Robotic fire safety inspection from statutes and BIM: abstaining judgment, observation-aware replanning and a physics-based driving check

[Author names; affiliations with full postal address and country (lowercase superscript letters); corresponding author with e-mail. The journal uses single-anonymized review, so these stay in the manuscript.]

Abstract

Fire safety conditions such as blocked egress routes change after a building is approved, which makes them candidates for robotic inspection. This study formalises Korean fire and evacuation regulations as inspection task rules, compiles tasks from Industry Foundation Classes (IFC) models, and evaluates a mission planner whose judgment abstains when evidence is missing. On two held-out buildings, generated tasks reached precision of 0.83–1.00 and recall of 0.60–1.00 against a licensed engineer’s reference. In simulation across three buildings (207,360 main runs), three-state judgment lowered wrong-verdict rates by 4.85–7.93 percentage points compared with binary judgment, and observation-aware replanning raised evidence completion by 0.118–0.175 over fixed poses. Isaac Sim driving completed 70.5 % of routes in one building but 26.7 % in another. The results support abstaining judgment and reactive replanning, and show that lightweight walkable-space models need physics checks in several buildings.

Keywords: Automated compliance checking; Fire safety inspection; Mobile robot; Building information modelling (BIM); Inspection planning; Isaac Sim

1. Introduction

Blocked egress routes, missing extinguishers and exit signs, and fire doors held open are operational conditions: they arise after a building is approved, so design-stage checking cannot detect them. They are also among the most

common findings of routine enforcement. In England, emergency routes and exits (Article 14 of the Regulatory Reform (Fire Safety) Order) were the most frequently breached article in the 50,195 fire safety audits of 2025/26, with 11,286 breaches [1]. In Korea, a national enforcement campaign in the first half of 2023 found violations at 1,026 of 5,427 inspected premises [2]. Post-fire investigations repeatedly describe such conditions: in the 2017 Jecheon sports centre fire (29 deaths) a second exit route was blocked by stored items and stair fire doors were open [3], and the Grenfell Tower Inquiry found that many self-closing devices on flat entrance doors failed to work effectively and some were missing entirely, and that no effective inspection and maintenance programme for them had been instituted [4, para. 2.66]. These conditions are visible during a walk-through, and checking them is repetitive and recurs throughout a building’s life, which makes it a candidate for mobile robots.

Two lines of work approach this problem from different sides. Automated compliance checking formalises regulations and applies them to BIM data, but mostly to the design as modelled. Robotic inspection uses BIM to plan and navigate, but mostly to collect data or detect objects, not to reach a legal verdict. Between them lie three questions that this study addresses. Can inspection tasks be generated from statutes and IFC models for buildings that were not used to develop the method? What should a robot report when evidence is missing: if it must give a pass or fail verdict, unobserved violations may be reported as compliant? And does replanning after a failed observation raise the evidence collected, at what cost in travel, compared with simple alternatives such as exhaustive coverage?

This paper makes four contributions:

1. A rule formalism and compiler that generate inspection tasks from Korean fire and evacuation regulations and IFC models, evaluated against a reference written by a licensed fire protection engineer on one development and two held-out buildings.
2. A three-state judgment that abstains with a stated reason (unreachable, not observed, missing attribute, observation constraint unmet), compared with a binary judgment on the same observations.
3. An observation-aware replanning planner compared with fixed-pose, ordering and coverage baselines and with ablations, in a factorial simulation.
4. A physics-based driving test that bounds these claims, and a disclosed account of how it changed the simulation and where it still fails.

2. Related work

2.1 Automated compliance checking and missing information

Rule-based checking of building models has a long history [5,6], and regulatory text is increasingly translated into executable rules, first with semantic NLP and logic [7], then with ontologies [8] and large language models [9,10]. Korean building legislation has been translated into executable form for permit review [11]. Fire and egress rules have been checked on design models: tall timber buildings [12], escape routes [13], and fire codes translated by NLP and evaluated on ten models against expert review [14]. Chen et al. moved fire safety checking to buildings in operation, combining layouts, site images and clauses in a knowledge graph, evaluated on a semi-synthetic case [15]. Whether operational-phase checking is feasible in practice has also been examined [16].

Missing information is a recognised obstacle. Regulation-specific information is often absent from BIM [17], and missing, incomplete or implicit information is usually addressed by semantic enrichment before checking [18]. For fire safety equipment, symbols on escape plans have been used to add missing extinguishers and signs to the model [19], and machine learning has been used to label exit doors [20]. When information remains missing, checkers can fail silently in either direction: in one fire safety checker, a wall whose fire-resistance rating had no value was reported as non-compliant, while a room the modeller had not identified could not produce any finding, and only the flame-spread check returned an unknown outcome [12]. A third outcome for missing data has been part of the definition of rule checking since Eastman et al., whose results include “unknown” for cases where the needed data are incomplete or missing [5], but it is rarely kept in practice. Jiang et al. sort results into pass, warning (to be re-checked manually) and failed; an incomplete model is stopped by a completeness check and must be modified before checking, and several of their rules record a failure whenever no passing fact can be derived, so a missing item is reported as non-compliance [8]. The “unclear” outcome of Wang et al. denotes clauses without an applicable constraint rather than missing evidence [14]. Hjelseth and Li name pass, fail and undetermined as results of a reasoner, but fill incomplete or missing information with default assumptions and what-if scenarios [21]. In all of these, the third outcome, where it exists, concerns information in the model. Here it concerns evidence in the building, and it carries the reason the evidence is missing.

2.2 BIM-driven robotic inspection

BIM has been used to build robot task and simulation environments [22,23], to plan scans with quadruped robots [24,25], and to derive inspection tasks

for facility management from ontologies, with fire-door inspection as an example [26]. Tandon et al. derive inspection poses at a fixed offset in front of selected IFC elements and execute them with behaviour trees [27]. For fire safety equipment, a legged robot has followed a predefined route to detect extinguishers and read their inspection tags [28], and a computer vision framework detects equipment, reads inspection tags and localises components in an as-is BIM model from images, without a robot, leaving rule-based compliance checks to future work [29].

The closest system is that of Tuomisto et al. [30]. A Spot quadruped localised with BIM inspects six user-selected BIM objects, three of them fire extinguishers, choosing viewpoints by next-best-view. When the view quality at a reached viewpoint falls more than 20 % below the planned score, it requests a new viewpoint, up to three attempts per target, and a vision-language model answers yes/no questions on presence and installation. It was tested in an Unreal Engine 5 digital twin of one building and in that building, and compared with two ablated variants of itself. Two of its design choices show the problem addressed here. A viewpoint reported as unreachable is removed from the inspection queue, so the target receives no outcome. And in the real-world test, an extinguisher installed on the far side of a column from its BIM position was not visible from the planned viewpoint; the image was counted as a successful capture and the extinguisher was recorded as not present. In our terms, the first is an unreachable task and the second a task whose evidence was not observed, both of which the three-state judge reports as INDETERMINATE rather than resolving silently or as a violation.

