Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Information, Not Instruction: What Improved an AI Agent Reading Interior Millwork Drawings

Donald Dujardin

donald@codraft.nl

CoDraft

Research Article

Keywords: multimodal models, AI agents, technical drawings, interior millwork, quantity takeoff, placebo-controlled evaluation, construction

Posted Date: November 9th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11638798/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare potential competing interests as follows: The author is the founder and owner of CoDraft, which develops the commercial quoting software in which the described agent is used.

Information, Not Instruction: What Improved an AI Agent Reading Interior Millwork Drawings

Donald Dujardin
CoDraft, Oisterwijk, the Netherlands
donald@codraft.nl

November 2026

Abstract

Interior millwork drawings spread each cabinet across elevations, sections and notes, and a priced calculation needs every part with its dimensions. We built an agent that reads such drawings into a structured per-item dossier and leaves every derived quantity to deterministic code, and between July and November 2026 tested more than 250 changes against answer keys fixed before each experiment. This report describes the ones that held, and what they share: each gave the agent new information rather than new instructions. Viewing sheets in tiles raised the share of dimensions found from 15.4% to 96.2% on one sheet, and from 2 to 34 of 62 on another. Using labels from the PDF text layer as fixed anchors raised correct object binding from 48.4% to 93.3%, against 20.6% for placebo labels. One worked example per tool cut malformed tool calls by 73%. A domain fact delivered at the moment a part is booked raised correct cabinet depths from 9 to 16 of 23, and a targeted question from a code detector was followed by a repair in 35% of cases, against 13% for values that were not raised. Generic reading instructions, by contrast, did no better than a length-matched placebo. To our knowledge this is the first measured study of an agent reading interior millwork drawings.

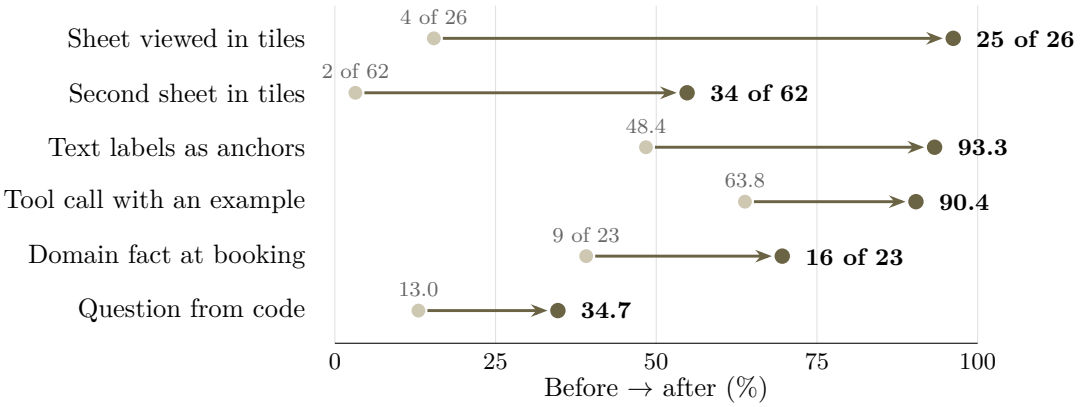


Figure 1: Each intervention reported here, before and after, as a percentage of its own measure: dimensions found on a sheet, objects bound to the right element, tool calls well formed, cabinet depths correct, flagged values repaired. Measures differ per row; Table 2 gives them in full.

1 Introduction

A quote for bespoke interior work starts with drawings: elevations, sections, plans and notes, often produced by an architect who did not draw for the workshop. An estimator reads them

into a parts list, every panel, front and shelf with its dimensions and material, before any price is computed. That reading is the costly step, and it is where multimodal models are weakest: on construction and engineering drawings they read text well but count and measure poorly [2–4].

Throughout, a *multimodal model* is a language model that also reads images (often called a vision-language model), and an *agent* is such a model working through a drawing set in a loop of tool calls over many steps. We built an agent that performs this reading for a commercial quoting product and measured, change by change, what improved it. Every change was tested against answer keys fixed before the experiment, and since October every change to the instructions has been tested beside a placebo of the same length. Most changes did not survive. The ones that did share one property: they gave the agent information it did not have, as pixels, anchors, examples, domain facts or findings from code, rather than telling it more firmly how to behave.

