

Human-Gated ASGs from BIM to Assembly

Abstract. Computational design makes it increasingly feasible to describe buildings as structured digital models, yet physical assembly still involves uncertainty that cannot be fully anticipated in advance, including pose deviation, contact mismatch, tolerance accumulation, and incomplete runtime information. This paper proposes Human-Gated Action Skill Graphs (HG-ASGs) as a contract-oriented translation layer from BIM-derived component data to accountable assembly execution. Instead of treating robotic execution as unrestricted automation, HG-ASGs compile each component into a canonical sequence of action primitives with explicit motion parameters, guard predicates, non-decisions, bounded micro-recovery rules, and human escalation conditions. In this formulation, the robot is authorized to act only within a validated decision envelope, while uncertain or out-of-contract situations are transferred to a mixed-reality gate and recorded as traceable events. We implement the framework in a Unity-based digital twin of an eight-actuator cable-driven parallel robot and evaluate it on 10 benchmark assembly scenarios with 38 components each under two uncertainty profiles representing high-precision prefabrication and lower-precision site conditions. Compared with an open-loop baseline, HG-ASG-Auto increases mean completion from 56.8% to 83.4% in the high-precision profile and from 31.6% to 73.7% in the low-precision profile, while reducing timeouts, collisions, unsafe contacts, and damage incidents, with only modest runtime overhead. The results show that contract-based decision governance can improve assembly robustness, safety, and accountability under uncertainty.

Keywords: Human-Gated Action Skill Graphs, Building Information Modeling, Construction Automation

1 Introduction

Computational design now makes it routine to specify buildings as richly structured digital models. Yet when these models meet physical assembly, outcomes remain shaped by factors that are difficult to predict or fully encode, including contact conditions, tolerance accumulation, temporary supports, tool access, and local material variability. In such settings, the challenge is not simply how to automate more, but how to allocate agency under uncertainty: which decisions should remain human, which can be delegated to machines, and how those delegations remain accountable when conditions exceed what the model can safely anticipate [1][2].

BIM-to-robot pipelines have shown that BIM can support robot task planning, simulation, and construction operations through geometry, semantics, and process information. Reviews identify task planning and intelligent assembly as core BIM-robotics application areas [3]. Recent systems have linked BIM to ROS for flexible construction planning [4], integrated 4D BIM with action-level robot task planning for wall-frame installation [5], and streamlined BIM-informed design-to-

manufacturing workflows for robotic wood framing [6]. However, BIM authoring and exchange workflows mainly organize geometry and lifecycle information rather than explicit runtime decision clauses for robotic execution. Considerable enrichment is still required to represent insertion constraints, bounded recovery, escalation conditions, and traceable downstream reasoning [7][8]. Reviews on BIM and IFC readiness likewise show persistent gaps in semantics, extractability, and interoperability, making transformation and toolchain preparation necessary [9][10]. As a result, decision-making under uncertainty is often deferred to ad hoc controller logic or tacit operator judgment, limiting transparency and accountability.

This paper proposes Human-Gated Action Skill Graphs (HG-ASGs) as a contract-oriented translation layer from BIM-derived component data to accountable execution. Rather than treating robot execution as the endpoint of automation, HG-ASGs frame execution as the enactment of a Decision Contract. The contract explicitly encodes permissible actions, non-decisions, bounded micro-recovery after verification failure, and the conditions under which uncertainty must trigger escalation to a human gate. The human gate is implemented through a mixed-reality (MR) interface that requests minimal structured confirmations at predefined decision points, turning intervention from an informal override into an explicit logged event. This design is informed by recent XR-robotics reviews and AR-enabled intervention studies in construction, which emphasize structured human supervision, task sharing, and recovery pathways in robotic fabrication workflows [11][12]. Fig.1 contrasts a conventional BIM-to-robot execution pipeline with the proposed HG-ASG pipeline, highlighting the shift from geometry transfer to governed runtime execution under uncertainty.