Across these studies, tasks are chosen by the user or from element classes, not compiled from statutory clauses, and the output is detection or data, not a compliance verdict that can abstain.

2.3 Observation planning, replanning and coverage

BIM-based scan planning is mostly an offline optimisation of coverage [24]. BIM-based coverage path planning for indoor robots [31] is the natural coverage baseline. Online replanning after a failed observation has appeared only recently: Duraimany and Bosché replan terrestrial laser scanning positions when occlusion reduces visibility, in simulation [32], and Tuomisto et al. re-plan viewpoints [30]. Neither targets the observation constraints that a legal verdict requires, and we did not find a study that compares replanning with fixed-pose, ordering and coverage baselines.

2.4 From BIM to drivable space

IFC models have been converted to simulation worlds and navigation maps, mostly for Gazebo [22,33], and to topology and occupancy maps that account for robot size [34]. These studies show that BIM geometry is not the

same as drivable space, but we did not find a peer-reviewed study that drives BIM-derived inspection routes with contact-level physics and reports where and why the robot fails. Simulation results also need not predict real performance: in a well-known navigation benchmark, simulated and real success correlated poorly until simulator settings were corrected [35]. We therefore treat the physics test as a bound on the lightweight simulation, not as evidence of real-world performance.

Table 1 sets the closest studies against the properties that this work combines. The first column asks whether inspection tasks, with their observation requirements, are compiled from statutory clauses (~ = regulation is turned into executable checking rules, but for design models, not inspection tasks). The second asks whether the method was evaluated on a building that was not used to develop it. The fourth asks whether the judgment can abstain when evidence is missing (~ = the third outcome concerns information in the model, not evidence in the building). The fifth asks for sensor observation by a robot (~ = simulated robot only). The seventh asks for a comparison with baseline planners (~ = ablated variants of the authors' own system). The last asks for driving the planned routes in a physics simulator or with a real robot (~ = driven, but without contact-level analysis of where and why the robot fails). Y = shown, N = not present in the work as we read it, ? = not confirmed (read from abstract or report only, or the sources disagree). N means the property was not demonstrated, not that it is impossible.

Table 1. Closest studies compared on the properties combined in this work (Y = shown, ~ = partial, N = not present, ? = not confirmed).

Study	Statute to task	Evaluated on non-development building	Legal verdict	Abstains on missing evidence	Robot observation	Replanning	Baseline comparison	Physical driving check
Tuomisto et al. 2026 [30]	N	?	N	N	Y	Y	~	~
Chen D. et al. 2026 [15]	?	N	Y	?	N	N	?	N
Jiang et al. 2022 [8]	~	?	Y	~	N	N	N	N
Wang et al. 2023 [14]	~	?	Y	N	N	N	N	N

Study	Statute to task	Evaluated on non-development building	Legal verdict	Abstains on missing evidence	Robot observation	Replanning	Baseline comparison	Physical driving check
Hjelseth and Li 2021 [21]	~	?	Y	~	N	N	N	N
Aziz et al. 2024 [28]	N	?	N	N	Y	N	?	?
Aziz et al. 2025 [29]	N	?	N	N	N	N	?	N
Chen J. et al. 2023 [26]	N	?	N	N	?	N	?	?
Chen Z. et al. 2024 [31]	N	?	N	N	N	N	?	?
Duraimany and Bosché 2025 [32]	N	N	N	N	N	Y	?	N
Gopee et al. 2023 [33]	N	N	N	N	N	N	N	~
Stührenberg et al. 2026 [34]	N	?	N	N	N	N	?	N
This study	Y	Y	Y	Y	~	Y	Y	Y

Notes: Tuomisto et al. were tested in one building and compared with two ablated variants of their own system; unreachable viewpoints were dropped and an occluded extinguisher was counted as captured. Chen D. et al. evaluated one semi-synthetic case. Duraimany and Bosché replan terrestrial laser scanning positions in simulation on one model, without a robot. Gopee et al. tested in one laboratory in Gazebo and with a real robot. For Chen J. et al., the sources we used disagree on whether a robot observed the inspected element. In this study the robot is simulated, and the physics test is treated as a bound on the lightweight simulation (Section 2.4).

2.5 Gap

In the works we found, no study combines statute-derived task generation evaluated on held-out buildings, a judgment that abstains with a reason when evidence is missing, evaluation of the planner against ordering, coverage and ablation baselines, and a physics-based check on drivability.

The closest robotic system [30] has no statutory rules or abstention; rule checkers report missing data as failure or fill it with assumptions [8,21] and work on models, not on observations; and operational-phase checking [15] has no robot.

3. Methods

3.1 Overview

The method has three stages: statutory requirements are formalised as machine-readable inspection task rules; a compiler generates inspection tasks for a building from its IFC model; and a robot mission is simulated against those tasks, with a judgment that may abstain when evidence is insufficient. Task generation was evaluated against an expert reference (E1), the judgment and planning hypotheses in a lightweight simulation (H1–H3), and route drivability in a physics simulator (E4). Fig. 1 shows the stages and where each evaluation applies.

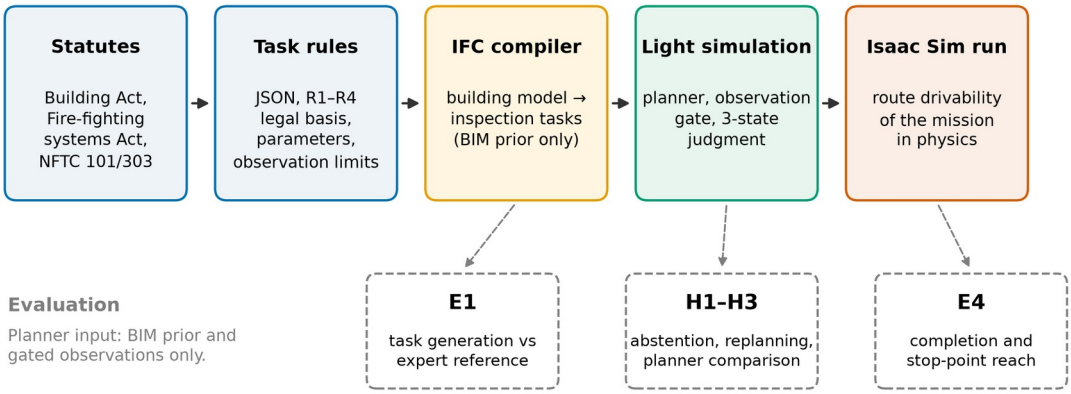


Fig. 1. Stages of the method and the evaluations E1, H1–H3 and E4.

