs) fail to extract the correct part thickness from production engineering drawings across dozens of repeated runs. The author prompted frontier VLMs from OpenAI, Anthropic, and Google for the bounding dimensions of machined parts on released production drawings. The models returned the correct length and width in essentially all runs. No run returned the correct thickness. Every failed run exhibited one of two failure modes. The first is a dimension-annotation association error, and its mechanism is spatial proximity bias: the model locates the correct dimension line but binds it to a nearer, unrelated numeral. GPT-5.5 (medium effort) returned 6.35 mm instead of 12.700 mm on drawing 201533-020. The second is a section-view selection error, and its mechanism is cross-view axis conflation: the model selects the wrong section view and reports a Y-axis feature dimension as the Z-axis thickness. Opus 4.6 (medium effort) returned 85.7 mm instead of 25.4 mm on drawing 201548-013. Both failures occur after correct character recognition. A system prompt that stated declarative line-type conventions reduced the first failure mode in follow-up testing. A procedural instruction to trace dimension lines did not.

01 Correct dimension extraction gates AI usefulness in machining workflows

Machinable builds AI software for CNC machining workflows. Quoting, stock selection, and setup planning all begin with the bounding dimensions of the part, and the part print is the authoritative source for those dimensions. If a model returns the wrong thickness dimension, it corrupts the stock-selection sequence. Machinable therefore cannot work unless the dimension-extraction step works reliably.

This paper documents two distinct failure modes on this extraction task. Both survive correct optical character recognition. The models read every numeral on the drawing correctly and still return the wrong number, because the errors occur in the spatial-reasoning steps that connect numerals to geometry. This paper names the two failure modes, locates each one at a specific stage of the interpretation pipeline, and reports which prompt-level mitigation worked and which did not.

02 The task: return three bounding dimensions from a multi-view 2D drawing

The test task is bounding-dimension extraction. The model receives a complete engineering drawing as an image and the prompt “Return bounding dimensions for this part.” For a prismatic part, a correct response contains three values: Thk (Z-axis), W (X-axis), and L (Y-axis). Table 1 lists the correct values for both test drawings. Each drawing dual-dimensions every value as millimeters [inches].

This task suits failure analysis: the ground truth is unambiguous, every response is easy to classify, and the task requires the model to coordinate information across multiple orthographic views, which is the capability under test.

03 Each failure mode is demonstrated on the production drawing that still reproduces it

Both test drawings are released production drawings from Jus-Rite Engineering Inc. The author owns the IP and reproduces them without redaction. The same engineer detailed both drawings, so both follow the same conventions: third-angle orthographic projection, ordinate dimensioning from marked zero datums, dual millimeter [inch] units, and a prismatic part shown in a front view, section views, and an isometric view.

DRAWING	PART	THK (Z-AXIS)	W (X-AXIS)	L (Y-AXIS)	DEMON- STRATES
201533-020	Bottom plate, MIC-6 aluminum	12.7 mm	215.9 mm	184.2 mm	Failure Mode 1A
201548-013	Locator block, 4140 steel	25.4 mm	228.6 mm	241.3 mm	Failure Mode 1B

Table 1 The two test drawings.

On both drawings, the bottom section view carries the thickness dimension and departs from drafting convention in two ways. First, convention places a dimension value at the midpoint of its dimension line, but on these drawings the numeral in that position is a hole-position callout, not the thickness. Second, the actual thickness value sits above the dimension line and offset to the left, farther from its line than convention places it. These departures confuse human readers as well. They make the drawings strong test instruments because resolving the ambiguity requires following line geometry instead of relying on text proximity.

The author assigned each failure mode to the drawing that still reproduces it. Six months before this paper, drawing 201548-013 repeatedly elicited both failure modes. As of August 2026, current models resolve the thickness association on 201548-013, so the author can no longer reproduce Failure Mode 1A there. The same failure still reproduces on 201533-020, and Failure Mode 1B still reproduces on 201548-013.

