BlueprintAgent: Constraint-Triggered Targeted Revisits for Simulation-Ready Generation from Scanned Structural Blueprints

Zhouyuan Xu, Chen Yang, Linhao Wang,
Jiansheng Fan, Chen Wang
Tsinghua University, Beijing 100084, China
Correspondence: fanjsh@tsinghua.edu.cn, chwang@tsinghua.edu.cn

Abstract

Converting in-service reinforced-concrete (RC) building blueprints into simulation-ready models—structured frame representations that support deterministic FEM export and qualified-engineer review—underpins safety assessment and seismic retrofit, but the process remains manual. Direct prompting of a multimodal large language model (MLLM) over a scanned sheet is unreliable: outputs often violate engineering constraints on beam–column support, span count, or 3D continuity. We present **BlueprintAgent (BPA)**, a constraint-triggered multimodal agent for simulation-ready frame extraction from scanned blueprints. BPA treats the MLLM as the primary reader and decision maker, with OCR and computer vision supplying localized evidence. Its central mechanism realizes engineering constraints as callable validators whose entity-level conflict reports trigger targeted MLLM revisits over the local region—an inference-time control distinct from fixed pipelines and free-form self-reflection. We evaluate BPA on 300 real scanned blueprint sheets from 20 anonymized RC frame projects, against five baselines and six ablations. BPA reaches a macro-averaged Beam F1 of **0.994**, against 0.301 for single-MLLM zero-shot and 0.820 for a fixed pipeline; removing MLLM-led axis adjudication collapses Beam and Column F1 on complex multi-sheet projects. For dense technical drawings, engineering constraints are best deployed as triggers for entity-level targeted revisits rather than as post-hoc output filters.

1 Introduction

Multimodal large language models are increasingly applied to dense technical documents such as scientific tables, financial filings, and engineering drawings. The challenge in these tasks lies not in recognizing text or graphics on a page, but in producing an externally checkable, structurally consistent representation that downstream programs can consume. Scanned structural blueprints are a prototypical case: safety assessment, seismic retrofit, and digital-twin construction all require structural models, yet many in-service buildings exist only as paper or scanned construction drawings. Engineers read these axes, beams, columns, sections, elevations, materials, and notes across many sheets, then build FEM models by hand.

Structural blueprints are not ordinary OCR targets. The text they contain encodes engineering semantics, not free prose. For example, the `4` in `KL8(4)` denotes a four-span frame beam; `250×600` becomes meaningful only when bound to a specific beam or column; an axis label is not merely a character but also defines a spatial coordinate in the simulatable 3D model. Detecting *where* text or line segments appear is insufficient for producing a computable structural model. The system must know which label belongs to which beam, which beams are supported by which columns, which columns extend continuously across floors, and which dimensions can enter the FEM geometry.

Single-shot whole-sheet prompting of current MLLMs fails in three recurring ways that are generic to dense multimodal document understanding. Whole-sheet resolution downsampling drops small text, secondary axis labels, and dense dimension annotations. Without external constraint signals, the model over-connects: every grid intersection becomes a physical support and the system fabricates beam–column connections that are not actually there. Even schema-valid JSON can violate domain geometric, annotation, or 3D-consistency rules. These failure modes motivate a domain constraint layer that cross-checks perception output and re-reads local evidence at conflict sites.

We present **BlueprintAgent (BPA)**, a constraint-triggered multimodal reading-and-modeling agent. A project is represented as a hierarchy of **sheets, regions, evidence records, and a simulatable 3D**

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 13304–13317
October 24-29, 2026 ©2026 Association for Computational Linguistics

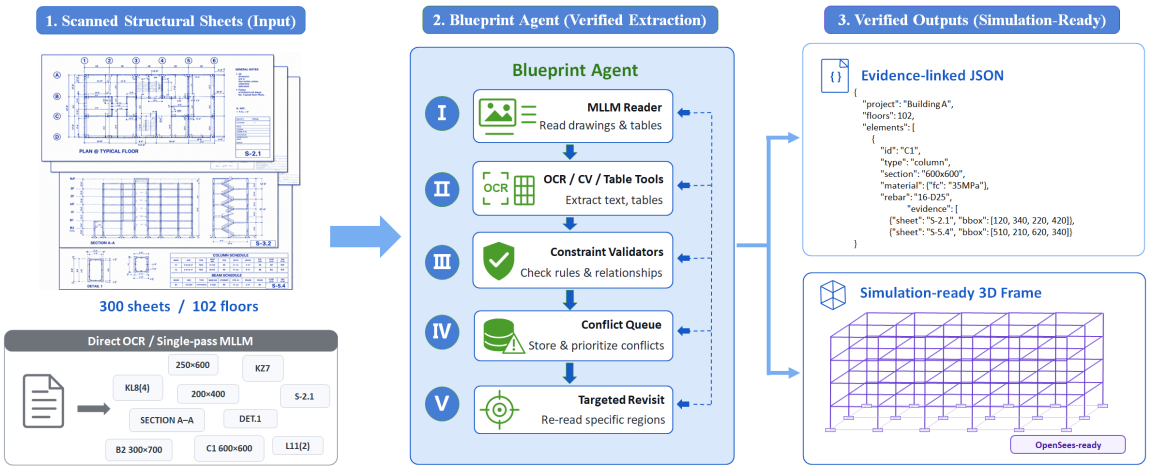

Figure 1: BPA end-to-end pipeline: scanned sheets (left) are processed by a verified extraction pipeline (center) into evidence-linked JSON and a simulation-ready 3D frame (right).

model. The MLLM acts as the primary perceiver and decision maker: it reads full sheets, main structural views, and local crops, and invokes OCR, CV-based axis probing, local re-OCR, axis-aware cropping, and table reading on demand. After each update, the system runs four families of constraints over the current simulatable 3D model: axis-grid, geometric-topological, annotation-consistency, and 3D-consistency. When a constraint fails, the system does not rewrite the model with hard rules; instead, it pushes a structured conflict item into a queue (for example, "the endpoint `1-B` of beam `KL4` lacks column support"). The MLLM then revisits the relevant local region and, grounded in the evidence, decides to revise, mark as uncertain, or accept it as a documented anomaly.

This paper focuses on frame-level reconstruction within **RC frame structures**: axis grids, columns, primary beams, sections, floors, materials, 3D frame consistency, and simulation-ready output. Shear-wall, steel, and masonry structures, as well as slab reinforcement, column longitudinal reinforcement, stirrup densification zones, and joint-core reinforcement, are not main-task metrics.

Our contributions are:

1. **Method.** We propose BlueprintAgent, which couples MLLM-based reading with OCR / CV / local-re-OCR / axis-aware-cropping tools and four families of engineering-constraint validators. The key design is to let constraint failures trigger entity-level targeted revisits rather than act only as post-hoc filters. The validator interface is model-independent by design, and targeted replacement checks preserve the mechanism's improvement direction across GPT-5.4, Claude 4.6 Opus, and Qwen 3.6 Plus.
2. **Task.** We formalize *generating simulation-ready structural models from scanned structural blueprints* as a multi-sheet technical-document understanding task. The input is the full set of scanned blueprint sheets per project; the output is evidence-linked structural JSON, a 3D frame, and OpenSees-ready FEM input.
3. **Evaluation.** We evaluate the design on **300 real scanned blueprint sheets covering 102 floors from 20 anonymized RC-frame projects**, reporting 5 baselines, 6 ablations, and 6 end-to-end metrics.