3.2 Inspection task rules

Four rules were formalised (Table 2):

Table 2. Inspection task rules, parameters and observation constraints.

Rule	Requirement	Main parameters	Observation constraints (standoff, line of sight, facing angle)
R1	Obstruction of egress routes	clear width 1.2 m; exit zone	0.5–3.0 m; required; 60°

Rule	Requirement	Main parameters	Observation constraints (standoff, line of sight, facing angle)
	and exit front zones	depth 1.5 m; occupancy threshold 0.3; coverage 0.9	
R2	Fire door kept closed	open angle threshold 15°	0.8–2.0 m; required; 30°
R3	Portable extinguisher presence and walking distance	walking distance 20 m (small), 30 m (large); mounting height 1.5 m; extra provision for rooms of 33 m²	0.8–2.5 m; required; 45°
R4	Exit sign presence and placement	exit sign height 1.5 m; corridor sign interval 20 m; corridor sign height 1.0 m	1.0–4.0 m; required; 45°

Each rule records its legal basis, including the Act on the Installation and Management of Fire-Fighting Systems (소방시설 설치 및 관리에 관한 법률), Art. 16; the Building Act (건축법), Art. 49, and its Enforcement Decree (건축법 시행령), Art. 64; the Rules on Standards for Escape and Fire-Resistant Structures of Buildings (건축물의 피난 · 방화구조 등의 기준에 관한 규칙), Arts. 9, 14, 15-2 and 34; and the fire safety technical standards for portable extinguishers (NFTC 101) and exit lights and signs (NFTC 303). English titles are our translations. After comparison with the expert reference on the development building, rule scope was changed once (decision record 1): R1 covers all doors facing corridors and halls; doors with no fire rating in the IFC are not assumed to be fire doors, so their R2 tasks are *undetermined*; and for R3, small rooms are credited through the storey walking-distance task, with the strict comparison also reported.

Each rule is a JSON file with the same fields (Listing 1): the legal basis with statutory excerpts; applicability conditions on IFC and overlay attributes; lawful exceptions; a target selector; the evidence required for a verdict; observation constraints; an ordered decision rule with a default verdict; abstention conditions, each with a reason; and parameters, each with its source (statute or researcher-defined) and rationale. Statutory values are

read only from these files, and researcher-defined simulation values only from configuration files, so neither is written into the code. The rule files are hashed, and the hash is stored with every simulation run.

Listing 1. Rule R2 (fire door kept closed), abridged. “…” marks omitted fields or text; the statutory excerpts (Korean) are omitted, and Korean strings are given in English translation.

```
{
  "rule_id": "R2",
  "title_en": "Fire door kept closed",
  "legal_basis": [
    { "law": "Rules on Standards for Escape and Fire-Resistant
Structures of Buildings",
      "article": "Art. 14(2) item 1", "role": "primary", "excerpt":
"…", "source_doc": "…" },
    …
  ],
  "applicability": { "all_of": [
    { "attr": "door.is_fire_compartment_door", "eq": true },
    { "attr": "door.fire_rating", "in_vocab": "fire_rating_kr" } ] },
  "exceptions": [
    { "when": { "door.closing_type": "auto_closing" }, "effect":
"open_state_lawful" } ],
  "target_selector": { "ifc_class": "IfcDoor",
    "filters": [
      { "pset": "Pset_DoorCommon", "property": "FireRating", "op":
"in_vocab", "vocab": "fire_rating_kr" },
      { "pset": "Pset_DoorCommon", "property": "IsExternal", "op":
"==", "value": false } ],
    "overlay_required": [ "closing_type", "is_fire_compartment_door" ]
},
  "target_scope": "point",
  "required_evidence": [ "open_angle_deg", "closing_type",
"wedge_present" ],
  "observation_constraints": { "pose_rule": "door_front",
"standoff_m": [0.8, 2.0],
                               "line_of_sight": true,
"face_angle_deg": 30, … },
  "decision_rule": { "conditions": [
    { "when": { "closing_type": "normally_closed" }, "measure":
"open_angle_deg", "op": ">",
      "threshold_ref": "params.open_angle_threshold_deg", "verdict":
"NON_COMPLIANT" },
    { "when": { "closing_type": "normally_closed" }, "measure":
"wedge_present", "op": "==",
      "value": true, "verdict": "NON_COMPLIANT" },
    { "when": { "closing_type": "auto_closing" }, "measure":
"wedge_present", "op": "==",
      "value": true, "verdict": "NON_COMPLIANT" } ],
```

```
      "default_verdict": "COMPLIANT" },
    "abstain_if": [
      { "missing": "fire_rating", "reason": "missing_attribute" },
      { "missing": "closing_type", "reason": "missing_attribute" },
      { "unmet": "observation_constraints", "reason": "constraint_unmet"
},
      { "missing": "open_angle_deg", "reason": "not_observed" } ],
    "params": { "open_angle_threshold_deg": { "value": 15, "unit":
"deg",
      "source": "researcher-defined",
      "rationale": "No statutory value; allowance for residual opening
by a door closer.", … } }
}
```

The IFC schema has no standard property for the closing type of a door, so R2 requires it from an overlay; when it is missing the rule abstains with missing_attribute. The observation gate (Section 3.5) checks the standoff range, line of sight and, for R1 corridor segments, the share of the segment in view. The facing angle is recorded in the rules but was not checked, because the lightweight simulation does not model the robot's heading.

3.3 Buildings

Table 3 lists the three buildings.

Table 3. Buildings used in the study.

Building	Role	Storeys simulated	Source
Hwayang start-up support centre	Development	7	Private model
KIT Institute-Var-2 (AC11, IFC2x3)	Held-out	4	opensourceBIM/TestFiles, commit 2fdf948
HITOS (IFC2x3)	Held-out	5	opensourceBIM/TestFiles, commit 2fdf948, SHA-256 fe339277…

The held-out buildings are foreign, so the Korean rules were applied on the assumption that the buildings stand in Korea (office/research use for KIT, educational/research use for HITOS); the expert reference used the same assumption (decision record 3). Both held-out models are publicly available IFC models; they are used as published and are not redistributed with this study.

The walkable world is built from IFC walls, spaces and doors: walls thicker than 0.8 m are excluded as model errors, door openings are cut with width max(door width, 0.9 m), and rooms, corridors and halls are classified by name. Columns, stairs, ramps and other building elements are not subtracted from walkable space; Section 4.11 reports the consequence and a sensitivity test. The IFC models contain no extinguishers or exit signs, so fixtures were placed by rule. Fixture positions were therefore not expert-verified, and the same rule logic informs both placement and inspection (Section 5.7). Fig. 2 shows one storey of each building with its walls, doors and generated tasks; for the development building only the storey outline is shown.