Our contributions are:

1. Five interventions that measurably improved drawing reading, each with its measure, baseline and effect (§6, Table 2).
2. A placebo-controlled test of the instruction channel, in which curated reading rules did no better than text of the same length without task content (§6.3).
3. An evaluation protocol for agent harness work: frozen answer keys, pre-registered decision rules, placebo arms and an audit of our own measurements (§5).

To our knowledge no earlier study measures an agent reading interior millwork drawings down to every part and its dimensions (§2).

2 Related work

Drawings. Handoff-H1 [1] produces residential takeoffs with perception models and orchestrated vision agents and scores close to professional estimators, but reports no ablations. DrawingVQA [2] finds quantity takeoff the weakest category for current models. AECV-Bench [3] reports low exact-match accuracy for counting doors and windows on floor plans, and MechVQA [4] benchmarks mechanical drawing understanding at scale. BLUEPRINT [5] retrieves from engineering archives by reading identifiers in detected drawing regions, and agent pipelines for structural drawings, one with component ablations, are described in [6, 7]. Table 1 places this work beside them.

Seeing detail. Multimodal models fail to translate fine visual detail into words [8] but know where to look, so cropping helps [9]; learned crop routing raises accuracy further [10], and marks, sketching and agentic visual tools extend the idea [11–13]. Extracted text helps retrieval more than reading [14].

Instructions and context. Input length alone can lower performance even when all relevant information is retrieved [15], which motivates our length-matched placebo. Tool and interface design shape agent performance [16–18].

Feedback. Models rarely correct their own errors without new external information [19–21]; moving an error into an external role is what makes it visible [22]. Agents also commit to an interpretation early [23].

Evaluation. Randomness in agentic evaluation [24] and the validity of model-based judges [25, 26] motivate our protocol.

Work	Drawings	Agent	Ablations	Output
Handoff-H1 [1]	residential	yes	no	quantities
DrawingVQA [2]	construction	no	no	answers
AECV-Bench [3]	floor plans	no	no	counts
MechVQA [4]	mechanical	no	no	answers
BLUEPRINT [5]	engineering archive	no	oracle only	retrieval
BlueprintAgent [6]	structural	yes	yes	frame model
This work	**interior millwork**	**yes**	**yes**	**every part + dimensions**

Table 1: Work on AI for technical drawings. Agent: a model that uses tools over several steps. Ablations: the contribution of each component is measured.

3 Task and data

A project is a set of PDF sheets: elevations with dimension chains, plans, sections, sometimes perspectives and renders, and a legend that assigns materials to kinds of parts. Many sheets are raster images without a usable text or vector layer, and CAD files are never available. Not every dimension is printed: in a count over eight projects, 133 of 392 length and width dimensions do not appear as a number and must be derived from a chain, taken from a neighbouring part or measured at scale.

The output is a parts list per furniture item: each part with its kind, two dimensions with their source, material and count. Hidden construction such as bottoms, backs and internal walls is completed by code from the outer dimensions.

We use real projects from Dutch interior workshops, drawn by different offices. Answer keys now exist for seven projects (A–G). Drawings are client property and are not shown; Figure 3 uses an example drawn for this paper.

4 System

The agent receives the drawing set, views each sheet whole and then in tiles and targeted crops, and records findings in a text dossier organised per furniture item through a small set of tools. Deterministic code converts the dossier into a calculation: carcass parts, edging, hardware, sheet material and labour. The model interprets, code computes, and a person reviews every result and decides every price (Figure 2).

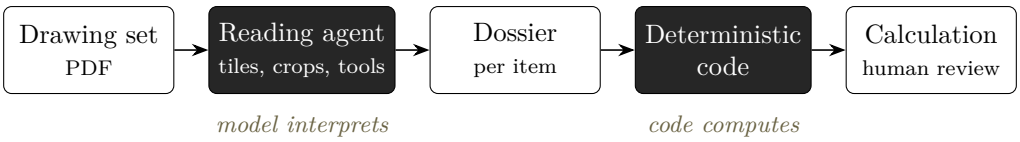


Figure 2: Reading pipeline. Only the agent uses a language model; every quantity after the dossier is computed by code, so the same dossier always yields the same calculation.