2 Related Work

Engineering drawing digitization. Research on engineering drawing digitization has explored object detection, OCR, image processing, and BIM reconstruction. Detector-based methods identify local targets such as beams, columns, and symbols (Zhao et al., 2020); BIM-oriented systems typically combine OCR, image processing, and hand-crafted rules to recover building components (Zhao et al., 2021). Recent MLLM-based work on engineering drawings attempts end-to-end parsing of annotations, GD&T information, and local fields (Khan et al., 2026a,b, 2025). Related work has also studied image-based structural analysis, capability-based evaluation of engineering agents, and the codification of expert knowledge for agent design (Altamirano-Muñiz, 2026; Molinari et al., 2026; Ulan uulu et al., 2026). These efforts demonstrate the feasibility of engineering drawing understanding but mostly focus on local elements or anno-

tation extraction within a single drawing; closure of a multi-sheet simulatable 3D model has not yet been addressed. We instead target a multi-sheet simulatable 3D model whose correctness depends on cross-sheet text parsing, beam–column support topology, floor alignment, and FEM-readiness.

Floor plan and CAD understanding. Floor plan and CAD understanding centers on walls, doors, windows, rooms, and furniture, with representative datasets such as CubiCasa5K (Kalervo et al., 2019) and FloorPlanCAD (Fan et al., 2021), floor-plan text detection and recognition (Schönfelder et al., 2024), and more recent MLLM-based floor-plan generation and editing (Qin et al., 2026). Structural blueprints differ from indoor floor plans: the core entities are axis grids, RC columns, frame beams, section labels, floor elevations, and material schedules; the target output is not a semantic segmentation map or room topology, but rather a load-bearing structural representation that engineering analysis can consume.

Visual document understanding and OCR. Visual document understanding models such as LayoutLMv3 (Huang et al., 2022), Donut (Kim et al., 2022), and Pix2Struct (Lee et al., 2023), alongside multi-page document QA (Ma et al., 2024), document-layout segmentation (Pfitzmann et al., 2022), logical document structuring (Li et al., 2024), and recent LLM-based document understanding and verification (Zhu et al., 2026; Qian et al., 2025), are isomorphic in output form to our "drawing → structured JSON" framing, but their structural boundary is defined by document elements rather than engineering entities under cross-sheet topological constraints. Scanned structural blueprints are harder: a single isolated number may be a primary-axis span, a sub-span, a local detailing dimension, or a table value, with semantics depending on its relation to the axis grid, the region, and surrounding components. BPA therefore consumes OCR and CV only as evidence under the active constraint context.

Tool-augmented and reflective agents. ReAct (Yao et al., 2023b), Toolformer (Schick et al., 2023), Reflexion (Shinn et al., 2023), and Self-Refine (Madaan et al., 2023) show that language models can interleave tool calls and improve outputs through feedback; Tree-of-Thoughts (Yao et al., 2023a) extends this to deliberative search, while Chain-of-Verification (Dhuliawala et al., 2024) shows that external verifier signals can be used to reduce hallucinations. Recent document-oriented agents (Jin et al., 2026; Li et al., 2024) and agent-evaluation benchmarks (Ye et al., 2024; Bogin et al., 2024) extend these ideas to multi-page documents, tool learning, and research repositories. BPA sits in this lineage but differs in the level of action and the trigger source: the verifier signal comes from a domain-level callable validator rather than a generic process reward model or free-form reflection, and constraints are injected into the MLLM's next planning step as structured JSON conflict items (containing entity ID, violated constraint, and supporting evidence), driving the MLLM to re-inspect the relevant local visual evidence.

Constrained and structured generation. Constrained generation in NLP includes lexically constrained decoding (Hokamp and Liu, 2017; Post and Vilar, 2018), grammar-constrained decoding (Geng et al., 2023), and neuro-symbolic decoding (Lu et al., 2022). Planning–verification loops have also been applied to factuality and logical consistency checking (Dhuliawala et al., 2024). BPA shares with these efforts the idea of *constraints participating in generation*, but operates at a different level: token-level constrained decoding regulates a single text generation pass, and natural-language neuro-symbolic methods act on textual inference; BPA consumes constraint-failure signals to re-acquire visual evidence, re-extract entities, and update the simulatable 3D model.

Feature comparison with prior work. Appendix F contrasts BPA with representative verification, reflection, and tool-augmented agents along four key design dimensions (Table 8).

Positioning of this work. This work sits at the intersection of engineering drawing digitization, visual document understanding, tool-augmented multimodal agents, and constraint-aware structured generation. Our contribution is not a new OCR model, detector, or FEM solver, but a reading-perception design built on an MLLM for dense technical drawings: it converts domain constraints into auditable triggers for multimodal evidence collection and entity revision.

3 Task and Data

Input unit. The input unit is **a complete set of scanned structural blueprints for one RC**

frame project: a project $S = \{s_1, \ldots, s_n\}$ typically contains from a few to several dozen sheets, covering beam, column, and slab plan views, title blocks, material notes, story-height tables, and detail views. The agent aggregates cross-sheet evidence and makes structural decisions over it. Each sheet typically holds one or two primary structural views (e.g., beam, column, or slab plans; elevations; sections), plus auxiliary regions such as joint or stairway details. A sheet is further partitioned into multiple regions $s_i \rightarrow \{r_i^{\text{main}}, r_i^{\text{table}}, r_i^{\text{title}}, r_i^{\text{note}}, r_i^{\text{detail}}, \ldots\}$. The system labels each region with a role but tolerates label errors; the role only governs how the region's evidence enters the simulatable 3D model.

Output. The canonical output is an evidence-linked structured JSON state H, comprising: an axis grid (labels, ordering, span lengths, grid intersections); floors (labels, elevations, story heights, corresponding sheets); columns (object ID, axis-grid position, section, floor, status, confidence, evidence references); primary beams (beam identifier, node sequence, span count, section, orientation, status, confidence, evidence references); materials (concrete grade, rebar grade, or other material specifications); and validation status (constraint check results, uncertain items, accepted anomalies). The MLLM creates and revises H from localized evidence. Once the key geometric and material fields are complete, a deterministic exporter generates an OpenSees `.tcl` file, and the 3D frame visualizes the same state.

Scope. This paper addresses frame-level structural reconstruction only: axes, columns, primary beams, sections, floors, materials, and 3D frame consistency. Fine-grained reinforcement (slab positive and negative reinforcement, column longitudinal reinforcement, stirrup densification zones, joint-core reinforcement) does not enter the main experimental metrics. Such reinforcement may be recorded as `non-core region` content or auxiliary evidence, but it is not used to evaluate simulation-ready frame recovery.