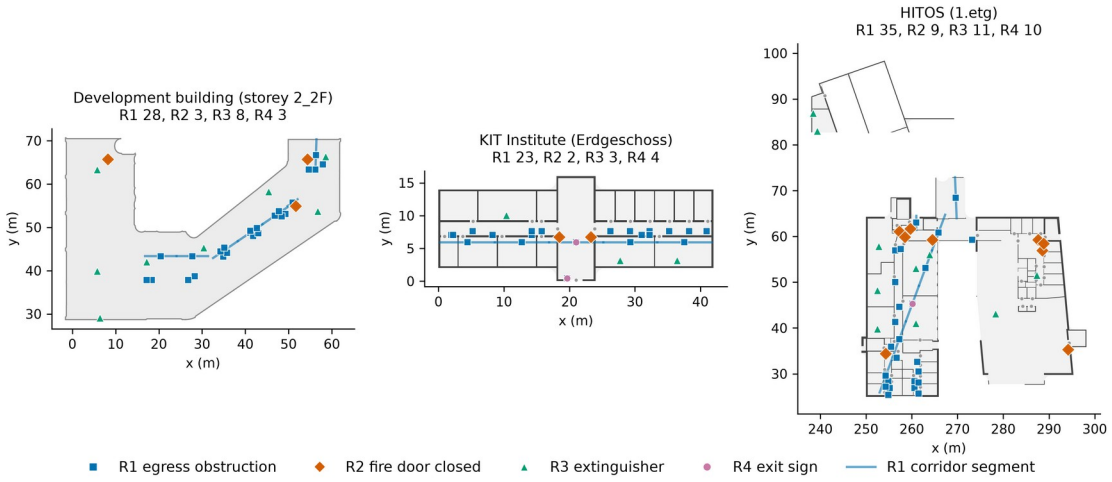


Fig. 2. One storey of each building with the generated tasks by rule R1–R4 (the development building as outline only).

3.4 Task generation and evaluation (E1)

The compiler generates tasks per rule and building. It works on the walkable world of Section 3.3 and the rule files, and uses no information about the condition of the building. Algorithm 1 summarises it; parameters named in it come from the rule files (statutory) or the compiler configuration (researcher-defined). Each task has a rule, a storey, a scope (point, space, segment or network), a target, the required evidence and, except for network tasks, a precompiled observation pose.

Algorithm 1. Task compilation for one building.

```
Input : walkable world (per storey: free space, walls, rooms,
corridors, doors with
        overlay attributes); rules R1–R4; compiler parameters
Output: inspection tasks T; synthetic fixtures F (extinguishers, exit
and corridor signs)

for each storey s:
```

```
    G ← grid graph over the free space of s
    # R2 — fire doors (point tasks)
    for each door d of s:
        if d is a fire-compartment door: add R2 task on d, status
"determined"
        else if d is a fire-door candidate (stair door, fire rating
unknown):
            add R2 task on d, status "undetermined"
(missing_attribute)
    # R1 — egress routes (segment tasks) and door-front zones (space
tasks)
    for each corridor c of s:
        split the axis of c into equal segments no longer than the
segment length;
        add one R1 segment task per segment
    for each door d that is an exit, a stair entry or faces a corridor
or hall:
        add R1 space task on the zone in front of d (zone depth from
R1)
    # R3 — extinguishers (synthetic placement, point tasks, one
network task)
    O ← grid nodes in rooms (except stairs), corridors and unlabelled
rooms
    E ← one extinguisher in each habitable room at or above the room-
area value of R3
    C ← candidate nodes along corridor axes and at stair doors
    while some node of O is beyond the walking-distance limit of R3
from every e ∈ E:
        add to E the candidate that brings most uncovered nodes of O
within the limit
        (if no candidate does, add an uncovered node of O as a new
candidate)
    add each e ∈ E to F and an R3 point task on e
    add one R3 network task: every node of O within the walking-
distance limit of E
    # R4 — exit signs (point tasks) and corridor signs (segment tasks)
    for each exit or stair-entry door d: add an exit sign at d to F
and an R4 point task on it
    for each corridor c whose axis is not shorter than the exception
length of R4:
        place corridor signs along the axis at spacing no larger than
the R4 interval; add them to F
        add one R4 segment task on these signs
    for each new point or space task: pose ← grid node at the pose
standoff from the target,
        on a corridor where possible; for segment tasks: the midpoint
of the segment
return T, F
```

The synthetic fixtures meet R3 and R4 by construction; missing fixtures enter only through the scenario generator (Section 3.5). Undetermined R2 tasks are visited like other tasks; the judge does not receive a fire rating or closing type for those doors, so their verdict is INDETERMINATE, and they are excluded from the wrong-verdict denominator (Section 3.8).

Evaluation (E1). A licensed fire protection professional engineer (A2) independently wrote the reference from floor plans, element tables and statutory excerpts, without seeing generated tasks. Precision is the share of generated tasks in the reference; recall is the share of mandatory reference tasks generated. An IFC-only baseline, using only objects in the IFC file, serves as comparison. A second annotator (A3, a fire safety practitioner with ten years of experience) independently annotated both held-out buildings from the same package, without seeing A2's reference or generated tasks; A3 used an LLM only to format a summary document. A3 added an 'undeterminable' label to the mandatory column. Agreement on the mandatory label was computed per door or space with Cohen's kappa and positive agreement.

3.5 Lightweight simulation

Each storey mission runs on a 0.3 m grid with a wall clearance of 0.297 m (measured Nova Carter half-width 0.247 m + 0.05 m, version v1.1) and a clutter clearance of 0.3 m. The robot enters zones only through stair doors and exits, not elevators; a transfer between zones costs 30 m. This movement assumption was fixed by the researcher before the held-out runs.

Scenario factors: clutter 0/5/10/20 %; fire door closed, 30 % open, or 30 % open with a third of those wedged (about 10 % of doors); extinguishers removed 0/20/50 %; signs removed 0/20 %. This gives 72 combinations, each with 15 repetitions, 2 information conditions (complete; closing type missing for 30 %), and 3 sensor levels (Table 4):

Table 4. Sensor levels in the lightweight simulation.

Sensor	Localisation σ	Angle σ	Width σ	Occupancy σ	Miss probability
Ideal	0	0	0	0	0
Noisy	0.1 m	5°	0.05 m	0.03	0.05
Degraded	0.3 m	10°	0.15 m	0.08	0.15

The observation gate returns a measurement only when the robot's pose, perturbed by the localisation noise, meets the rule's observation constraints: for point and space targets, a distance within the standoff range and a line of sight not blocked by walls or clutter; for R1 corridor segments, a share of sample points along the segment within sensor range and in line of sight (walls only) of at least the coverage value of R1. The returned value is the

true value from the scenario with the sensor’s Gaussian noise added; a present fixture or a wedge is missed with the sensor’s miss probability. For R2 the gate also returns the closing type, unless it is missing under the information condition. Measurement is an oracle with Gaussian noise; no perception model is simulated. Ground truth comes from the scenario generator and is available only to the evaluator; planners see only the BIM prior and observations passing the observation gate. Seeds are derived by SHA-256 from building, storey, combination and repetition.