The work ran in two phases. From July to August 2026 the reader was a single vision call per sheet with passes added around it; from September it became a tool-using agent that writes the dossier. Sections 6.1 and 6.3 start in phase one; the other results come from phase two. Phase one used Claude Opus 5 and phase two Claude Opus 5.5 (Anthropic), both through the Anthropic API.

5 Evaluation protocol

Answer keys. For each project a key lists every part the calculation needs, with its dimensions. It is fixed before the experiments that use it and never shown to the agent. Disagreements are adjudicated on the drawing; when the key is wrong it is corrected and versioned.

Pre-registration. Each experiment fixes its arms, metric and decision rule before the first model call. In seven of nine pre-registered experiments the effect was smaller than predicted. Since October 2026 every change to the instructions runs beside a length-matched placebo arm.

Unit of measurement. Each result is reported in its own measure, the one fixed before that experiment: dimensions found on a sheet, objects bound, tool calls, values correct or errors repaired. Small experiments replay the moment a decision is made with fresh readers; larger ones use full runs on a project.

Noise. Two runs of an identical configuration on one project scored 54 and 49 of 55 parts. We therefore report every run where there are several and treat differences of a few parts as noise.

Auditing the instrument. We hold our measurements to the same standard as the agent. An audit of all 117 experiment packages found and corrected measurement errors in 33 of them: instruments grading themselves, replicas that were not the production function, wrong denominators, and metrics rewarding side effects. Only 4 of the 64 instruments behind standing conclusions were affected (6.3%), and those conclusions were corrected. After a later correction of the scorer, all 378 stored runs were re-counted.

6 Results

Table 2 lists every intervention reported here; the subsections describe each with its experiment.

Intervention	Measure	Before	After	Effect	95% CI (%)	n
Viewing in tiles (§6.1)	Dimensions found	4/26	25/26	+80.8 pp	6–34 / 81–99	8 calls
second sheet	Dimensions found	2/62	34/62	+51.6 pp	1–11 / 43–67	3 runs/arm
Text-layer anchors (§6.2)	Objects bound	15/31	28/30	+44.9 pp	32–65 / 79–98	15 calls
Placebo vs. rules (§6.3)	Parts bound	58.6%	69.0%	+10.4 pp	41–75 / 51–83	48 calls
One example per tool (§6.4)	Malformed per 100	36.2	9.6	−73%	30–43 / 6–15	2 + 1 runs
Domain fact at booking (§6.5)	Depths correct	9/23	16/23	+7	22–59 / 49–84	144 decisions
Question from code (§6.6)	Flags repaired	13.0%	34.7%	+21.7 pp	10–17 / 28–42	142 runs

Table 2: Interventions reported in this paper. 95% CI: Wilson intervals of the before and after proportions, in percent; for tool calls, pooled over the two baseline runs. The placebo row compares two arms (rules before, placebo after) rather than a change over time. For the anchors, placebo labels scored 7 of 34.