04 Method: repeated single-prompt runs across frontier VLMs produced zero correct thickness extractions

The author ran the extraction task dozens of times across the major frontier VLMs from OpenAI, Anthropic, and Google, in both chat interfaces and API calls, using the bounding-dimension prompt described above. The author also ran the same prompt against a set of four drawings that share these dimensioning characteristics. This paper documents two runs with screenshots: GPT-5.5 (medium effort) on drawing 201533-020, and Opus 4.6 (medium effort) on drawing 201548-013.

Every run produced the same three results. The models returned the correct W (X-axis) and L (Y-axis) values. No run returned the correct thickness. Every incorrect run matched one of the two failure modes described in Sections 5 and 6. The author did not retain per-run logs with model versions and counts, so this paper reports these results qualitatively.

05 Failure Mode 1A: the model binds the thickness dimension line to the nearest numeral instead of the displaced correct value

Drawing 201533-020 demonstrates this failure mode (Figures 1 through 3). In the bottom section view, the correct thickness value, 12.700 [.5000], sits above the dimension line and offset to the left. The numeral nearest the dimension line is 6.36 [.251], the Z-axis position callout of two #10-32 tapped holes measured from the datum face. GPT-5.5 (medium effort) returned a thickness of 6.35 mm (0.250 in) instead of 12.700 mm (Figure 4). The drawing also prints 6.35-family values in its dowel callouts (6.350 [.2500]), and the model's cited source string matches that form. Under either resolution, the model bound the thickness to a nearby wrong numeral instead of following the dimension line to the displaced 12.700 [.5000] label.

We label this failure mode a dimension-annotation association error, and we attribute it to spatial proximity bias: when the intended value sits unusually far from its dimension line and an unrelated value sits closer, the model selects the nearby value. The model selects the correct view and the correct dimension line, and it reads every numeral correctly. It fails only at the association step. The

Drawing 201548-013 elicited the same failure mode six months earlier: models returned 12.7 (a dowel-hole position callout) instead of the displaced thickness value 25.4. In both drawings the misbound numeral marks a hole position near mid-thickness, so the model returns roughly half of the true thickness.