Dataset. The evaluation set uses **sheet** as the primary unit of count: **300 real scanned structural blueprint sheets** covering **102 floors**, drawn from partner-authorized structural archives for 20 anonymized RC frame projects, spanning beam, column, and slab plan views, foundation plans, elevations, sections, title blocks, material notes, story-height tables, and detail views. GPT-5.4 is used for inference without parameter fine-tuning, and the 300 sheets form the evaluation corpus. Difficulty tiers are split into simple, medium, and complex based on the number of floors, per-project sheet count, axis-grid density, and scan quality.

Ground truth. Complete project-level ground-truth JSON and OpenSees `.tcl` models were manually annotated from the full project sheet sets and reviewed entity by entity under the structural annotation protocol.

Data availability. The public academic release accompanying this paper contains the 300 anonymized blueprint sheets, project-level ground-truth JSON, manually annotated OpenSees `.tcl` models, a project manifest, the ground-truth schema and annotation protocol, evaluation code, and a datasheet.

Terminology. Throughout the paper we use *sheet* (one scanned blueprint), *region* (a semantic region inside a sheet), *entity* (a structural object: axis / column / beam / floor / material), *evidence* E (a localized OCR / CV / MLLM observation), *conflict* $q \in Q$ (a validator violation report), and *simulatable 3D model* H (the project-level evidence-linked structural representation). A *validator* is a callable constraint checker $v : H \rightarrow \text{List[Conflict]}$; a *targeted revisit* is the MLLM action of revisiting relevant local evidence and revising an entity, triggered by a validator conflict signal. `KL` / `KZ` are the frame-beam / frame-column labeling prefixes in Chinese structural drawings (e.g., `KL8(4)` = frame beam No. 8, four-span); the `axis_px_span_ratio` gate (Appendix E) checks axis-span pixel ratios against annotated span ratios within $\pm 15\%$ and corresponds to ablation A5.

4 Method

Overview. BPA is a tool-augmented multimodal agent (Figure 1; full pseudocode in Appendix I). It builds a project-level evidence pool through OCR caching and sheet–region analysis, then sequentially handles the core entity classes (axis grid, columns, beams, floors, materials) through axis-aware cropping, primary-element extraction, semantic validation, 3D consistency check, and FEM export, gradually closing the simulatable 3D model H. Within each stage, an *assess–plan–execute–interpret–update–validate–revisit* loop consumes validator feedback and performs targeted revisits

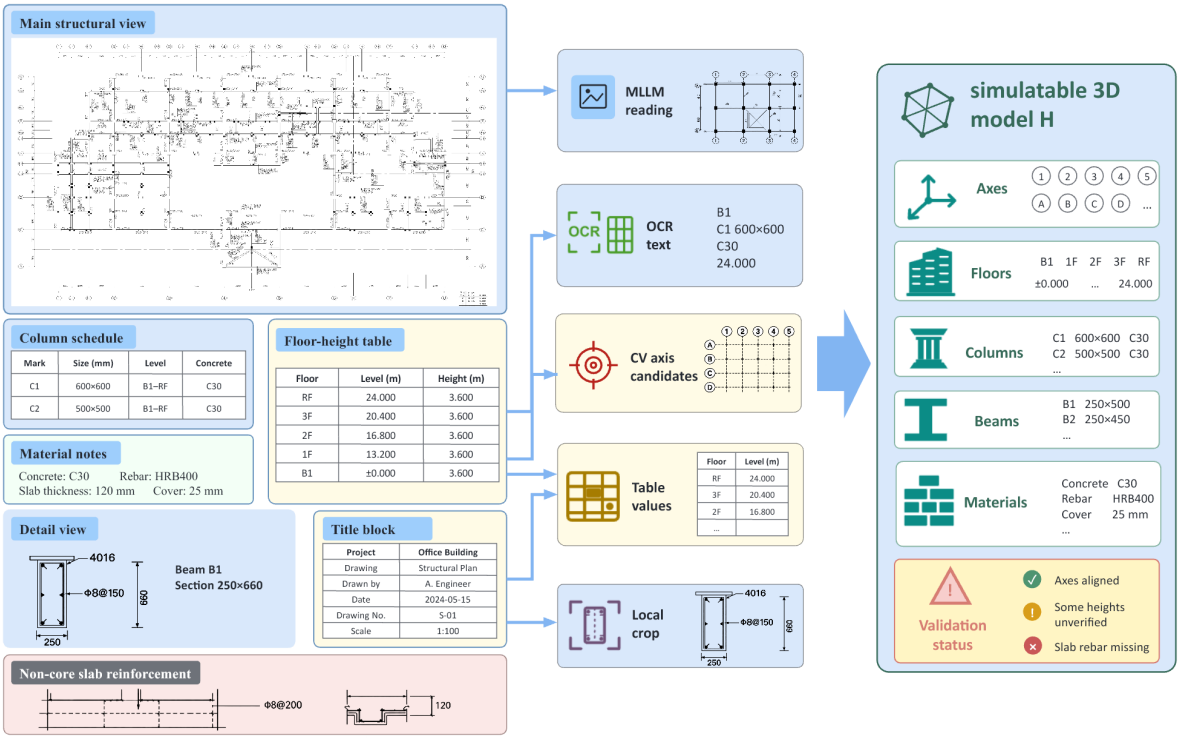

Figure 2: Sheet–region–evidence representation: a sheet (left) is partitioned into regions whose evidence (center) flows into the simulatable 3D model with a validation-status panel (right).

on low-confidence or conflicting entities until convergence or budget exhaustion.

Design choice: MLLM as the primary reader. Real scanned blueprints vary widely in line work, annotation style, resolution, and aging; the key problems are not local detection (where task-specialized detectors like YOLO or DETR excel) but cross-sheet aggregation, 3D topology alignment, and engineering-exception judgment. BPA therefore takes the MLLM as the primary reader and decision maker; OCR, CV, and table tools provide evidence. The MLLM's three roles map to specific ablation cells: perception to B1/A1/A2; reasoning to A6 (axis adjudication); decision to B4 (constraint feedback disabled during inference).

Sheet–region–evidence simulatable 3D model. BPA maintains the simulatable 3D model

$$H = \{\text{sheets}, \text{regions}, \text{axes}, \text{floors}, \\ \text{columns}, \text{beams}, \text{materials}, E, V, Q\},$$

where E is the set of evidence records, V the validation state, and Q the conflict queue. Each entity carries `status`, `confidence`, `evidence_refs`, and `provenance`. Structural reasoning is grounded in axis coordinates, node sequences, real dimension annotations, floors, and component attributes, not in pixel distances. Figure 2 shows how a sheet is partitioned into regions and how evidence flows into H.