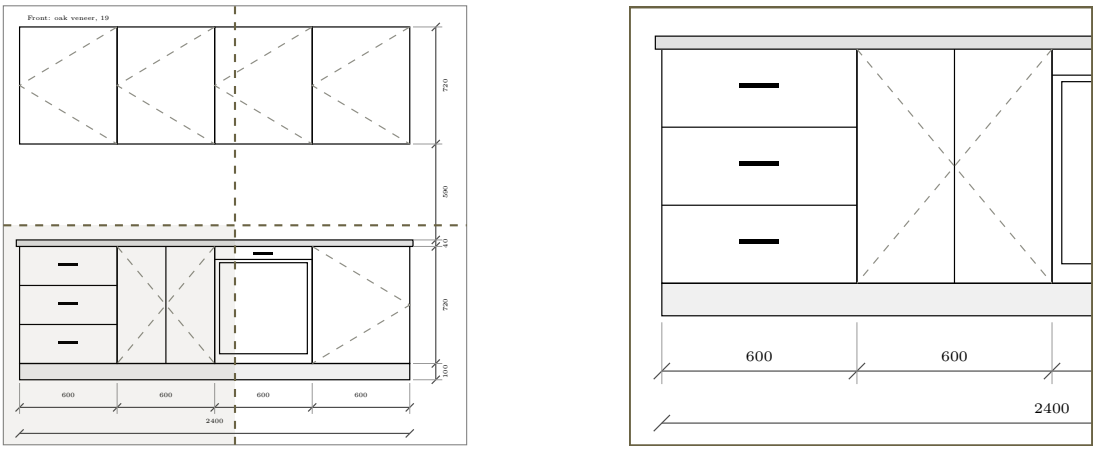
6.1 Show the sheet

What the literature predicts. Models miss fine detail in a whole image but know where to look [8–10]. On drawings, dimension figures are small relative to the sheet.

Experiment. We compared reading a sheet whole against reading it in a 2×2 grid of tiles at a larger scale (Figure 3), with the same prompt and model, so that the image was the only difference.

Result. On one sheet the tiled reading found 25 of 26 dimensions (96.2%), against 4 of 26 (15.4%) for an earlier reading of the whole sheet (Figure 4). On a sheet whose dimension figures

were 4.7 pixels high, with prompt and model held fixed, tiles found 34 of 62 dimensions against 2 of 62 (three runs per arm), and on a third, 31 dimension findings against 2. A second pass over regions the first pass left empty added 19.1 percentage points of coverage across projects. Beyond legibility, more resolution did not help: once the agent could zoom, a different zoom tool gave the same 17 correct answers in each of three draws, re-rendering 709 real zooms at 400 dpi gave no gain in 619 of them, and a 2×2 grid on sparse sheets quadrupled cost without adding content.



(a) Whole sheet, four tiles. **(b)** Lower-left tile at twice the scale.

Figure 3: Viewing a sheet in tiles. In the whole sheet the dimension figures are a few pixels high; in a tile they are legible. Example drawing made for this paper.

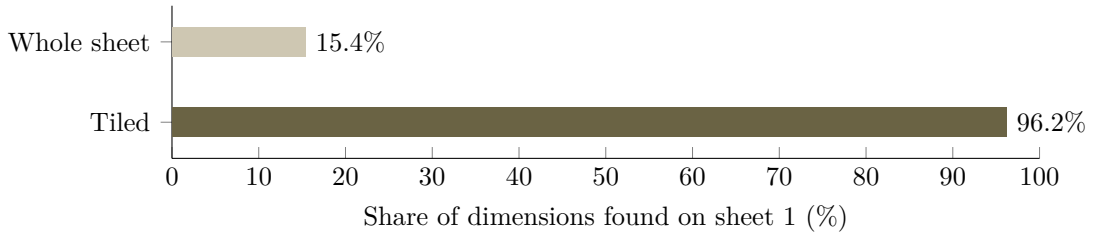


Figure 4: Viewing one sheet in tiles instead of as a whole: 25 of 26 dimensions found, against 4 of 26.

6.2 Text as an anchor, not as content

What the literature predicts. Extracted text helps a model find a region more than read it [14]; marks on an image improve grounding [11].

Experiment. Adding the PDF text layer as extra content to read did not help (97 against 95 of 112 points). We then used the same labels differently: as fixed anchors to which the agent binds each object, against no anchors and against placebo labels.

Result. Over 15 calls on five sheets from four projects, correct object binding rose from 15 of 31 (48.4%) to 28 of 30 (93.3%, 95% interval 79–98%); placebo labels of the same length scored 7 of 34 (20.6%), and none of 44 anchor citations was invented (Figure 5). On tiles, anchored binding went from 0 of 25 to 29 of 31.

6.3 Instructions against a length-matched placebo

What the literature predicts. Input length alone changes model behaviour, even with irrelevant tokens replaced by whitespace [15].

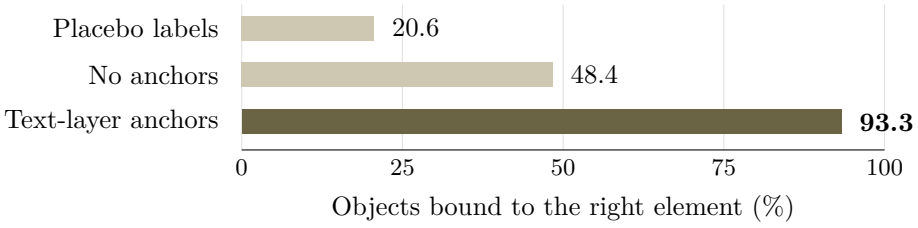


Figure 5: The text layer used as anchors, against no anchors and placebo labels.