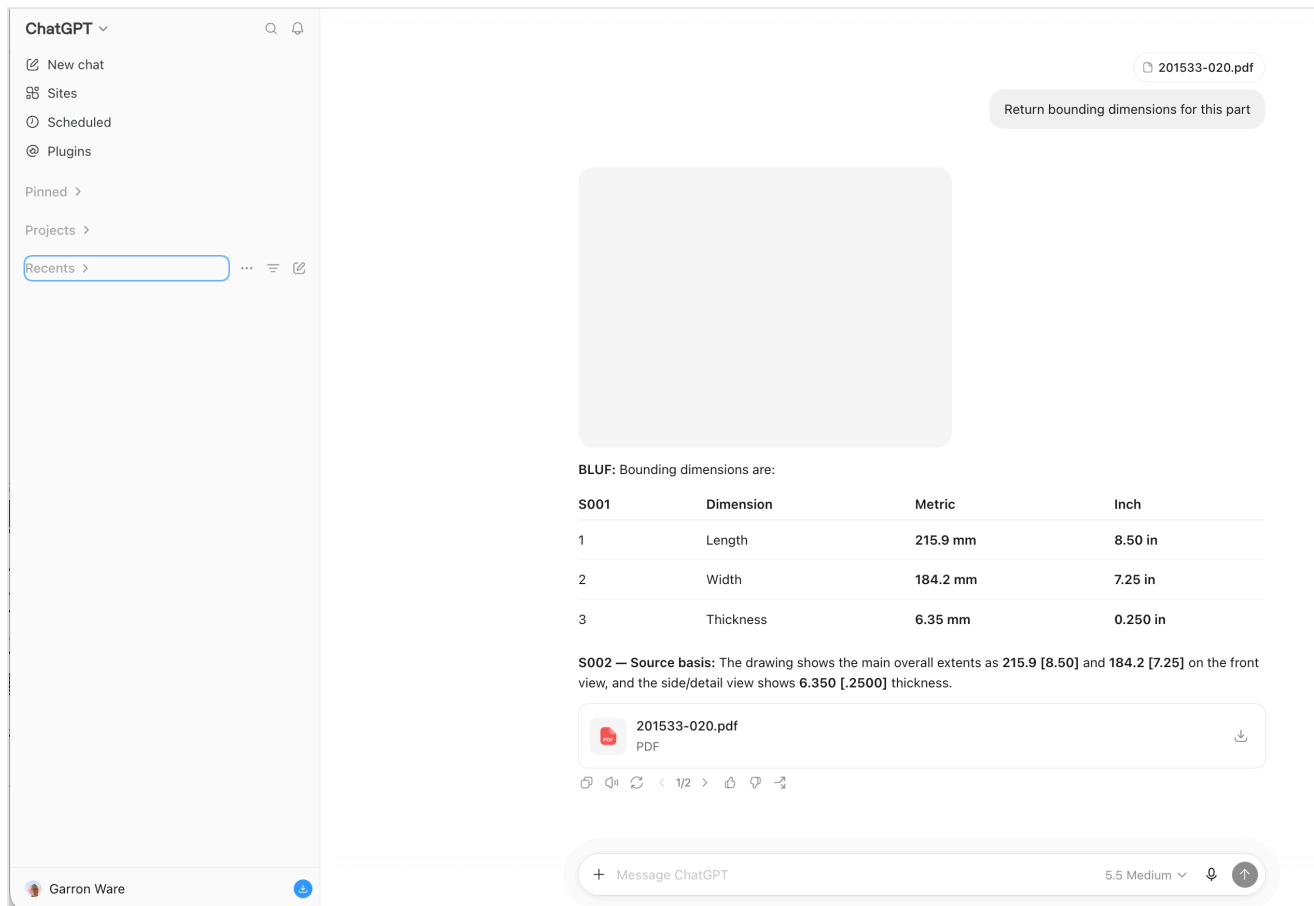


Figure 4 GPT-5.5 (medium effort) returns 6.35 mm as the thickness of a 12.700 mm part.

06 Failure Mode 1B: the model selects the wrong section view and reports a Y-axis span as the Z-axis thickness

Drawing 201548-013 demonstrates this failure mode (Figures 5 through 7). In this failure mode, the model does not use the bottom section view at all. It selects the right-side section view, takes the 85.7 [3.38] dimension from it, and reports 85.7 as the thickness. That dimension measures the distance from the edge of the part to a machined groove. In the right-side section, the vertical direction of the view corresponds to the part's Y-axis, not its Z-axis. The model treats “vertical in a side view” as “thickness” and returns a Y-axis feature span as the Z-axis bounding extent.

Figure 8 shows a documented run. Opus 4.6 (medium effort) returned, under the model's own labels, 228.6 mm as length, 241.3 mm as width, and 85.7 mm as height. This output is internally inconsistent as a bounding box: 241.3 and 85.7 both measure along the part's Y-axis, and the Z-axis appears nowhere in the answer. The model did not run that consistency check.

We label this failure mode a section-view selection error with cross-view axis conflation. The model does not maintain a global part-coordinate frame across the orthographic views. Each view is a projection of the same X-, Y-, Z-frame, and the axis that appears vertical differs by view: the Z-axis is vertical in the bottom section, and the Y-axis is vertical in the right-side section. The model instead pattern-matches, picking a view that looks like a side view and a dimension that looks like a thickness.