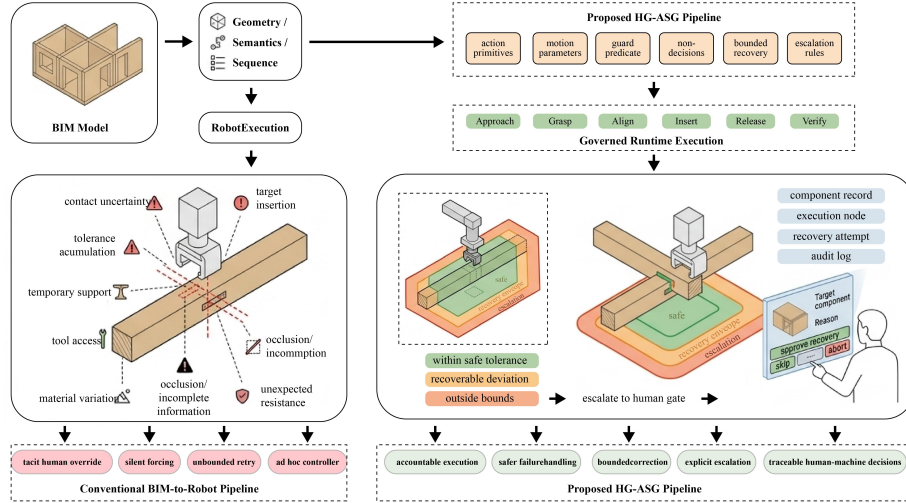


Fig. 1. ASG Comparison between a conventional BIM-to-robot pipeline and the proposed HG-ASG pipeline. Unlike conventional pipelines, which lack explicit runtime decision governance under uncertainty, HG-ASG introduces a decision-contract layer with bounded recovery, human-gated escalation, and traceable accountability.

Methodologically, HG-ASGs are compiled from BIM-native component records exported into a structured intermediate representation. Each component is mapped into a canonical sequence of approach, grasp, align, insert, release, and verify, together with contract clauses including motion parameters, guard predicates, and normative constraints that define non-decisions, such as near-vertical insertion and tilt limits during seating. When verify fails, the contract triggers bounded micro-recovery through an allowable corrective action set and a recovery budget; if recovery exceeds those bounds, the system escalates rather than continuing autonomously. In this sense, HG-ASGs operationalize recent work on meaningful human control and variable autonomy directly within an executable assembly representation rather than leaving these ideas at the level of supervisory policy alone [13][14].

To address generalisability beyond high-precision prefabricated timber, we evaluate HG-ASGs under two uncertainty regimes in a physics-based digital twin: a high-precision profile approximating prefabrication, and a low-precision profile approximating broader site conditions with larger and more frequent deviations. The goal is not to claim universal autonomy, but to show how contract parameters reconfigure what the robot is allowed to decide as construction conditions become less predictable.

The contributions are threefold:

- (1) a contract-oriented BIM-to-assembly translation that makes delegation explicit through non-decisions, bounded recovery, and human gates;
- (2) a runtime architecture coupling contract checking, MR gating, and provenance logging for end-to-end auditability; and
- (3) a comparative study showing how HG-ASGs shift failure handling from unbounded retries or silent forcing to bounded correction and explicit escalation.

2 Related Work

2.1 BIM-to-robot execution and the missing decision layer

Recent BIM-to-robot research has moved beyond simple geometry transfer toward richer planning interfaces that translate model semantics into robot-oriented construction actions. Reviews show that BIM-robotics interaction now spans task planning, intelligent assembly, additive manufacturing, and inspection, while still facing persistent barriers in interoperability and standardized downstream use [15]. Recent frameworks connect BIM with ROS through intermediate action nodes that decompose construction planning into robot-executable activities [16], and design-to-manufacturing pipelines have further streamlined the translation of BIM data into robotic framing procedures in industrialized construction [17]. At the same time, reviews of BIM and IFC data readiness show that substantial enrichment and task-specific transformation are still required before BIM exports can reliably support AI or robotic reasoning [18]. These studies establish the path from model information to robotic action, but leave runtime decision authority under-specified when execution encounters mismatch, incomplete information, or contact uncertainty.

2.2 Human–machine collaboration, gating, and meaningful human control

A parallel line of work argues that human participation in robotic systems should be treated as an explicit design condition rather than a residual fallback. Studies on meaningful human control and variable autonomy emphasize accountability, transparency, and responsibility allocation as core requirements for human-robot teams [19]. In formal planning, Schuppe et al. incorporate human input through assume-guarantee reasoning, treating intervention as part of a correct-by-design framework rather than an informal override [20]. Construction-oriented work reaches similar conclusions. XR-robotics reviews show that AR and VR are increasingly used to support supervision, teleoperation, and human-robot coordination in construction [21], while studies in timber prefabrication identify safety, coordination, and role allocation as central HRC challenges [22]. AR-based prefabrication research further examines failure cases and user interventions when robotic placement encounters unexpected events [23]. However, in most of this literature, the human remains a supervisor or fallback operator rather than an explicitly compiled gate within the executable task representation itself.