3.6 Planners

Tasks are grouped by zone and ordered by nearest neighbour, using the BIM prior only. Table 5 lists the planners compared.

Table 5. Planners compared. All use the BIM prior only.

Planner	Order	Replanning	Re-check of “absent”	Observation constraints on alternative poses
Fixed-pose nearest-neighbour (FP)	Nearest neighbour	No	No	_
P	Nearest neighbour	Yes, up to 5	Yes	Yes
2-opt (FP-2opt)	Nearest neighbour + 2-opt	No	No	_
Checklist (CL)	Compiled task order	No	No	_
Coverage (COV)	Boustrophed on over 2.0 m cells	No	No	_
P_noreplan	Nearest neighbour	No	Yes	Yes
P_norecheck	Nearest neighbour	Yes	No	Yes
P_noobs	Nearest neighbour	Yes	Yes	No

The last five were added after the main analysis to restore the comparisons of the research plan, and are reported as a secondary analysis. The 2.0 m coverage spacing is a researcher-defined value.

Algorithm 2 gives planner P. Observation units are the targets the robot must look at: one per point or space task, one per R1 segment and one per sign of an R4 segment task; R3 network tasks are judged from the point observations and have no unit. Zones are the connected parts of the prior walkable grid; the robot enters a zone only through a stair door or exit (Section 3.5).

Algorithm 2. Planner P for one storey mission.

```
Input : BIM prior (grid graph, walls, zones and zone entries), tasks T
with precompiled
        poses, observation constraints of the rules; gated
observations only
Param : replanning limit N; sensing radius r_s; zone transfer cost c_z
Output: route; observation log L (a measurement or a failure reason
per unit)

 1  U ← observation units of T
 2  order U by zone (start zone, other enterable zones, unenterable
zones last) and,
    within a zone, by nearest neighbour (Euclidean, prior poses) from
the zone entry
 3  for each unit u in this order:
 4      if zone(u) ≠ current zone: move to the entry of zone(u);
distance += c_z
 5      q ← precompiled pose of u   (signs: the prior pose with wall
line of sight whose
                                     distance is nearest the middle of
the standoff range)
 6      ρ ← unreachable
 7      for attempt = 0 … N:
 8          if attempt > 0:                                          #
replanning
 9              K ← prior grid nodes meeting u's standoff range and
wall line of sight
                    (R1 segments: nodes within a search radius of the
segment midpoint),
                    excluding nodes the robot has sensed within r_s as
blocked by clutter
10              if K has no reachable node: stop
11              q ← node of K with the shortest path from the robot
12          if q is blocked or not reachable: ρ ← unreachable;
continue
13          drive to q on the shortest path; m ← Gate(u, robot pose)
14          if m reports the fixture absent:                         #
re-check
15              drive to another prior pose meeting u's constraints; m
′ ← Gate(u, pose)
16              if m′ reports the fixture present: m ← m′
17          if m ≠ ∅: L[u] ← m; continue with the next unit
```

```
        ρ ← constraint_unmet
    L[u] ← failure ρ
return route, L
```

The other planners change parts of Algorithm 2: FP has N = 0 and no re-check (lines 14–16); P_noreplan has N = 0; P_norecheck omits lines 14–16; P_noobs replaces the candidates of line 9 and line 15 by free nodes within the search radius of the target, nearest first, with no standoff or line-of-sight condition; FP-2opt improves the order of line 2 by 2-opt on Euclidean distances; CL keeps the compiled task order. COV does not use units for ordering: it visits one grid node per coverage cell of each enterable zone in boustrophedon order and, at each node, attempts every unobserved unit within reach; a unit never observed is recorded as constraint unmet if it was attempted, as not observed if it was not, and as unreachable if its zone cannot be entered.

Reachability and path length (lines 10–13) are computed on the grid with clutter cells removed, which stands in for the robot's local navigation; travel distance is the sum of these shortest paths and the zone transfer costs. The visiting order uses only the prior; candidate poses are generated from the prior and the clutter sensed within r_s, and the nearest reachable one is taken. Fig. 3 shows FP and P on one KIT storey mission.

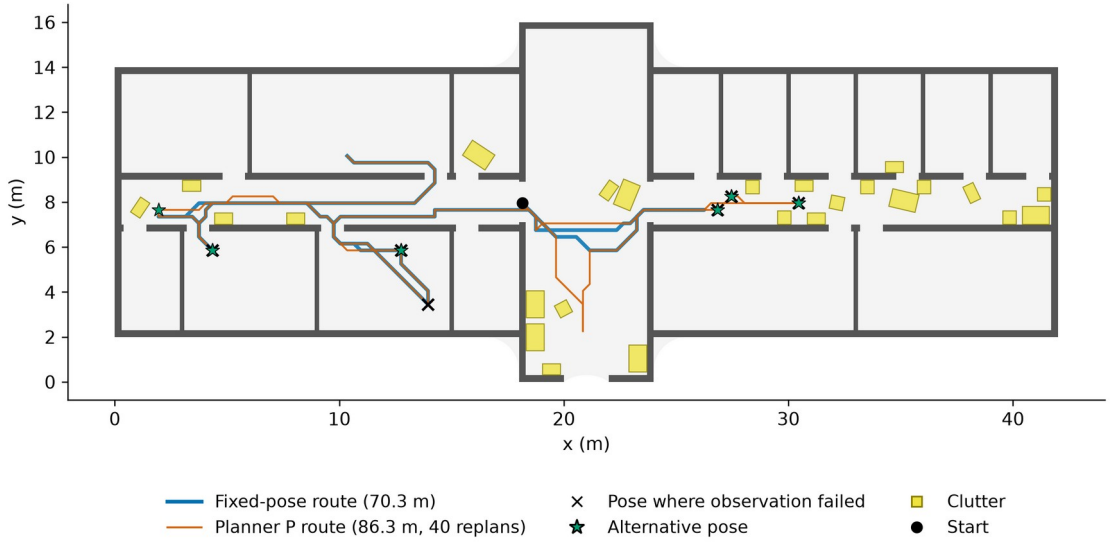


Fig. 3. Replanning example on one KIT storey mission (clutter 10 %, ideal sensor): the fixed-pose route (FP) and the route of planner P, which chooses another pose after a failed observation. Failure points are inferred from the route file as the first stop of each task that P visited more than once. One example run, not selected for effect size.