Experiment. We compared three versions of one instruction block of equal length: five curated reading rules, one worked example replacing them, and a placebo without task content; 48 calls in two rounds.

Result. Coverage was identical in all three arms (26 of 29), and on correct part binding the placebo scored highest (Figure 6). In none of the four placebo-controlled experiments of the first phase could content be told apart from filler, and in two of them the placebo was the best arm; through the summer it won or tied in every one, and in October a new reading rule again scored no better than its placebo (31 against 33 rows) and was removed. Since then every instruction change is tested beside a placebo, which is how the interventions below were found.

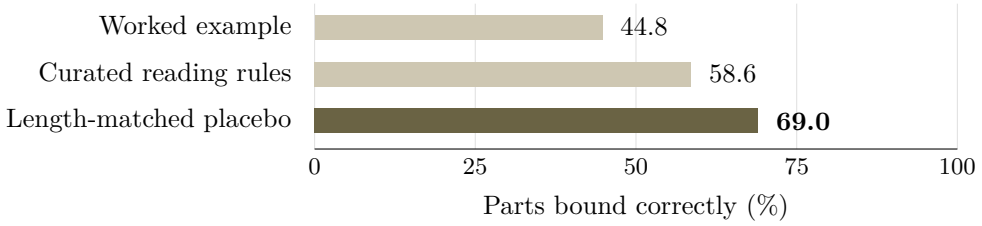


Figure 6: Three instruction blocks of equal length on the same sheets.

6.4 One example per tool

What the literature predicts. The interface between agent and environment shapes performance as much as the model [16]; tool descriptions benefit from examples [18].

Experiment. We added one short worked example to each tool description, and separately replaced 45 editing tools with one.

Result. Malformed tool calls fell from 39.0 and 33.3 per 100 calls in two baseline runs (mean 36.2) to 9.6 (16 of 166 calls) with the examples (Figure 7), with the score unchanged within noise (38 of 48 calculable parts, against 39 and 43 in two baseline runs). A single editing tool kept the score (51 of 55, against 47 and 48) and halved false findings, from 4 to 2.

6.5 A domain fact at the moment of booking

What the literature predicts. Generic instructions did not help (§6.3); the question is whether specific knowledge does when it arrives where it is needed.

Experiment. At the moment the agent books a part, a fresh reader received the workshop's domain facts for that kind of part, against the same reader without them, on 144 decisions across two projects, with fresh readers alternating between arms. Separately, a rule for a layer drawn without a dimension was replayed at six booking moments, with the rule alone and with the rule plus an example card.

Result. Correct decisions rose from 106 to 115 of 144, meeting the pre-registered threshold of +7; most of the gain came from cabinet depths on one project, from 9 to 16 of 23 (Figure 8).

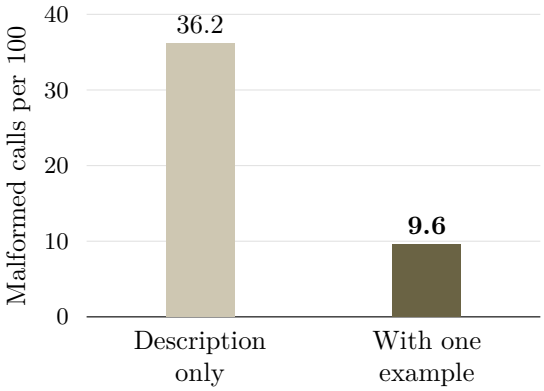


Figure 7: Malformed tool calls before and after adding one example to each tool description.

The layer rule, tested with 24 fresh readers on four projects, raised correct values from 0 to 5 of 12, and to 7 of 12 with the example card, while all 12 control cases stayed correct.