Evidence tools. The tool library splits into *perceivers*—MLLM reading, multi-scale OCR cache, local re-OCR (multi-candidate readings inside an MLLM-specified box), and a CV axis probe (axis-label circles, lines, local symbols)—and *methods*—layout region detection, axis-aware cropping (preserving axis-label context), beam / column / text-binding candidate generation, the four semantic validators below, and an FEM exporter to OpenSees-ready `.tcl`.

Axis-grid-aware spatial representation. The axis grid is the spatial skeleton of the simulatable 3D model. Columns and beams are represented by grid intersections, axis labels, and node sequences, not by pixel coordinates. Axis-label adjudication follows two principles. *MLLM first:* when an axis label is visually readable, the MLLM reads and adjudicates it directly. *On-demand evidence:* when an axis label, secondary axis, or span length is unclear, the agent requests local OCR, CV axis candidates, or a region crop. As a design-time safety gate, the system additionally maintains an `axis_px_span_ratio` sanity check, which com-

pares the pixel-distance ratio between adjacent axis spans against the annotated span ratio (tolerance $\pm 15\%$); a significant deviation flags an axis-grid misalignment and triggers a targeted revisit. The gate targets a low-frequency but high-impact failure mode (axis-grid misalignment) and its ablation corresponds to A5.

Targeted revisit mechanism. The core mechanism of BPA is **validator-triggered targeted revisit**, an inference-time control mechanism in which domain-level validators emit structured conflict signals that gate entity-level multimodal revisitation. When a validator detects a conflict, the system injects the structured conflict item $q \in Q$ as evidence into the next `plan` call, letting the MLLM revisit the relevant region with the specific conflict in hand. This loop resembles ReAct's thought-action cycle, Reflexion's self-reflection signal, and Self-Refine's critique-revise. Its **trigger source** is an external schema-level constraint rather than internal natural-language reflection, distinguishing verification-driven from self-reflection-driven agents (see Section 2 and Table 8). BPA converts errors into auditable conflict records and steers the MLLM back to the corresponding visual evidence. Figure 3 illustrates the inner loop with a representative conflict record.

Constraint validators. BPA exposes four families of validators. C_{axis} (axis-grid constraints) checks axis-label legality, ordering, span-length evidence, secondary axes, and cross-floor axis-grid consistency; it is a spatial prerequisite and is not removed as an isolated ablation. C_{geo} (geometric-topological constraints) checks whether columns sit at grid intersections, whether primary beams run along grid lines, whether beam endpoints rest on columns or legitimate supports, and whether continuous beams pass through intermediate supports. C_{anno} (annotation-consistency constraints) checks whether the span count in `KL8(4)` is consistent with topology, whether `250×600` is bound to the correct beam or column, and whether KZ identifiers can be matched to the column schedule or column-annotation evidence. C_{3D} (3D consistency constraints) checks column continuity across floors, beam support, floor-elevation monotonicity, and component attribute completeness.

Validators only emit lists of violations; corrective actions are taken by the MLLM during the revisit loop. Formally, each $v \in C$ is a mapping $v : H \to \text{List[Conflict]}$, and $\textsc{validate}(H, C) = \bigcup_{v \in C} v(H)$. The current implementation defines 12 conflict types across C_{geo}, C_{anno}, and C_{3D} (C_{axis} is a spatial prerequisite and is not exposed as a separate ablation; full type list in the supplementary material). Load constraints and code-compliance constraints (minimum reinforcement ratios, seismic category) are left to future work.

Conflict handling. When a constraint fails, the validator writes a structured record into Q with fields `type` / `entity` / `violated` / `evidence` / `priority` / `status` (an example record appears in Figure 3; numerical checks may add `observed` / `expected` / `severity`). In the revisit, the MLLM takes one of three actions: *revise* (re-read local evidence and modify the entity), *mark uncertain* (flag for human review), or *accept anomaly* (accept a legitimate exception, e.g., cantilever beam, intentionally terminated column). This avoids forcing brittle rules onto the many legitimate engineering exceptions that real structural drawings contain.

5 Experiments

5.1 Research questions

Our experiments answer three questions. **RQ1 (Effectiveness):** does treating engineering constraints as entity-level targeted-revisit triggers improve end-to-end structural extraction? **RQ2 (Mechanism attribution):** which components are most critical on complex multi-sheet projects (cross-sheet aggregation A3, C_{3D} A4, the `axis_px_span_ratio` gate A5, MLLM-led axis adjudication A6)? **RQ3 (Bounds):** where do BPA's gains over pure-OCR rules, single-MLLM zero-shot, and the fixed pipeline come from?

5.2 Baselines

Five systems are compared, all MLLM-driven ones sharing the same backbone so that gains are attributable to framework design. **B1: Pure-OCR rules**—only OCR and hand-written rules; no MLLM. **B2: Single-MLLM zero-shot**—feeds a full sheet to the MLLM for direct structural JSON; no tools, constraints, or revisits. **B3: Fixed pipeline**—invokes OCR, CV, layout, and candidate-generation tools in a fixed order; no agent decision making, engineering-constraint validation, or conflict queue. **B4: Agent without constraint feedback**—retains the agent loop, tool calls, and output contract, but does not feed conflicts from C_{geo}, C_{anno}, or C_{3D} back into inference

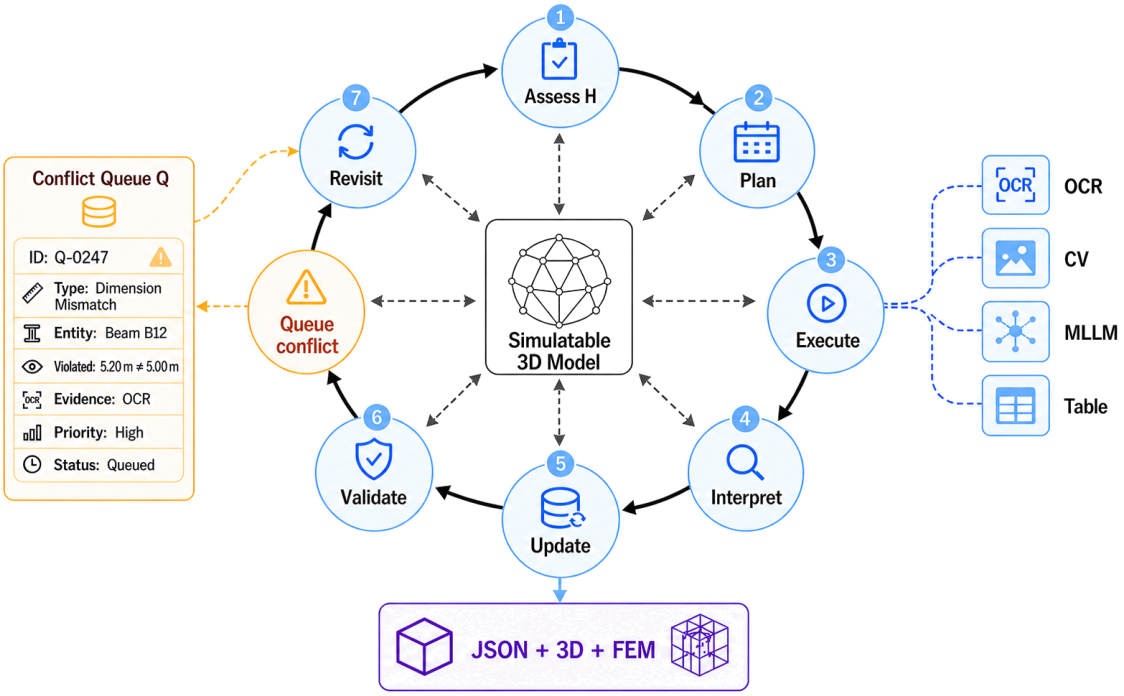

Figure 3: Targeted revisit loop: a seven-step inner loop consumes evidence from four tools and uses a structured conflict record (left) to revise the simulatable 3D model.

to trigger revisits. **B5: Full BlueprintAgent**—the complete system including the four constraint-validator families, the conflict queue, and targeted revisits.