2.3 Auditability, provenance, and the research gap

A further gap concerns accountability. Responsible-AI research in cyber-physical production argues that trust, safety, and accountability must be engineered as system properties rather than treated as external aspirations [24]. In parallel, provenance research provides mature models for representing entities, activities, and agents in traceable workflows [25][26]. Yet these ideas have rarely been embedded into BIM-to-assembly runtime execution. As a result, existing systems often record outcomes, but not the full chain linking a component record, an execution node, a failed guard, a recovery attempt, and a human decision. Taken together, the literature reveals three unresolved issues: BIM-to-robot studies formalize actions but not runtime decision rights under uncertainty; HRC and XR studies demonstrate the value of human oversight but usually treat it as supervision rather than a compiled gate; and provenance frameworks offer concepts for accountability but are not deeply integrated into execution itself. HG-ASGs address these gaps by framing assembly execution as a governable decision contract that specifies non-decisions, bounded recovery, human escalation, and traceable runtime events.

3 Methodology

3.1 Overview: From Geometry to Governance

HG-ASGs translate BIM descriptions into an executable representation that explicitly allocates decision rights between human operators and machine execution. Unlike conventional automation approaches that assume deterministic assembly conditions, our pipeline treats the transition from BIM to physical execution as a governed process. As shown in Fig. 2, the workflow comprises three stages: (i) exporting BIM

component instances into structured component records with material-specific tolerances; (ii) compiling these records into an HG-ASG contract containing action primitives, constraints, and escalation rules; and (iii) executing the contract with runtime checking, mixed-reality (MR) gating, and event logging.

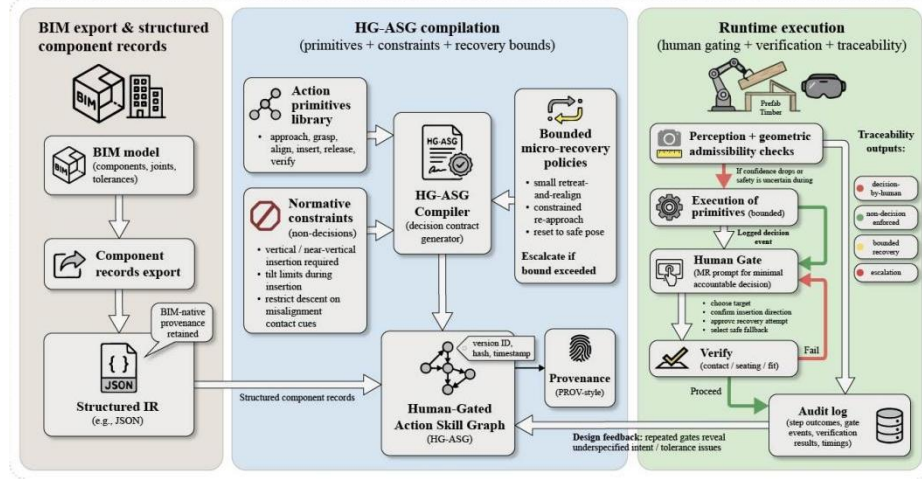


Fig. 2. Human-Gated ASG pipeline: BIM export to structured component records, ASG compilation into primitives and constraints, runtime execution with gating and logging

Conceptually, HG-ASGs implement a governed cyber-physical loop. BIM-derived intent is translated into executable clauses, while runtime execution produces observable outcomes. Deviations are treated not only as failures but also as traceable signals for incident reconstruction and later refinement. In this sense, the robot is framed not as a replacement for human labor but as a semi-autonomous agent operating within an explicitly bounded mandate.

To bridge the interoperability gap between BIM authoring and robot execution, we use a tool-agnostic intermediate representation (IR) that separates geometry from governance metadata. Each component record contains a stable identity, world pose, collision proxy, and tolerance specifications appropriate to the material context. These fields serve as the ground truth for downstream guard checking. The IR is strictly typed: if a critical field is missing, the compiler defaults to a `requires_gate` state, converting missing information into an explicit runtime condition rather than an implicit assumption.

An HG-ASG is a directed graph $G = (V, E)$, where nodes V are parameterized action primitives and edges E encode admissible transitions. We interpret the graph as a Decision Contract: nodes define permissible actions, while edges govern when control is preserved, transferred, or escalated. Each component is compiled into a canonical primitive sequence of Approach, Grasp, Align, Insert, Release, and Verify. As illustrated in Fig. 3 and summarized in Table 1, each primitive carries the parameters required for runtime control, including pose, tolerance, insertion direction, and recovery settings.