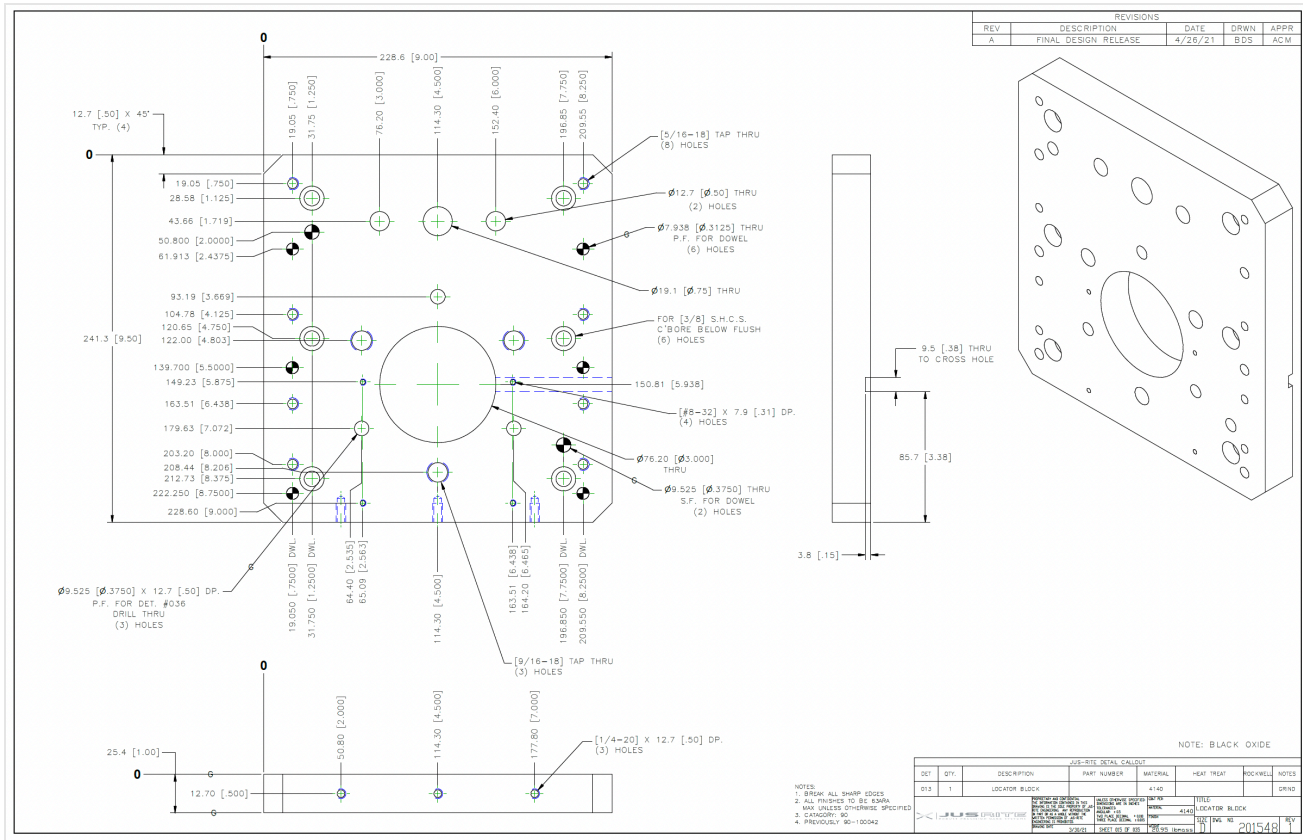


Figure 5 Drawing 201548-013, locator block, unmarked.