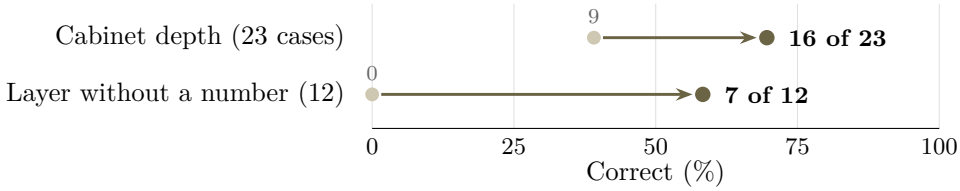


Figure 8: Correct values with the domain fact delivered at the moment of booking.

6.6 A targeted question from code

What the literature predicts. Models rarely repair their own errors without new information [19, 20], and an error becomes visible when it comes from an external role [22].

Experiment. When the agent first reported that it was done, code detectors checked the dossier, and their findings were put to the agent as targeted questions. Over 142 stored runs we counted which values that were wrong at that moment were repaired by the end of the run, for values that were raised and values that were not. This is a count over runs, not a randomised comparison.

Result. Values raised as a question were repaired almost three times as often: 58 of 167 (34.7%) against 41 of 315 (13.0%) for values not raised, while 8 correct values were changed to wrong against 2 (Figure 9). A recompute tool that let the agent check its own arithmetic changed nothing in six of six cases where it disagreed.

7 Discussion

Across the five interventions, what helped was information the agent did not have: more pixels of the right region, a fixed anchor, an example of a correct call, a domain fact, a finding from code. What did not help was more or firmer instruction about how to read, which in our placebo tests behaved like text of the same length. For practitioners this suggests a test for any proposed change to an agent: what does it let the agent see or know that it could not before? A change with no answer is a candidate for a placebo arm.

Validity. Results come from real projects by different drawing offices and from one model family (Claude). Agent runs vary between identical configurations, so we report every run

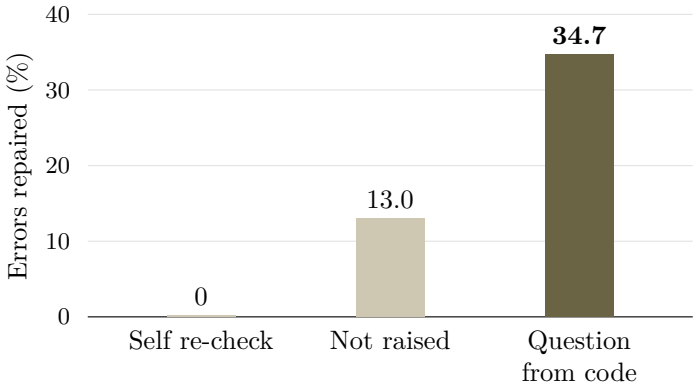


Figure 9: Wrong values repaired by the end of the run, by what the agent received. Self re-check: $n = 6$; not raised: $n = 315$; raised as a question: $n = 167$.

where there are several and treat differences of a few parts as noise. Each result is reported in the measure fixed before its experiment, and measures differ between results. The anchor result applies to sheets that carry a PDF text layer. Drawings are client property; answer keys and runs are described, not released.

8 Conclusion

An agent reading interior millwork drawings improved when it was given information, and not when it was given instructions. Showing the sheet in tiles, anchoring objects to the text layer, adding an example to each tool, delivering a domain fact at the moment of booking and raising findings from code as targeted questions each moved its measure. Instruction changes should be tested against a length-matched placebo before they are trusted.

Data availability

Drawings and answer keys are client property and cannot be released.

Use of AI

Code, experiments and analyses were produced with AI coding agents directed by the author. All reported numbers trace to logged experiments with pre-registered decision rules.

References

[1] B. Chicelli, H. Alves, R. Anselmo et al. Handoff-H1: An orchestrated vision-agent system for material quantity takeoff from construction blueprints. arXiv:2608.15032, 2026.

[2] Y. Jung, J. Fu and M. Golparvar-Fard. DrawingVQA: A real-world benchmark for multi-depth visual-textual reasoning on construction drawings. CVPR Findings, arXiv:2607.15418, 2026.