5.3 Ablations

Six ablations each remove one component from B5. **A1 (no axis-aware cropping):** naive bounding-box crops only. **A2 (no axis-label repaste):** local crops lose global axis-label context. **A3 (no cross-sheet aggregation):** cross-sheet structures merged only by naive ID matching. **A4 (no C_{3D} closure):** removes the cross-floor 3D-closure stage from B5. **A5 (no `axis_px_span_ratio` gate):** removes the consistency check between axis-span pixel ratios and annotated span lengths. **A6 (no MLLM-led axis adjudication):** accepts CV axis candidates directly, without MLLM re-checking.

5.4 Metrics

The evaluation suite contains six end-to-end metrics (full definitions in Appendix C): **Axis F1** (axis-label match + position IoU ≥ 0.85), **Col F1** (column ID match after grid alignment), **Beam F1** (beam ID match + endpoint axis multiset match), **Bind F1** (KL text content + target beam ID match), **JSON Validity** (output passes the schema), and **Structured-Model Completeness (SMC)**. The project-level SMC gate equals 1 iff C_{3D} validation is enabled, the normalized evaluator status is `OK`, the predicted column and primary-beam inventories are both nonempty, and each has at least one ground-truth match; otherwise it equals 0. We report the project-equal mean over all 20 projects. Every variant uses the same output contract and evaluator, except that A4 disables the C_{3D} gate by definition.

Reproducibility setup. Each main-table cell reports one complete run. Except for deterministic B1, all systems call GPT-5.4 at temperature 0.2 with a fixed system prompt; per-cell prompts, runtime configurations, and metric JSON files are provided in the supplementary material. Run-to-run stability for the mechanism-bearing B4–B5 contrast and controlled backbone replacements are evaluated on a prespecified stratified five-project subset using project-equal aggregation. The main tables use the 20-project evaluation.

Metric behavior across variants. Axis F1, Col F1, Bind F1, JSON Validity, and SMC reach 1.000 for B5 while measuring distinct properties: axis identity and position, post-aggregation column inventory, KL-to-beam binding, schema conformance, and protocol-level structured-model completion. Each metric decreases in at least one base-

Method	Axis	Col	Beam	Bind	JSON	SMC
B1: OCR rules	0.000	0.000	0.000	0.000	1.000	0.000
B2: MLLM zero-shot	0.987	0.537	0.301	0.327	1.000	0.350
B3: Fixed pipeline	0.991	0.920	0.820	0.868	1.000	1.000
B4: no C feedback	0.993	0.959	0.768	0.873	1.000	1.000
B5: Full BPA	**1.000**	**1.000**	**0.994**	**1.000**	**1.000**	**1.000**

Table 1: Main results (300 sheets / 102 floors; single run; macro average over 20 projects). Column headers abbreviate Axis F1, Col F1, Beam F1, KL Bind F1, JSON Schema Validity, and Structured-Model Completeness. B4 suppresses $C_{\text{geo}}/C_{\text{anno}}/C_{\text{3D}}$ feedback during inference while retaining the common output contract and evaluator.

Variant	Axis	Col	Beam	SMC
B5 Full BPA	1.000	1.000	0.994	1.000
A1: no crop	1.000	0.838	0.674	0.900
A2: no repaste	1.000	0.827	0.896	0.850
A3: no x-sheet aggr.	0.981	0.516	0.564	0.850
A4: no C_{3D} closure	1.000	0.648	0.551	0.000
A5: no span-ratio gate	1.000	0.831	0.828	0.850
A6: no axis adjud.	0.892	0.614	0.369	0.650

Table 2: Ablation results. A1 = no axis-aware crop; A2 = no axis-label repaste; A3 = no cross-sheet aggregation; A4 = no C_{3D} closure; A5 = no `axis_px_span_ratio` gate; A6 = no MLLM-led axis adjudication.

line or ablation. The B3, B4, and A6 cells identify the mechanism-bearing gaps, while the remaining 0.6% Beam-F1 error for B5 is concentrated in the perception and dense-binding cases reported in Appendix H.

5.5 Main results

Table 1 presents the main results. **Perception-only systems cannot recover structural topology.** B1's Axis F1 and Beam F1 are both 0.000 (schema-valid but empty JSON); B2's Axis F1 reaches 0.987 but its Beam F1 is 0.301 and Col F1 is 0.537. Whole-sheet prompting can identify part of the axis grid on simple drawings but cannot jointly verify span count, supporting columns, and beam topology in one inference pass. **The fixed pipeline improves perception but cannot self-correct.** B3 adds OCR, CV, and candidate generation, raising Beam F1 to 0.820—still well below B5's 0.994: once an early axis-label, candidate, or binding error occurs, a fixed pipeline has no mechanism to localize the error back to the visual evidence and re-read it.

Constraint-triggered revisits drive the main gain. B4 retains the agent loop and tools but suppresses engineering-constraint feedback during inference; its Beam F1 is close to B3, whereas B5 improves Col F1, Beam F1, and Bind F1. JSON Validity and SMC are already saturated for B4. The B4–B5 contrast therefore localizes the gain to integrating structured conflicts into the loop as entity-level revisit triggers.

5.6 Ablation study

Table 2 reports the six ablations. A6 (no MLLM-led axis adjudication) has the largest extraction decline: macro Beam F1 drops to 0.369 and Col F1 to 0.614, with the effect intensifying on multi-floor multi-sheet composite-grid projects where an erroneous axis grid corrupts both the column inventory and beam topology. A3 (no cross-sheet aggregation) most strongly affects column deduplication: the same physical column is counted across plan views, inflating the inventory by 2.0–3.0$\times$ and reducing Col F1 to 0.516. A5 (no `axis_px_span_ratio` gate) yields Beam F1 of 0.828 and Col F1 of 0.831, consistent with its focus on infrequent axis-grid misalignment. A4 retains planar extraction metrics but disables C_{3D} validation, so the protocol assigns SMC 0 to every project.