3.7 Judgment

The three-state judge returns COMPLIANT, NON_COMPLIANT or INDETERMINATE with a reason (not observed, unreachable, missing attribute, constraint unmet). It abstains when a measurement lies within 1.5 σ of a threshold. The binary baseline applies the same rules to the same observations but must return a verdict: it treats missing observations as compliant, and an unknown closing type as normally closed.

Algorithm 3 gives the judgment of one task. The judge receives the rule file, the entry of the observation log and the noise level σ of the sensor in use; it does not receive the scenario. For R1 corridor segments the width threshold is the statutory clear width, capped by the corridor's own width in the model less a small margin, so that a corridor narrower than the statutory width in the design is not reported as obstructed. Point, space and segment tasks are judged from their own observation; R3 network and R4 segment tasks are aggregated from the point observations of their fixtures.

Algorithm 3. Three-state judgment of one task, and the binary baseline.

```
Input : rule r; observation m from the log, or ∅ with the mission's
failure reason ρ;
        sensor noise σ; margin factor k
Output: verdict ∈ {COMPLIANT, NON_COMPLIANT, INDETERMINATE} and reason

Point, space and segment tasks
 1  if m = ∅: return INDETERMINATE, ρ        (ρ ∈ {unreachable,
constraint_unmet, not_observed})
 2  if m reports the fixture absent: return NON_COMPLIANT
 3  if r = R2 and the closing type is unknown: return INDETERMINATE,
missing_attribute
 4  for each condition c of r.decision_rule, in order:
 5      if c.when does not match m, or m has no value for c.measure:
continue
 6      θ ← threshold of c (from r.params; R1 segments: capped width
threshold)
 7      if σ(c.measure) > 0 and |m(c.measure) − θ| < k·σ(c.measure):
 8          return INDETERMINATE, constraint_unmet
 9      if m(c.measure) c.op θ: return c.verdict
10  return r.default_verdict

Aggregate tasks (Pr = fixtures observed present; Un = fixtures not
observed)
11  R3 network: COMPLIANT if the extinguishers in Pr keep every node
covered by the designed
    placement within the walking-distance limit; else INDETERMINATE,
not_observed if Un ≠ ∅; else NON_COMPLIANT
12  R4 segment: NON_COMPLIANT if the largest gap between signs exceeds
the interval even with
```

```
    Pr ∪ Un; COMPLIANT if it does not exceed it with Pr alone; else
INDETERMINATE, not_observed

Binary baseline: line 1 returns COMPLIANT; line 3 takes the door as
normally closed and
continues; lines 7–8 are skipped; lines 11–12 count Un as present.
```

Line 7 is the margin rule: a measurement closer to the threshold than k = 1.5 standard deviations of the sensor noise does not support either verdict. It applies only to continuous measurements (opening angle, clear width, zone occupancy); presence and wedge observations have no noise term and are judged directly. Under ideal sensing σ = 0 and the margin rule never applies.

3.8 Metrics, hypotheses and statistics

Metrics are computed per run, that is, one storey mission under one scenario, information condition, sensor level and planner. For run j with task set T_j, let y_t be the ground truth of task t (COMPLIANT, NON_COMPLIANT, or UNDETERMINABLE for R2 tasks on doors of unknown fire rating), v_t the three-state verdict and b_t the binary verdict, and let $D_j = \{t \in T_j : y_t \neq \text{UNDETERMINABLE}\}$ be the determinable tasks, and let $card$ denote the number of elements of a set. Then

$$WR_j^3 = \frac{card\{t \in D_j : v_t \notin \{y_t, \text{INDETERMINATE}\}\}}{card\, D_j}, \quad (1)$$

$$WR_j^{bin} = \frac{card\{t \in D_j : b_t \neq y_t\}}{card\, D_j}, \quad (2)$$

$$ECR_j = \frac{card\{t \in D_j : v_t \neq \text{INDETERMINATE}\}}{card\, D_j}, \quad (3)$$

$$FA_j = card\{t \in T_j : y_t = \text{COMPLIANT}, v_t = \text{NON_COMPLIANT}\}. \quad (4)$$

The wrong-verdict rate WR counts an abstention as not wrong for the three-state judge. Evidence is complete for a task when the three-state judge returns a verdict, so ECR falls both when the robot fails to observe a task and when the margin rule withholds a verdict. A false alarm FA is a count per run.

Each hypothesis effect is the mean of run-level paired differences, with pairs matched on storey, scenario combination, repetition, information condition and sensor level:

$$\Delta_{H1} = \overline{WR_j^{bin} - WR_j^3}(\text{planner P, all sensors}), \quad (5)$$

$$\Delta_{H2} = \overline{ECR_j^P - ECR_j^{FP}}(\text{all sensors}), \quad (6)$$

$$\Delta_{H3} = \overline{FA_j^{FP} - FA_j^{P}} \text{ (three-state judge, noisy and degraded sensors)}. \quad (7)$$

Positive values favour the hypothesis. For the interval, the storeys of a building are drawn with replacement, the mean is recomputed over all pairs of the drawn storeys, and this is repeated B times; the 95 % interval is given by the 2.5th and 97.5th percentiles of the resampled means Δ^{ι}, and the two-sided p-value is $p = max\{2\,min(Pr[\Delta^{\iota} \le 0], Pr[\Delta^{\iota} \ge 0]), 1/B\}$.

The directions of the hypotheses were fixed in advance: H1, the three-state judge has a lower wrong-verdict rate than the binary judge; H2, P has higher evidence completion (ECR) than the fixed-pose baseline; H3, P has fewer three-state false alarms under noisy and degraded sensing. A fourth, exploratory hypothesis (H4) concerned missing closing-type attributes; the plan did not specify a test for it, and it was tested post hoc with the same design, outside the Holm family (Section 4.5). Effects are run-level paired differences as defined above, with 95 % cluster-bootstrap intervals (B = 10,000 resamples, storeys as clusters; 7, 4 and 5 clusters), with Holm correction over H1–H3 within each building. With so few clusters the intervals should be read with caution.

The study was not preregistered on an external registry. The pipeline was frozen and the reference locked by hash before the first run on the held-out buildings (compiler-v1.0-dev, aggregate SHA-256 79029fda…, commit 01e2de3; reference commit 9284f18). The development building was run before the plan was fixed and is reported as exploratory.

3.9 Post-hoc changes

All changes made after results were seen were recorded as numbered decision records before the affected runs and are reported here (Table 6):

Table 6. Changes made after results were seen.