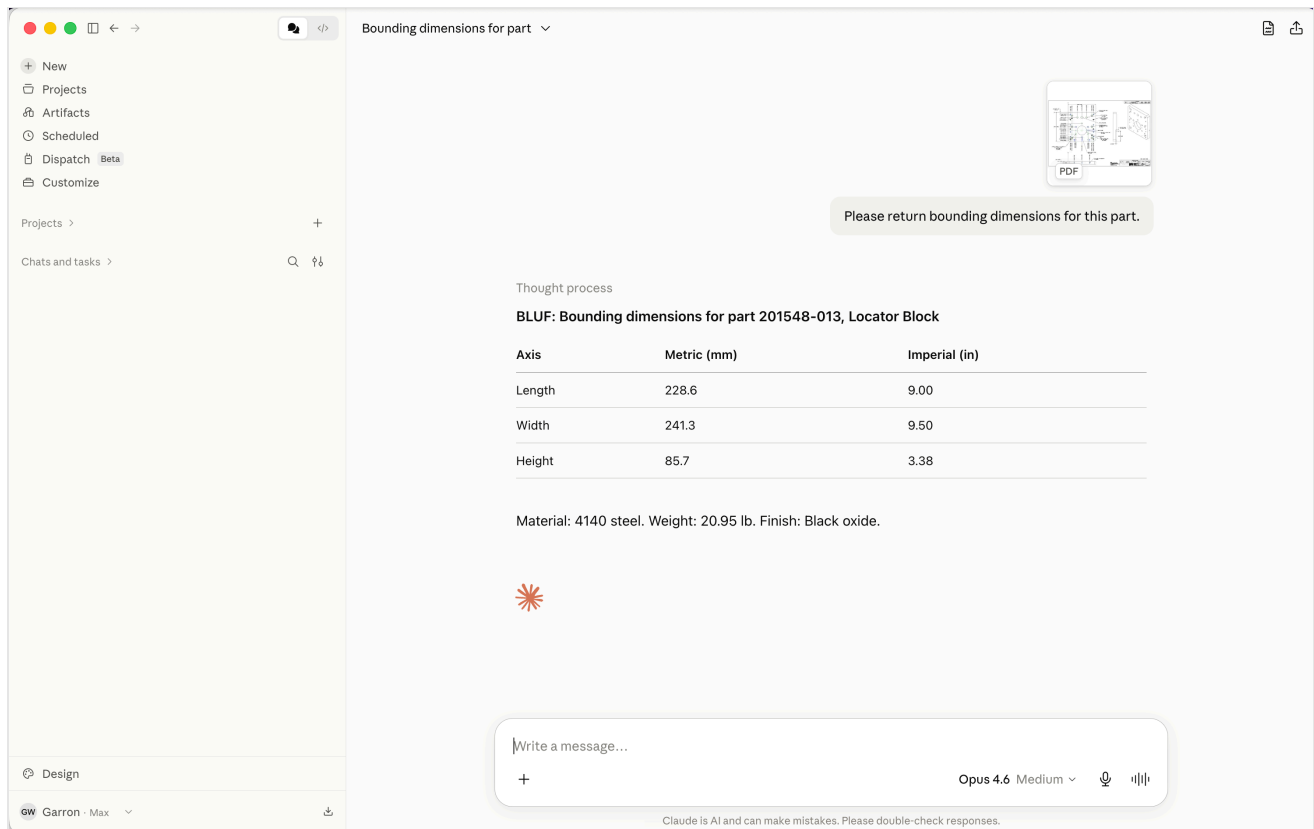


Figure 8 Opus 4.6 (medium effort) returns 85.7 mm as the height of the part.

07 The two failure modes break different stages of the same interpretation pipeline

Extracting a named dimension from a multi-view drawing decomposes into five stages: (1) recognize the characters, (2) bind each numeral to its dimension line, (3) select the view that carries the target dimension, (4) map that view's axes to the part's global coordinate frame, and (5) report the value. Failure Mode 1A breaks stage 2. Failure Mode 1B breaks stages 3 and 4. Neither breaks stage 1, and this is the central observation: current frontier VLMs read engineering drawings at the character level and fail at the geometric-reasoning level.

COMPARISON	Failure Mode 1A	Failure Mode 1B
Name	Dimension-annotation association error	Section-view selection error
Mechanism	Spatial proximity bias	Cross-view axis conflation
Documented run	GPT-5.5 medium, drawing 201533-020	Opus 4.6 medium, drawing 201548-013
Returned value	6.35 mm	85.7 mm
Correct value	12.700 mm	25.4 mm
View selected	Correct	Wrong (right-side section)
Dimension line selected	Correct line, wrong label	Wrong line entirely
Pipeline stage broken	Numeral-to-line binding	View selection and axis mapping
Error detectable from output alone	No	Yes (Y-axis used twice, Z-axis absent)

Table 2 The two failure modes compared.