5.7 Targeted robustness and isolation checks

Table 3 summarizes targeted checks on a pre-specified stratified subset containing all three complex projects and one medium and one simple project. For GPT-5.4, three repeats per B4/B5 configuration give project-equal Beam F1 of 0.7894 ± 0.0121 for B4 and 0.9609 ± 0.0074 for B5, where the dispersion is pooled within-project run SD; all 15 project-by-repeat-index contrasts favor B5. Applying the same validators only after B4 terminates yields B6 Beam precision/recall/F1 of $0.8996/0.8069/0.8505$, compared with $0.9518/0.9704/0.9609$ for B5. Thus B5 exceeds B6 by 0.1104 Beam F1, and 23 of 32 conflicts classified as requiring localized visual rereading are resolved after rereading. Controlled replacements with Claude 4.6 Opus and Qwen 3.6 Plus preserve the B5-over-B4 direction on every evaluated project. Together, these checks support both the run stability of the central contrast and the contribution of evidence reacquisition beyond post-hoc constraint processing.

5.8 Case studies and counterexamples

Figure 4 presents three representative modeling cases; Appendix G records their diagnostic mechanisms. The A6 counterexample admits *ghost axes* when CV candidates bypass MLLM re-checking,

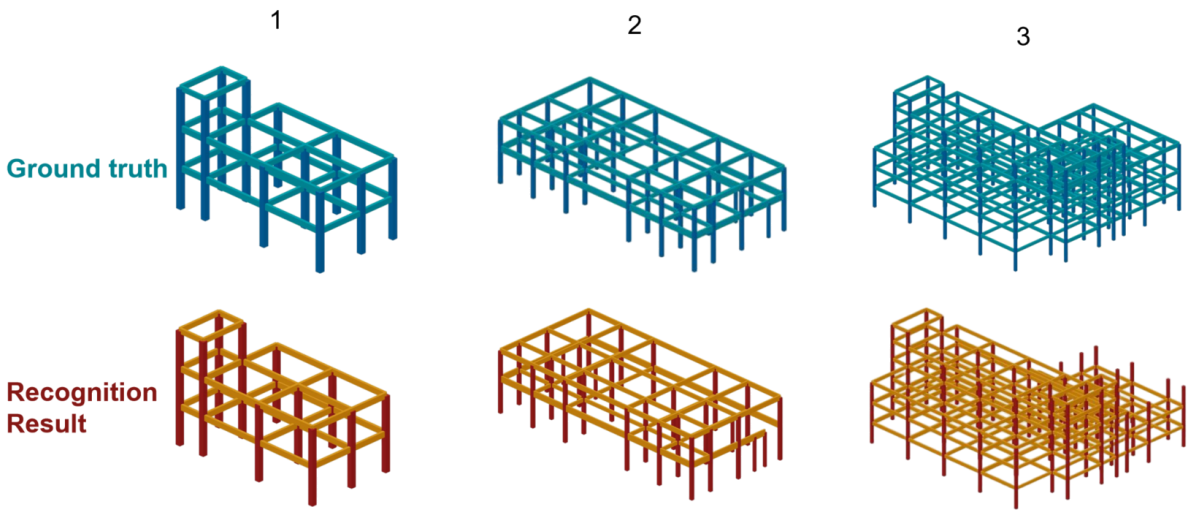

Figure 4: Representative 3D frame modeling results for three projects. Each column pairs the ground-truth model (top) with the corresponding BlueprintAgent output (bottom) under the same orthographic camera; colors distinguish columns and beams.

Backbone	B4	B6	B5
GPT-5.4	0.7894±0.0121	0.8505	0.9609±0.0074
Claude 4.6 Opus	0.7846	–	0.9062
Qwen 3.6 Plus	0.7787	–	0.8829

Table 3: Project-equal Beam F1 on the targeted five-project subset. GPT-5.4 reports three repeats per B4/B5 configuration; B6 is deterministically derived from the corresponding B4 states. Replacement-backbone checks use one B4/B5 run per project.

corrupting the column inventory and beam topology. A second case passes C_{3D} and runs in OpenSees but yields anomalous top-floor displacement, reinforcing qualified-engineer review before engineering use.

5.9 Findings and analysis

The small B3–B4 and large B4–B5 gaps identify structured conflicts consumed as entity-level revisit triggers, rather than looping alone, as the operative mechanism. B6 confirms that post-hoc constraints do not recover the recall gained through localized rereading. The consistent B5-over-B4 direction across three tested backbones supports interface portability, with larger gains on complex projects (Appendix D).

5.10 Adjudicated consensus error profile

Two structural engineers independently categorized residual B5 errors, with disagreements arbitrated by a third (Appendix H). The adjudicated profile is 45% perception, 25% topology, 20% binding, and 10% completeness: perception dominates, while the remaining categories concentrate in dense regions or structured-model closure.

6 Conclusion

BlueprintAgent, the system introduced in this paper, treats engineering constraints not as a final-stage filter but as triggers that pull the MLLM back to the specific local region where its first read went wrong. The hierarchy is sheet → region → evidence → simulatable 3D model; the validators are callable, schema-level, and entity-scoped. On 300 real scanned structural blueprint sheets covering 102 floors, this pattern consistently improves beam topology, column identification, and text-binding metrics over single-MLLM zero-shot and over a fixed OCR-plus-CV pipeline that lacks the revisit mechanism.

Limitations

Sample scope. Our main evaluation uses 300 sheets covering 102 floors from 20 anonymized RC-frame projects. The empirical scope is RC frames; shear-wall, steel, masonry, and complex hybrid structures remain outside the evaluation.

Annotation depth and scale trade-off. Each core entity (axis, column, beam, KL label, floor, section) requires structural-engineer-level annotation and review. The evaluation prioritizes structural-semantic annotation depth across 300 sheets. Scaling to a larger drawing library requires

corresponding engineering-annotation effort.

Backbone scope. The main comparison fixes GPT-5.4 as the perception-and-decision core to avoid a model confound. Controlled B4/B5 replacements with Claude 4.6 Opus and Qwen 3.6 Plus preserve the same improvement direction on the targeted five-project subset. Our empirical portability claim is limited to these three tested backbones.

Evaluation repetition. Main results and ablations contain one complete run per cell (except deterministic B1). The targeted stability check therefore repeats the mechanism-bearing B4–B5 comparison three times per configuration on a pre-specified stratified five-project subset. This check establishes stability for the central contrast, while the complete main matrix remains a single-run evaluation.

Single FEM solver. We export to OpenSees and verify that linear-elastic static analysis runs. Dynamic analysis, push-over, time-history, and nonlinear analyses require additional structural parameters and are outside this version's output scope.

Experimental coverage gaps. A fine-tuned detector baseline is not included. Such a comparison would isolate *strong visual detector + post-processing* from *MLLM agent + constraint-driven revisits* and is a direct extension of the current evaluation.

Cost and latency. Targeted revisits raise the tool-call count and token cost on harder sheets. Wall-clock and token-cost comparisons are not included. Selective revisit triggering, OCR/CV caching, and policy distillation are practical directions for reducing this overhead.