Record	Change	Reason	Status of the earlier result
decision record 4	Wall clearance 0.15 → 0.297 m; stable sort (v1.1)	E4 showed the robot body could not pass; platform-dependent ties	v1.0 retained as comparison
decision record 5	Six further planners (v1.2)	Restore comparisons of the research plan	Secondary analysis; FP and P unchanged
decision record 6	E4 sensitivity sets, HITOS driving, P driving	Explain stuck runs and test other building and planner	Added sets; values fixed before running

Record	Change	Reason	Status of the earlier result
decision record 7	HITOS world with columns, stairs and ramps	Explain low completion in HITOS	Post-hoc sensitivity; v1.1 stays the main result

Held-out status is therefore qualified: KIT informed the v1.1 change, and HITOS remained unused for that change only.

3.10 Physical driving (E4)

Routes were exported from the lightweight simulation and driven in NVIDIA Isaac Sim by a Nova Carter. IFC models were converted to USD with colliders. Doors were made passable and furniture removed, to match the lightweight world; clutter was added as 1 m-high static boxes at the same positions. Control was pure pursuit with differential drive and no obstacle avoidance (default 0.6 m/s, lookahead 0.6 m; slow set 0.4 m/s, 0.4 m), physics at 60 Hz, with effective wheel parameters calibrated in simulation before each drive.

A run is stuck when progress stays below 0.1 m over 15 s; it completes when the route end (within 0.25 m) and all stops (within 0.3 m) are reached. Stops physically reached are reported with the stricter 0.25 m. Every result passes automatic checks (finite values, distance ratio 0.8–1.3 on completed runs, cross-track error below 5 m); runs that fail are kept in a separate folder. Under decision record 7, runs failing the 5 m check on the corrected HITOS world count as not completed. One run that crashed Isaac without producing a result was re-run (KIT slow set), with the failed attempt retained.

Routes covered 4 or 5 storeys × 4 clutter levels × 8 repetitions, with closed doors, no fixture removal, full information and ideal sensing, plus one open-door route in KIT. Routes with no stop (10 of 160 in HITOS) were not driven. Identical routes occur because 0 % clutter has no random element, so completion is reported for all and distinct routes. Causes of stuck runs were assessed geometrically from the last pose against the Isaac scene, not from contact records.

A route is driven as one leg per zone, and it is completed ($c_i=1$) when every leg is completed. With N routes in a set, including routes that failed an automatic check (counted as not completed), and S the set of routes that passed the checks, completion rate (CR) and stop reach rate (SR) are

$$CR=\frac{1}{N}\sum_{i=1}^{N} c_i, \quad (8)$$

$$SR = \frac{\sum_{i \in S} \sum_{s \in S_i} 1\left[\min_{\tau} \left\| x_i(\tau) - s \right\| \le 0.25\,m\right]}{\sum_{i \in S} card\, S_i}, \quad (9)$$

where S_i is the set of stops of route i and $x_i(\tau)$ the robot position at time τ. For the distinct-route figures, routes with the same storey, scenario combination, route length and number of stops are counted as one route, which is counted as completed only if all of its drives were completed.

E4 tests route following only. It does not include sensing, door operation or furniture, so it does not validate evidence collection.

3.11 Reproducibility

The KIT v1.1 batch was run independently on x86 and aarch64 machines; all runs and task-level decisions agreed apart from wall-clock time (Section 4.8). Code, configuration, results and decision records are held in one repository; the private development building's model and derivatives cannot be shared.

4. Results

4.1 Overview of the evaluation

The pipeline was developed on one building (Hwayang start-up support centre, 7 storeys; hereafter *development building*) and then applied, without any change to code or configuration, to two buildings it had never seen (*held-out buildings*): the KIT Institute-Var-2 office/laboratory model (4 storeys) and the HITOS university model (5 storeys). For the held-out buildings the compiler and all configuration files were frozen (aggregate SHA-256 79029fda…, commit 01e2de3) and the expert reference was locked by hash (commit 9284f18) before the compiler was run on them for the first time. The study was not preregistered on an external registry; the analysis plan, code and reference were fixed in the repository before execution.

After the held-out runs, the physical driving experiment (E4, Section 4.11) showed that the lightweight walking grid admitted robot positions closer to walls than the Nova Carter body allows. The lightweight simulation was therefore revised once (v1.1: footprint-aware wall clearance), recorded in decision record 4 before the re-run, and all simulation results were regenerated. Task generation (E1), rules, judgment logic, factors and statistics were not changed. Sections 4.3–4.7 report v1.1; Section 4.9 compares v1.0 and v1.1.

The compiler generated 293 inspection tasks for the development building (R1 175, R2 16, R3 58, R4 44), 130 for KIT (R1 92, R2 11, R3 14, R4 13) and 305 for HITOS (R1 171, R2 40, R3 44, R4 50). The simulation study used the

same factorial design in all three buildings — 72 scenario combinations (clutter 0/5/10/20 % × fire-door state 3 × extinguisher removal 0/20/50 % × sign removal 0/20 %) × 15 repetitions × 2 information conditions × 2 planners × 3 sensor levels — giving 90,720, 51,840 and 64,800 runs respectively.

4.2 Task generation accuracy (E1)

Table 7 compares the generated tasks with the reference written independently by a licensed fire protection professional engineer (A2), who saw only floor plans, element tables and statutory excerpts.

Table 7. Task generation accuracy against the expert reference (A2). P = precision, R = recall of mandatory tasks.

Rule	Development	KIT (held-out)	HITOS (held-out)
R1 door-front obstruction, P / R	0.829 / 0.830	1.000 / 1.000	1.000 / 0.825
R3 extinguisher (walking-distance task credited), P / R	0.978 / 0.975	0.944 / 1.000	0.833 / 1.000
R4 exit sign, P / R	0.976 / 1.000	1.000 / 1.000	1.000 / 0.600
R2 fire door (determined)	1.000 / 1.000	n/a (no mandatory)	n/a (no mandatory)

P = share of generated tasks that appear in the reference; R = share of mandatory reference tasks that were generated. An IFC-only baseline that uses only objects present in the IFC file produced no R1, R3 or R4 task in any building (precision and recall 0), because none of the three models contains extinguishers, exit signs or explicit evacuation-route objects.

On both held-out buildings every generated R1 and R4 task was also in the expert reference (precision 1.000). Recall on KIT was complete. On HITOS the compiler missed 24 of 137 mandatory R1 tasks and 12 of 30 mandatory R4 tasks. Post-hoc inspection of the expert notes (no code change) attributes these misses to (i) doors between two non-corridor rooms that the expert treated as part of the evacuation route, (ii) doors facing small corridors, and (iii) exterior doors whose type name carries no “Entrance” keyword, which the frozen rules therefore did not recognise as exits. The third cause is a limitation of detecting exits from type names (Section 5.6).