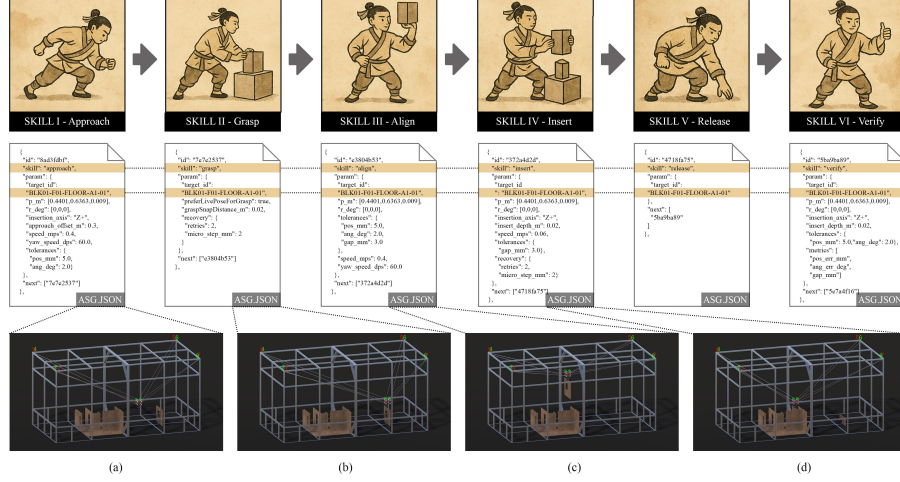


Fig. 3. ASG HG-ASG compilation and runtime execution mapping. Top: for each target component, the compiler generates a six-skill sequence, namely Approach, Grasp, Align, Insert, Release, and Verify, with excerpts of the corresponding asg.json nodes shown below each skill. Highlighted fields include the target identifier, planned pose, tolerances, and recovery parameters. Bottom: Unity snapshots show the cable robot executing representative stages of the same sequence: (a) Approach, (b) Grasp, (c) Align, and (d) Insert.

Table 1. HG-ASG primitive contract summary.

Primitive	Function	Guard	Recovery	Escalation
Approach	Pre-contact positioning	Identity, clearance	None	Missing data / timeout
Grasp	Component pickup	Reachability	None	Failed pickup
Align	Pre-insertion alignment	Geometry, state	Realign	Unresolved misalignment
Insert	Contact-aware placement	Geometry, state, tilt/contact limits	Retract, realign, reinsert	Recovery exhausted
Release	Payload detachment	Placement reached	None	Rare
Verify	Final state check	Pose / gap / seating tolerance	Retract, realign, reinsert	Verification failure

Each primitive is governed by three layers of contract conditions. First, motion parameters define target frames, speed caps, and force limits derived from component and material properties. Second, guard predicates specify pre- and post-conditions for compliance, including geometric admissibility and state admissibility. Third, non-decisions define actions that the system is explicitly forbidden to improvise, such as excessive rotation during timber insertion. By encoding such limits directly in the contract, HG-ASGs prevent the system from optimizing for speed at the expense of assembly integrity.

3.2 Bounded Micro-Recovery and Human Gating

A core innovation of the HG-ASG framework is its handling of failure. The Verify node is treated as a decision gate rather than a terminal flag. When verification fails, for instance, if a joint is not fully seated or a visual marker is occluded, the system does not enter an open-ended retry loop. Instead, it activates a Bounded Micro-Recovery Policy ($\mathbf{R}$).

The policy is formally defined as the tuple $\mathbf{R} = (\mathbf{A}, \Omega_{\text{bound}}, N_{\text{max}})$, where $\mathbf{A}$ represents the set of allowable corrective actions (e.g., local retraction, re-alignment, spiral search), Ω_{bound} defines the maximum permissive envelope for autonomous recovery, and N_{max} represents the recovery budget. Formally, the runtime decision logic is governed by the gating function $\Gamma(\mathbf{s}, \mathbf{k})$, where $\mathbf{s}$ denotes the current state error (e.g., force or pose deviation) and $\mathbf{k}$ denotes the current attempt count:

$$\Gamma(\mathbf{s}, \mathbf{k}) = \begin{cases} \textit{Proceed} & \text{if } \mathbf{s} \in \Omega_{\text{safe}} \\ \textit{Recover}(\mathbf{s}) & \text{if } \mathbf{s} \in \Omega_{\text{bound}} \setminus \Omega_{\text{safe}} \wedge \mathbf{k} < N_{\text{max}} \\ \textit{Escalate} & \text{otherwise} \end{cases} \quad (1)$$

where $\mathbf{s}$ denotes the current state error and $\mathbf{k}$ the attempt count. Here, Ω_{safe} represents the strict tolerance specified by the BIM contract, while $\Omega_{\text{bound}} \setminus \Omega_{\text{safe}}$ defines a negotiable region in which the robot may attempt bounded self-correction before escalating.