Ethical considerations

Data provenance and anonymization. The 300 sheets used for evaluation come from real structural construction archives released under partner permission. Project names, client identifiers, designer signatures, addresses, and similar information were removed before the system processed any sheet. The data contains no personally identifiable information.

Engineering safety and human oversight. BPA produces simulation-ready frame models that help structural engineers start their modeling and review workflows; it does not replace qualified-engineer judgment. Any application involving load-bearing modifications, seismic retrofit, code-compliance checks, or safety-critical decisions must be reviewed and signed off by an engineer-of-record.

Uncertainty disclosure. The `confidence`, `status`, `evidence_refs`, and `unresolved_items` fields in the output JSON are designed to reduce false confidence. Downstream systems should display these fields explicitly rather than presenting only the final 3D model or FEM file.

MLLM API dependence and reproducibility. When a commercial MLLM API is used, reproducibility is affected by model version and service-side policy. The supplementary material provides the model version, prompts, tool configuration, intermediate simulatable 3D model, and evidence trace, so that results can be re-evaluated on different backbones.

AI-assistant disclosure. The authors used LLMs to assist with English phrasing, code drafting, and literature screening; all technical claims, experimental numbers, and citations were independently verified by the authors.

Acknowledgments

This work was supported by the Undergraduate "Qiyan" Program of Beijing Natural Science Foundation (Grant No. QY26402) and the Beijing Nova Program (Grant No. 202604841290).

References

Emilio Fernando Altamirano-Muñiz. 2026. Image-based structural analysis using computer vision and LLMs: PhotoBeamSolver. ArXiv:2603.21432.

Ben Bogin, Kejuan Yang, Shashank Gupta, Kyle Richardson, Erin Bransom, Peter Clark, Ashish Sabharwal, and Tushar Khot. 2024. SUPER: Evaluating agents on setting up and executing tasks from research repositories. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 12622–12645. Outstanding Paper Award.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2024. Chain-of-Verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578.

Zhiwen Fan, Lingjie Zhu, Honghua Li, Xiaohao Chen, Siyu Zhu, and Ping Tan. 2021. FloorPlanCAD: A

large-scale CAD drawing dataset for panoptic symbol spotting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10108–10117.

Saibo Geng, Martin Josifoski, Maxime Peyrard, and Robert West. 2023. Grammar-constrained decoding for structured NLP tasks without finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 10932–10952.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. LayoutLMv3: Pre-training for document AI with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia (MM)*, pages 4083–4091.

Yiqiao Jin, Rachneet Kaur, Zhen Zeng, Sumitra Ganesh, and Srijan Kumar. 2026. SlideAgent: Hierarchical agentic framework for multi-page visual document understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14858–14881.

Ahti Kalervo, Juha Ylioinas, Markus Häikiö, Antti Karhu, and Juho Kannala. 2019. CubiCasa5K: A dataset and an improved multi-task model for floorplan image analysis. In *Image Analysis (SCIA 2019)*, volume 11482 of *Lecture Notes in Computer Science*, pages 28–40. Springer.

Muhammad Tayyab Khan, Lequn Chen, Ye Han Ng, Wenhe Feng, Nicholas Yew Jin Tan, and Seung Ki Moon. 2026a. Fine-tuning vision-language model for automated engineering drawing information extraction. In *Proceedings of the 9th International Conference on Innovation in Artificial Intelligence (ICIAI)*, pages 411–423. Springer.

Muhammad Tayyab Khan, Lequn Chen, Zane Yong, Jun Ming Tan, Wenhe Feng, and Seung Ki Moon. 2026b. From drawings to decisions: A hybrid vision-language framework for parsing 2D engineering drawings into structured manufacturing knowledge. *Robotics and Computer-Integrated Manufacturing*, 99:103186.

Muhammad Tayyab Khan, Zane Yong, Lequn Chen, Wenhe Feng, Nicholas Yew Jin Tan, and Seung Ki Moon. 2025. A multi-stage hybrid framework for automated interpretation of multi-view engineering drawings using vision language model. ArXiv:2510.21862.

Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. OCR-free document understanding transformer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 498–517.

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2Struct: Screenshot parsing as pretraining for visual language understanding. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 18893–18912. PMLR.

Zichao Li, Shaojie He, Meng Liao, Xuanang Chen, Yaojie Lu, Hongyu Lin, Yanxiong Lu, Xianpei Han, and Le Sun. 2024. Seg2Act: Global context-aware action generation for document logical structuring. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 18077–18088.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2024. MMLongBench-Doc: Benchmarking long-context document understanding with visualizations. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track (Spotlight)*, volume 37, pages 95963–96010.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 46534–46594.

Gioele Molinari, Florian Felten, Soheyl Massoudi, and Mark Fuge. 2026. EngiAI: Capability-based evaluation of tool-connected LLM agents for engineering design. ArXiv:2605.19743.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter W. J. Staar. 2022. DocLayNet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 3743–3751.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324.

Kun Qian, Wenjie Li, Tianyu Sun, Wenhong Wang, and Wenhan Luo. 2025. DocRefine: An intelligent framework for scientific document understanding and content optimization based on multimodal large model agents. ArXiv:2508.07021.

Sizhong Qin, Ramon Elias Weber, and Xinzheng Lu. 2026. Tokenization allows multimodal large language models to understand, generate and edit architectural floor plans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10430–10440.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 68539–68551.

Phillip Schönfelder, Fynn Stebel, Nikos Andreou, and Markus König. 2024. Deep learning-based text detection and recognition on architectural floor plans. *Automation in Construction*, 157:105156.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 8634–8652.

Choro Ulan uulu, Mikhail Kulyabin, Iris Fuhrmann, Jan Joosten, Nuno Miguel Martins Pacheco, Filippos Petridis, Rebecca Johnson, Jan Bosch, and Helena Holmström Olsson. 2026. How to build AI agents by augmenting LLMs with codified human expert domain knowledge? a software engineering framework. ArXiv:2601.15153.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of Thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. ReAct: Synergizing reasoning and acting in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*. ArXiv:2210.03629.

Junjie Ye, Yilong Wu, Songyang Gao, Caishuang Huang, Sixian Li, Guanyu Li, Xiaoran Fan, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. RoTBench: A multi-level benchmark for evaluating the robustness of large language models in tool learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 313–333.

Yunfan Zhao, Xueyuan Deng, and Huahui Lai. 2020. A deep learning-based method to detect components from scanned structural drawings for reconstructing 3D models. *Applied Sciences*, 10(6):2066.

Yunfan Zhao, Xueyuan Deng, and Huahui Lai. 2021. Reconstructing BIM from 2D structural drawings for existing buildings. *Automation in Construction*, 128:103750.

Dawei Zhu, Rui Meng, Jiefeng Chen, Sujian Li, Tomas Pfister, and Jinsung Yoon. 2026. DocLens: A tool-augmented multi-agent framework for long visual document understanding. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26804–26829.