The distinction matters for mitigation. Stage 2 failures respond to information about local line semantics. Stage 3 and 4 failures require the model to construct and hold a 3D coordinate frame across views, which no local rule supplies.

08 Declarative drafting conventions in the system prompt reduced Failure Mode 1A; procedural tracing instructions did not

The author tested mitigations by adding instructions to the system prompt of the VLM API calls. A procedural instruction failed to produce repeatable improvement: “Trace dimension and extension lines to the feature or outer edge they measure. Do not assign a number based only on nearby text placement.” Telling the model how to trace the drawing did not make it trace the drawing.

A declarative statement of drafting conventions worked on a repeatable basis. The working system prompt defined the line taxonomy of an engineering drawing: boundary lines (thick continuous lines marking visible part edges), center lines (dash-dot lines marking symmetry and circular features), extension lines (thin continuous lines projecting from the part to bound a dimension, never touching the part), dimension lines (thin continuous lines with arrowheads, annotated with the

dimension value), and hidden-edge lines (thin dashed lines marking edges not visible in the view). With those definitions supplied, the model resolved the thickness association correctly across repeated runs.

Model updates confound the attribution: frontier models improved during the testing period, so the author cannot assign the improvement cleanly to the prompt change alone. Drawing 201548-013 shows the same confound from the other direction: current models resolve, without any prompt support, the association that failed repeatably there six months earlier. With that caveat, the pattern suggests the models do not reliably retrieve drafting-convention knowledge at inference time, and that supplying the conventions as context recovers part of the missing capability. The author has not observed a prompt-level mitigation that repeatably corrects Failure Mode 1B.

09 Machinable narrows each production VLM call to a single read and verifies the results deterministically

Machinable's current implementation (private repository, reviewed August 2026) does not ask a general-purpose VLM for everything at once, and it no longer uses the Werk24 parser described in the author's earlier articles. One single-purpose GPT-5.6 Sol (high effort) call returns only the maximum external bounding dimensions of the finished part. Separate calls read drawing-specified stock callouts and the material callout, and the fields resolve independently, so one failed read does not corrupt the others. When the bounding read misses a stock-blocking cross-section dimension or the overall length, one conditional follow-up call reviews high-resolution crops around the candidate callouts. Deterministic Python applies the stock allowance and calculates cut length and yield. A typed correction endpoint lets a machinist fix any field without rerunning model calls. The review UI requires a machinist to verify every result against the original drawing before material is ordered or cut.

This architecture tracks the failure boundary documented above. Narrow single-purpose prompts shrink the response surface, high-resolution crops remove the competing numerals that trigger proximity bias, and mandatory human review catches the errors that remain.

10 Limitations

This paper reports the author's repeated testing, not a controlled benchmark. The author retained exact model versions only for the two screenshot-documented runs (GPT-5.5 medium and Opus 4.6 medium), and did not retain per-run logs or counts, so failure rates are reported qualitatively. The drawings come from one machine shop and one detailer, and they deliberately violate two

conventions, so the measured behavior bounds the worst case rather than the typical case. The reproducibility claims in Section 3 rest on the author's recent re-testing, not on logged runs. The mitigation result in Section 8 is confounded by model updates that occurred during the testing window. No ablation isolates how far a label must be displaced before proximity bias takes over.

REFERENCES

- 1 Ware, G. "MFG_06: Building AI for Machining Starts with the Part Print." *Imperfect Entrepreneurship*, Article 012.
imperfectentrepreneurship.substack.com/p/mfg_06-building-ai-for-machining
- 2 Ware, G. "018: Getting VLMs to Resolve Spatial Reasoning-Based Dimensional Ambiguity in 2D Engineering Drawings." *Imperfect Entrepreneurship*, Article 018, August 25, 2026.
imperfectentrepreneurship.substack.com/p/018-getting-vlms-to-resolve-spatial
- 3 Ware, G. "016: First Update to Machinable after Build Week." *Imperfect Entrepreneurship*, Article 016.
imperfectentrepreneurship.substack.com
- 4 Machinable research summary.
machinable.ai/research.html