This distinction allows the same framework to be adapted to different construction contexts. High-precision timber prefabrication may require a tight Ω_{bound} close to Ω_{safe} , whereas lower-precision site conditions may use a wider recovery envelope or a lower N_{max} to encourage earlier human intervention.

When Γ evaluates to *Escalate*, the robot halts and requests a decision through the runtime human-gate interface shown in Fig.4. The interface displays the active node, target component, failure reason, and recent execution events, and provides bounded options such as recovery approval, skip, and abort. This design prevents silent drift, in which an autonomous system continues operating under unresolved error and accumulates damage or deviation. By externalizing exception handling into an explicit gate, HG-ASGs preserve human authority over out-of-contract situations while maintaining a traceable record of intervention.

3.3 Runtime Execution and Auditability

At runtime, the executor parses the HG-ASG and instantiates primitives as parameterized skills. A Contract Checker runs in parallel at high frequency, evaluating guards and emitting structured events such as *node_start*, *guard_fail*, *recovery_attempt*, *gate_trigger*, and *human_response*.

To make system limits transparent and analyzable, the logging schema follows established data provenance standards. We treat the BIM record as the Entity, execution as the Activity, and the robot or human as Agents. This logging allows post-process auditing to distinguish system limitations, such as frequent escalations

caused by overly tight tolerances, from execution errors such as hardware failures. As a result, the assembly process becomes a data-rich feedback loop rather than a black box, enabling iterative refinement of both the design and robotic behavior.

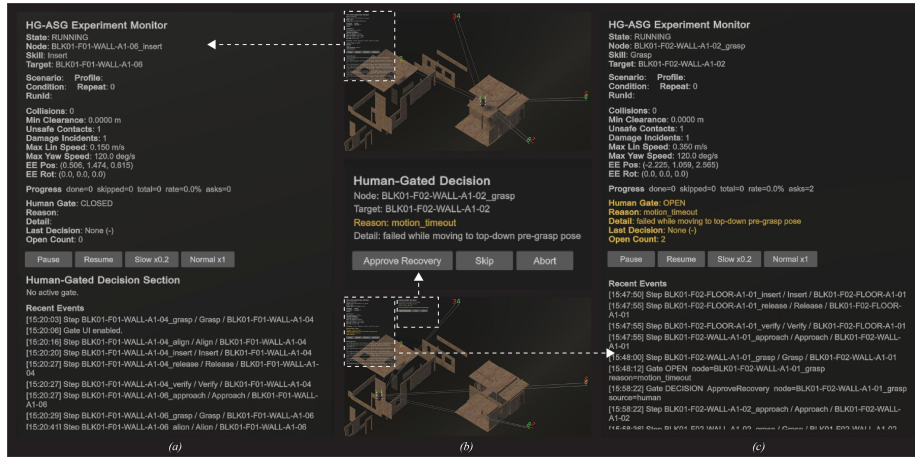


Fig. 4. Runtime human-gate interface for governed execution. (a) Normal monitoring view during autonomous execution. (b) Human-gated decision panel triggered by a contract violation or motion failure. (c) Open-gate state with runtime context and event log for traceable intervention.

4 Experiment

4.1 Experimental Setup and Uncertainty Injection

We evaluated HG-ASG in a physics-based verification environment developed in Unity. The environment serves as a digital twin of an eight-actuator cable-driven parallel robot (CDPR) and executes the same intermediate representation (IR) records and compiled HG-ASG contracts used in the physical pipeline. This setup enables controlled testing of contact events, misalignment, bounded recovery, and escalation logic before deployment.

Experiments were conducted on 10 benchmark assembly scenarios, each containing 38 components. All scenarios preserve the same compiler-to-runtime workflow while varying multi-floor adjacency, support relations, and local assembly geometry. To test robustness under different construction conditions, we injected procedural perturbations into staged component poses while keeping the contract representation unchanged. Two uncertainty profiles were used: **Profile A**, simulating high-precision timber prefabrication with minor Gaussian noise ($\sigma = 2$ mm translation, 0.5° rotation). **Profile B** simulating lower-precision site conditions with larger noise ($\sigma = 15$ mm translation, 3.0° rotation).