A Dataset statistics

Tier	Projects	Description
Simple	10	2–3 floors, regular grid
Medium	7	4–6 floors, mild interference
Complex	3	7+ floors, composite grid
Total	**20**	**300 sheets, 102 floors**

Table 4: Dataset statistics by difficulty tier: 300 sheets and 102 floors from 20 projects.

B OCR multi-scale protocol

The OCR subsystem returns a multi-scale candidate set with confidence scores for each bounding box specified by the MLLM. The MLLM then selects the most consistent reading under the surrounding axis grid, engineering-annotation conventions, and the current constraint context, thereby preventing OCR top-1 errors from propagating into the simulatable 3D model.

C Metric definitions

Axis F1. Matches when the predicted axis line is equivalent in label to the ground-truth axis line and position IoU ≥ 0.85.

Column F1. After axis-grid matching, matches when the predicted column sits at the same grid intersection as the ground-truth column and the column ID is equivalent.

Beam Topology F1. Matches when the beam ID matches and the endpoint axis multiset matches.

KL Text Binding F1. Matches when the normalized KL text content matches and the target beam ID matches.

JSON Schema Validity. The output JSON passes the evidence-linked JSON schema with zero errors.

Structured-Model Completeness (SMC). For project p, let $s_p = 1$ iff $C_{3\mathrm{D}}$ validation is enabled, the normalized evaluator status is `OK`, the predicted column and primary-beam inventories are both nonempty, and each has at least one ground-truth match; otherwise $s_p = 0$. We report $\frac{1}{20}\sum_{p=1}^{20} s_p$. Thus A4, which disables $C_{3\mathrm{D}}$ validation by construction, has $s_p = 0$ for every project.

D Per-tier results

To support the stratification observation in Section 5 (constraint contribution stratifies with complexity), Tables 5 and 6 report the macro-averaged Beam F1 and Col F1 for B5 and the key ablations on each difficulty tier (simple / medium / complex; project counts 10 / 7 / 3, see Table 4).

Variant	Simple (10)	Medium (7)	Complex (3)	Overall (20)
B5 Full BPA	1.000	1.000	0.960	0.994
A3: no x-sheet aggr.	0.662	0.541	0.291	0.564
A4: no $C_{3\mathrm{D}}$ closure	0.653	0.507	0.314	0.551
A6: no axis adjud.	0.571	0.217	0.050	0.369

Table 5: Per-tier Beam F1.

Variant	Simple (10)	Medium (7)	Complex (3)	Overall (20)
B5 Full BPA	1.000	1.000	1.000	1.000
A3: no x-sheet aggr.	0.689	0.402	0.205	0.516
A4: no $C_{3\mathrm{D}}$ closure	0.825	0.552	0.282	0.648
A6: no axis adjud.	0.849	0.497	0.104	0.614

Table 6: Per-tier Col F1.

E Terminology

Table 7 lists the paper-specific terms together with their definitions for quick reference.

Term	Definition
sheet	A single scanned structural blueprint
region	A semantic region within a sheet
entity	A structural object in H (axis / column / beam / floor / material)
evidence (E)	A localized OCR / CV / MLLM observation with pixel provenance
conflict (q)	A validator's violation report on H
H	The project-level evidence-linked structural representation
validator	A callable checker $v : H \to \mathrm{List}[\mathrm{Conflict}]$
KL / KZ	Frame-beam / frame-column labeling prefixes (e.g., `KL8(4)` = frame beam 8, four-span)
`axis_px_span_ratio`	Consistency check between axis-span pixel ratios and annotated span ratios ($\pm 15\%$); ablation A5
axis adjudication	The MLLM's final decision on axis labels, integrating OCR / CV evidence
targeted revisit	MLLM revisiting local evidence and revising an entity, triggered by a validator conflict
SMC	Project-level completion gate for nonempty matched column/beam inventories under $C_{3\mathrm{D}}$ validation

Table 7: Terminology used throughout the paper.

F Feature comparison with prior agents

Table 8 contrasts BPA with representative verification, reflection, and tool-augmented agents along

four design dimensions.

Work	Trigger	Signal	Granularity	Sym.
ReAct	model	NL thought	task	no
Reflexion	model	NL self-critique	episode	no
Self-Refine	model	NL critique	response	no
Self-Consist.	sampling	majority vote	answer	no
CRITIC	tool	tool output	response	part.
PRM	external	step score	step	no
BPA	**schema**	**JSON conf.**	**entity**	**yes**

Table 8: Feature comparison with prior work. BPA differs in trigger source (external schema validator), signal type (structured JSON conflict), revisit granularity (entity-level), and explicit evidence-grounded revisions.

G Representative cases

Table 9 summarizes three representative cases by failure mode, triggering constraint, and revisit outcome; Section 5.8 discusses Case 2 (the A6 counterexample) in detail.

Case	Failure mode	Trigger	Outcome
1	CV axis misreport from a dimension annotation	C_{axis} span-ratio mismatch	MLLM corrects misreport; entities consistent
2 (A6)	CV accepted → ghost axis	none (gate off)	Errors propagate; F1 collapses
3	C_{3D} passes; OpenSees runs with anomalous top-floor displacement	semantic constraints exhausted	Qualified-engineer review

Table 9: Representative cases by failure mode, triggering constraint, and revisit outcome.

H Adjudicated consensus error profile

Table 10 reports the four-category adjudicated consensus breakdown of residual B5 errors over the 300-sheet evaluation; annotations were carried out independently by two structural engineers, with disagreements arbitrated by a third.

Category	Share	Trigger
Perception	45%	Small font, low contrast, rotated, or aged scans
Topology	25%	Adjacent spans share a column but differ in section; beam-segment merge errors
Binding	20%	Densely placed KL labels may bind to a neighboring beam
Completeness	10%	Critical notes lie on sheets outside the current working set

Table 10: Adjudicated consensus error profile for B5 over the 300-sheet evaluation.

I BPA main loop pseudocode

Algorithm 1 BPA main loop. Outer loop walks through five sequential stages; inner loop runs a seven-step assess-to-revisit cycle until conflicts are resolved or the stage budget is exhausted. Validators are aggregated as $\textsc{validate}(H, C) = \bigcup_{v \in C} v(H)$, each $v : H \to \text{List[Conflict]}$.

Require: project P, MLLM backbone M, constraints $C = \{C_{\text{axis}}, C_{\text{geo}}, C_{\text{anno}}, C_{\text{3D}}\}$
Ensure: simulatable 3D model H, simulation-ready file F

```
H ← INIT_H(P)
cache ← OCR_CACHE(P)
regions ← ANALYZE_REGIONS(P)
for stage ∈ {axes, columns, beams, validation, 3D-check} do
    repeat
        a ← M.ASSESS(H, stage)
        (r, t) ← M.PLAN(H, stage, a)
        o ← EXECUTE(t, r)
        e ← M.INTERPRET(o)
        H ← M.UPDATE(H, e)
        Q ← VALIDATE(H, C)
        if Q = ∅ then break
        end if
        H ← M.REVISIT(H, Q)
    until stage budget exhausted
end for
F ← SIM_EXPORT(H)
return H, F
```