[3] A. Kondratenko, M. Birhane, H. E. Hsain et al. AECV-Bench: Benchmarking multimodal models on architectural and engineering drawings understanding. arXiv:2601.04819, 2026.

[4] Q. Kou, X. Shi, Y. Li et al. MechVQA: Benchmarking and enhancing multimodal LLMs on comprehensive mechanical drawing understanding. ICML, arXiv:2605.30794, 2026.

[5] E. Seefried, R. Eldegaway, S. Das et al. BLUEPRINT: Rebuilding a legacy: Multimodal retrieval for complex engineering drawings and documents. arXiv:2602.13345, 2026.

[6] Z. Xu, C. Yang, L. Wang et al. BlueprintAgent: Constraint-triggered targeted revisits for simulation-ready generation from scanned structural blueprints. arXiv:2609.07362, 2026.

[7] M. Talebi-Kalaleh and Q. Mei. Training-free agentic computer vision for structural component detection in 2D structural framing plans. arXiv:2608.17237, 2026.

[8] P. Rahmanzadehgervi, L. Bolton, M. R. Taesiri et al. Vision language models are blind: Failing to translate detailed visual features into words. arXiv:2407.06581, 2024.

[9] J. Zhang, M. Khayatkhoei, P. Chhikara et al. MLLMs know where to look: Training-free perception of small visual details with multimodal LLMs. arXiv:2502.17422, 2025.

[10] J. Zhu, R. Fu, Y. Ding et al. Learning to look again: Loss-gap supervision for free-form crop routing in vision-language models. arXiv:2608.21762, 2026.

[11] J. Yang, H. Zhang, F. Li et al. Set-of-Mark prompting unleashes extraordinary visual grounding in GPT-4V. arXiv:2310.11441, 2023.

[12] Y. Hu, W. Shi, X. Fu et al. Visual Sketchpad: Sketching as a visual chain of thought for multimodal language models. arXiv:2406.09403, 2024.

[13] S. Zhao, H. Zhang, S. Lin et al. PyVision: Agentic vision with dynamic tooling. arXiv:2507.07998, 2025.

[14] Q. Xia, S. Cheng, J. Bao et al. Found but not read: When extracted text closes the retrieval-reading gap in document vision-language models. arXiv:2610.02880, 2026.

[15] Y. Du, M. Tian, S. Ronanki et al. Context length alone hurts LLM performance despite perfect retrieval. arXiv:2510.05381, 2025.

[16] J. Yang, C. E. Jimenez, A. Wettig et al. SWE-agent: Agent-computer interfaces enable automated software engineering. arXiv:2405.15793, 2024.

[17] M. M. Liu, D. Garcia, F. Parllaku et al. ToolScope: Enhancing LLM agent tool use through tool merging and context-aware filtering. arXiv:2510.20036, 2025.

[18] Anthropic. Writing effective tools for agents (2025); Demystifying evals for AI agents (2026). Engineering blog, https://www.anthropic.com/engineering.

[19] J. Huang, X. Chen, S. Mishra et al. Large language models cannot self-correct reasoning yet. ICLR, arXiv:2310.01798, 2024.

[20] R. Kamoi, Y. Zhang, N. Zhang et al. When can LLMs actually correct their own mistakes? A critical survey of self-correction of LLMs. arXiv:2406.01297, 2024.

[21] K. Tsui. Self-Correction Bench: Uncovering and addressing the self-correction blind spot in large language models. arXiv:2507.02778, 2025.

[22] K.-Y. Chen, F.-Y. Su, S.-Y. Lin et al. The self-correction illusion: Role relabeling gates explicit error flagging in large language models. arXiv:2606.05976, 2026.

[23] A. Mehta and A. Datta. When agents commit too soon: Diagnosing premature commitment in LLM agents. arXiv:2606.22936, 2026.

[24] B. H. Bjarnason, A. Silva and M. Monperrus. On randomness in agentic evals. arXiv:2602.07150, 2026.

[25] S. Shankar, J. D. Zamfirescu-Pereira, B. Hartmann et al. Who validates the validators? Aligning LLM-assisted evaluation of LLM outputs with human preferences. arXiv:2404.12272, 2024.

[26] L. Zheng, W.-L. Chiang, Y. Sheng et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. arXiv:2306.05685, 2023.