To isolate the effect of contract-based decision governance, we compared three execution conditions: Baseline, which performs open-loop ASG execution without bounded recovery or explicit escalation; HG-ASG-Auto, which activates the proposed

contract logic with bounded micro-recovery and automatic gate resolution for batch evaluation; and HG-ASG-Human, which uses the same contract but resolves gate events through the runtime human-gated interface. Unless otherwise noted, the main quantitative comparison focuses on Baseline versus HG-ASG-Auto, while HG-ASG-Human is used only to characterize escalation behavior. Each condition-profile-scenario combination was repeated multiple times to reduce run-to-run variance.

We report four metric groups: outcome (task success, partial completion, timeout, abort), safety (collisions, unsafe contacts, damage, minimum clearance), governance (gate triggers, escalation rate, gate distributions, recovery attempts, and runs reaching $N_{\max}$), and efficiency (cycle time and per-primitive duration). Logs were recorded at primitive, gate, and run levels, enabling both aggregate analysis and trace-level inspection of blocked, recovered, and escalated decisions.

4.2 Results

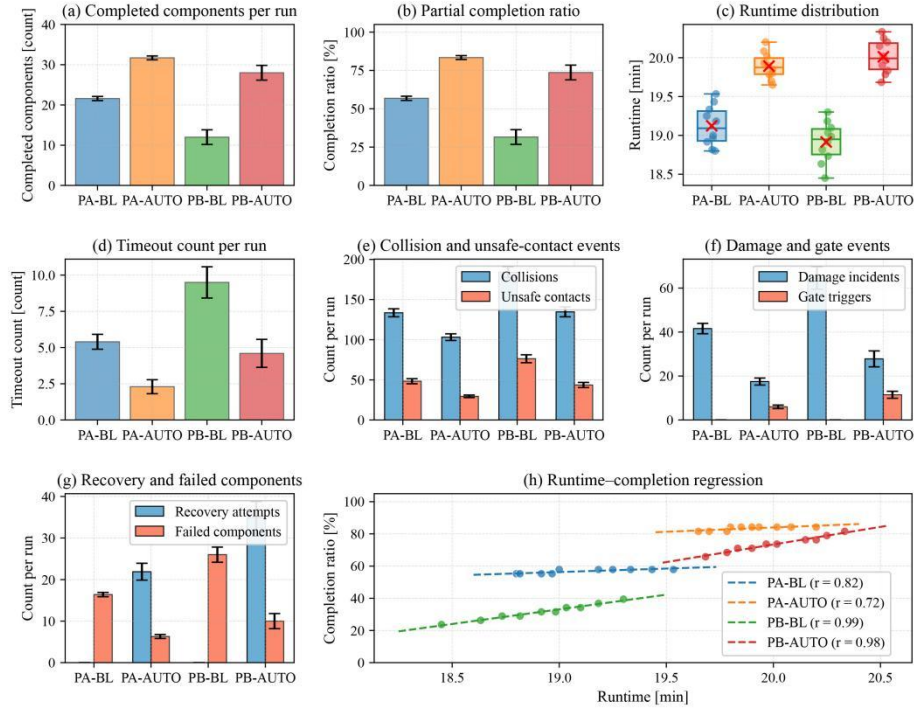


Fig. 5. Final results comparing Baseline and HG-ASG-Auto under Profile A and Profile B across 10 assembly scenarios. (a) Completed components per run. (b) Partial completion ratio. (c) Runtime distribution. (d) Timeout count per run. (e) Collision and unsafe-contact events. (f) Damage incidents and gate triggers. (g) Recovery attempts and failed components. (h) Runtime-completion relationship. PA-BL and PA-AUTO denote Profile A with Baseline and HG-ASG-Auto, respectively; PB-BL and PB-AUTO denote Profile B with Baseline and HG-ASG-Auto.

Table 2 summarizes the main quantitative outcomes, and Fig. 5 visualizes run-level performance and event-level behavior. Across both uncertainty profiles, HG-ASG-Auto consistently outperformed the open-loop Baseline by completing more components and reducing failure-related events, with only a modest increase in runtime. The gains were reflected not only in completion metrics but also in lower timeout, collision, unsafe-contact, and damage counts.

Under Profile A, the Baseline completed 21.6 of 38 components per run on average (56.8%), whereas HG-ASG-Auto increased this to 31.7 components (83.4%). Runtime rose only slightly, from 19.12 min to 19.89 min. At the same time, timeouts decreased from 5.4 to 2.3 per run, collisions from 133.5 to 103.2, unsafe contacts from 48.3 to 29.5, and damage incidents from 41.6 to 17.5. Governed execution also introduced explicit governance activity, with 6.0 gate triggers and 21.9 recovery attempts per run, while failed components decreased from 16.4 to 6.3.

Under Profile B, the same pattern became more pronounced. The Baseline completed 12.0 of 38 components per run on average (31.6%), whereas HG-ASG-Auto increased this to 28.0 components (73.7%). Runtime increased only from 18.92 min to 20.01 min, while timeouts fell from 9.5 to 4.6, collisions from 184.1 to 134.7, unsafe contacts from 76.3 to 43.6, and damage incidents from 64.6 to 27.8. The governed condition produced 11.5 gate triggers and 34.9 recovery attempts per run, while failed components dropped from 26.0 to 10.0.

Overall, the results show a stable cross-profile pattern: relative to the Baseline, HG-ASG-Auto achieved higher completion and lower failure incidence in both precision regimes, while adding only 46.1 s in Profile A and 65.8 s in Profile B. Table 2 and Fig. 5 therefore support the intended behavior of the HG-ASG decision-contract pipeline: improved completion and safer execution under uncertainty through bounded recovery and explicit escalation with limited time overhead.

Table 2. Quantitative comparison between Baseline and HG-ASG-Auto under two profiles.

Condition	Completed / 38	Completion (%)	Runtime (min)	Timeouts	Collisions	Unsafe contacts	Damage incidents
Profile A, Baseline	21.6 ± 0.52	56.8 ± 1.4	19.12 ± 0.26	5.4	133.5	48.3	41.6
Profile A, HG-ASG	31.7 ± 0.95	83.4 ± 2.5	19.89 ± 0.17	2.3	103.2	29.5	17.5
Profile B, Baseline	12.0 ± 1.83	31.6 ± 4.8	18.92 ± 0.26	9.5	184.1	76.3	64.6
Profile B, HG-ASG	28.0 ± 1.83	73.7 ± 4.8	20.01 ± 0.22	4.6	134.7	43.6	27.8

5 Conclusion and Discussion

This paper introduced the Human-Gated Action Skill Graph (HG-ASG) framework as a contract-oriented layer between BIM-derived assembly descriptions and runtime

execution under uncertainty. Instead of treating assembly automation as unrestricted task completion, the framework organizes execution through canonical action primitives, explicit guards, bounded micro-recovery, and human-gated escalation. In this formulation, routine and well-bounded operations are handled autonomously, while unresolved or unsafe situations are transferred to the human operator through an explicit governance mechanism.

The results in the Unity-based digital twin show a consistent pattern across both uncertainty profiles. Compared with the open-loop Baseline, HG-ASG-Auto achieved higher completion and lower timeout, collision, unsafe-contact, and damage counts, with only modest runtime increase. These findings indicate that the main value of the framework lies in governed robustness rather than speed alone. The human-gate mechanism further makes exception handling explicit and traceable, which is important for accountable deployment in uncertain assembly settings.

This study remains a controlled proof of concept. The current evaluation is limited to simulation and does not yet cover the full complexity of physical construction, such as multi-actor interaction, fastening operations, sensor degradation, or broader site logistics. Future work will therefore focus on physical validation, richer contract representations for more complex assembly constraints, and tighter integration between human intervention logs and adaptive planning. More broadly, the study suggests that effective construction automation may depend not on removing human involvement, but on structuring it explicitly within bounded and accountable execution.

References

1. Santoni de Sio, F., van den Hoven, J.: Meaningful human control over autonomous systems: A philosophical account. *Frontiers in Robotics and AI* 5, 15 (2018).
2. Reinmund, T., Salvini, P., Kunze, L., Jirotko, M., Winfield, A.F.T.: Variable autonomy through responsible robotics: Design guidelines and research agenda. *ACM Transactions on Human-Robot Interaction* 13(1), 1–36 (2024).
3. Yue, H., Wang, Q., Zhao, Z., Lai, S., Huang, G.: Interactions between BIM and robotics: Towards intelligent construction engineering and management. *Computers in Industry* 169, 104299 (2025).
4. Zhu, A., Pauwels, P., Torta, E., Zhang, H., de Vries, B.: Data linking and interaction between BIM and robotic operating system (ROS) for flexible construction planning. *Automation in Construction* 163, 105426 (2024).
5. Oyediran, H., Turner, W., Kim, K., Barrows, M.: Integration of 4D BIM and robot task planning: Creation and flow of construction-related information for action-level simulation of indoor wall frame installation. *Journal of Information Technology in Construction* 30, 352–374 (2025).
6. Tehrani, B.M., Alwisy, A.: Streamlining design-to-manufacturing for assembly-based robotics in wood panel framing tasks of industrialized construction: Introducing a BIM-to-BoT (B2B) framework. *Advanced Engineering Informatics* 65, 103393 (2025).
7. Kim, S., Peavy, M., Huang, P.-C., Kim, K.: Development of BIM-integrated construction robot task planning and simulation system. *Automation in Construction* 127, 103720 (2021).

8. Zhang, J., Luo, H., Xu, J.: Towards fully BIM-enabled building automation and robotics: A perspective of lifecycle information flow. *Computers in Industry* 135, 103570 (2022).
9. Du, S., Hou, L., Zhang, G., Tan, Y., Mao, P.: BIM and IFC data readiness for AI integration in the construction industry: A review approach. *Buildings* 14(10), 3305 (2024).
10. Wang, X., Yu, H., McGee, W., Menassa, C.C., Kamat, V.R.: Enabling Building Information Model-driven human-robot collaborative construction workflows with closed-loop digital twins. *Computers in Industry* 161, 104112 (2024).
11. Pan, M., Wong, M.O., Lam, C.C., Pan, W.: Integrating extended reality and robotics in construction: A critical review. *Advanced Engineering Informatics* 62, 102795 (2024).
12. Yang, X., Amtsberg, F., Menges, A.: Embracing failures: Empowering user interventions in robotic prefabrication with augmented reality. *Construction Robotics* 9, 14 (2025).
13. Verhagen, R.S., Neerincx, M.A., Tielman, M.L.: Meaningful human control and variable autonomy in human-robot teams for firefighting. *Frontiers in Robotics and AI* 11, 1323980 (2024).
14. Theodorou, A., Chiou, M., Lacerda, B., Rothfuß, S.: Editorial: Variable autonomy for human-robot teaming. *Frontiers in Robotics and AI* 11, 1465183 (2024).
15. Zhang, M., Xu, R., Wu, H., Pan, J., Luo, X.: Human-robot collaboration for on-site construction. *Automation in Construction* 150, 104812 (2023).
16. Wang, X., Wang, S., Menassa, C.C., Kamat, V.R.: Automatic high-level motion sequencing methods for enabling multi-tasking construction robots. *Automation in Construction* 155, 105071 (2023).
17. Adel, A., Ruan, D., McGee, W., Mozaffari, S.: Feedback-driven adaptive multi-robot timber construction. *Automation in Construction* 164, 105444 (2024).
18. Iturralde, K., Feucht, M., Illner, D., Hu, R., Pan, W., Linner, T., Bock, T., Eskudero, I., Rodriguez, M., Gorrotxategi, J., Izard, J.B., Astudillo, J., Cavalcanti Santos, J., Gouttefarde, M., Fabritius, M., Martin, C., Henninge, T., Nornes, S.M., Jacobsen, Y., et al.: Cable-driven parallel robot for curtain wall module installation. *Automation in Construction* 138, 104235 (2022).
19. Ohueri, C.C., Masrom, M.A.N., Noguchi, M.: Human-robot collaboration for building deconstruction in the context of Construction 5.0. *Automation in Construction* 167, 105723 (2024).
20. Schuppe, G., Torre, I., Leite, I., Tumova, J.: Follow my advice: Assume-guarantee approach to task planning with human in the loop. In: *Robotics: Science and Systems XIX*, Daegu, Republic of Korea (2023).
21. Chang, C.T., Hayes, B.: A survey of augmented reality for human-robot collaboration. *Machines* 12(8), 540 (2024).
22. Yang, X., Amtsberg, F., Sedlmair, M., Menges, A.: Challenges and potential for human-robot collaboration in timber prefabrication. *Automation in Construction* 160, 105333 (2024).
23. Yogeewaran, K., Chen, Q., García de Soto, B.: Utilizing augmented reality for the assembly and disassembly of panelized construction. *Journal of Information Technology in Construction* 28, 571–584 (2023).
24. Mezgár, I., Váncza, J.: From ethics to standards – A path via responsible AI to cyber-physical production systems. *Annual Reviews in Control* 53, 391–404 (2022).
25. Moreau, L., Groth, P., Cheney, J., Lebo, T., Miles, S.: The rationale of PROV. *Journal of Web Semantics* 35, 235–257 (2015).
26. Moreau, L., Missier, P. (eds.): PROV-DM: The PROV Data Model. W3C Recommendation, World Wide Web Consortium (2013).