Inter-annotator agreement. The two annotators agreed closely where they recorded the same unit: on evacuation-route spaces for R1 (kappa 0.84 KIT, 0.95 HITOS) and on doors needing an exit sign for R4 (0.93, 0.85). They differed in granularity for R1 at doors: A2 listed every door individually (76 and 137 mandatory), whereas A3 covered most door fronts through the

corridor spaces and listed only key doors (10 and 29, all of which A2 also marked mandatory), so door-level kappa is low (0.00, 0.14) without indicating a disagreement about obligations. For R3, A3 marked rooms of 33 m² or more; in KIT these five rooms were all in A2's list, which additionally named corridors as placement candidates (kappa 0.28 KIT, 0.61 HITOS). Neither annotator marked any fire door as mandatory; A3 labelled all of them undeterminable (12 and 37 doors), consistent with the compiler's undetermined R2 tasks.

For R2, neither held-out model carries a fire rating. The compiler therefore created *undetermined* fire-door tasks for all stair-adjacent doors (KIT 11, HITOS 40), and the expert likewise marked every fire-door item as conditional (no mandatory R2 item). Of the undetermined tasks, 9/11 (KIT) and 18/40 (HITOS) appear in the reference list.

4.3 Evidence-aware three-state judgment reduces wrong verdicts (H1)

Table 8 and Fig. 4 report the wrong-verdict rate of the replanning planner P under three sensor levels. Wrong verdict = a COMPLIANT/NON_COMPLIANT verdict that disagrees with the ground truth, divided by all determinable tasks; tasks on doors with unknown fire rating are excluded from the denominator (decision record 1).

Table 8. Wrong-verdict rate of planner P by sensor level (three-state / binary judge).

Building	Ideal: 3-state / binary	Noisy	Degraded
Development	0.00 / 7.68 %	0.09 / 7.92 %	0.65 / 8.87 %
KIT	0.00 / 5.02 %	0.09 / 5.46 %	0.60 / 6.60 %
HITOS	0.00 / 4.59 %	0.07 / 4.74 %	0.53 / 5.42 %

The binary baseline must return a verdict even when the evidence is missing; it therefore labels most unobserved violations as compliant. With planner P and ideal sensing, the binary judge missed 46,508 of 78,068 violations in the development building, 12,894 of 25,374 in KIT and 26,248 of 45,554 in HITOS; the three-state judge missed none (it returned INDETERMINATE instead). When the three-state judge did commit to a verdict, it was correct in 98.6–100 % of cases across sensor levels in the development building, 99.0–100 % in KIT and 98.8–100 % in HITOS.

The paired run-level difference (binary − three-state, planner P, all sensors) was +7.93 percentage points (95 % cluster-bootstrap CI +5.42 to +10.49) in the development building, +5.47 (+3.67 to +6.70) in KIT and +4.85 (+4.14 to +5.51) in HITOS (Fig. 5). The confidence intervals exclude zero in all three buildings (Holm-adjusted p = 0.0003, the resolution limit of 10,000 bootstrap

resamples). Clusters were storeys: 7, 4 and 5 respectively, so the intervals should be read conservatively.

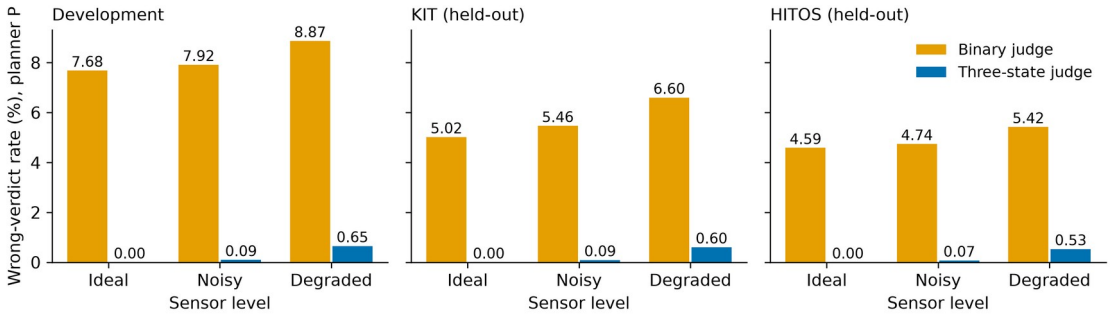


Fig. 4. Wrong-verdict rate by sensor level for the three-state and binary judges (planner P).

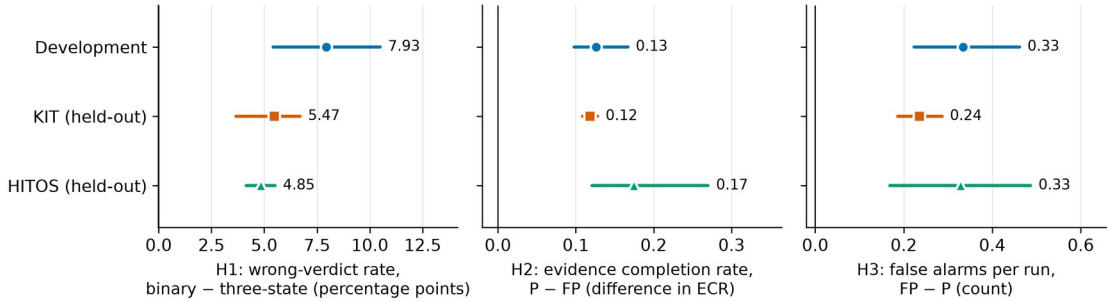


Fig. 5. Effects (dots) and 95 % cluster-bootstrap confidence intervals (lines) for H1–H3 in the three buildings.

4.4 Replanning after failed observation raises evidence completeness (H2)

Evidence completion rate (ECR) is the share of tasks for which the required evidence was obtained. Table 9 and Fig. 6 show ECR by clutter level.

Table 9. Evidence completion rate (ECR) by clutter level (P / FP).

Clutter	Development P / FP	KIT P / FP	HITOS P / FP
0 %	0.748 / 0.623	0.938 / 0.878	0.809 / 0.593
5 %	0.586 / 0.442	0.699 / 0.568	0.516 / 0.338
10 %	0.426 / 0.291	0.510 / 0.365	0.380 / 0.219
20 %	0.215 / 0.114	0.314 / 0.178	0.267 / 0.122

Planner P (observation-aware replanning) exceeded the fixed-pose baseline (FP) at every clutter level in every building. The paired difference P − FP averaged over clutter levels was +0.126 (95 % CI +0.098 to +0.167), +0.118 (+0.108 to +0.128) and +0.175 (+0.121 to +0.270). The gain costs

travel: at 0 % clutter P travelled 192.0 m vs 181.5 m (development), 101.8 m vs 94.1 m (KIT) and 305.7 m vs 235.4 m (HITOS) per storey mission, with on average 22.3, 7.8 and 32.0 replans. Travel is reported per storey mission, the unit of simulation.

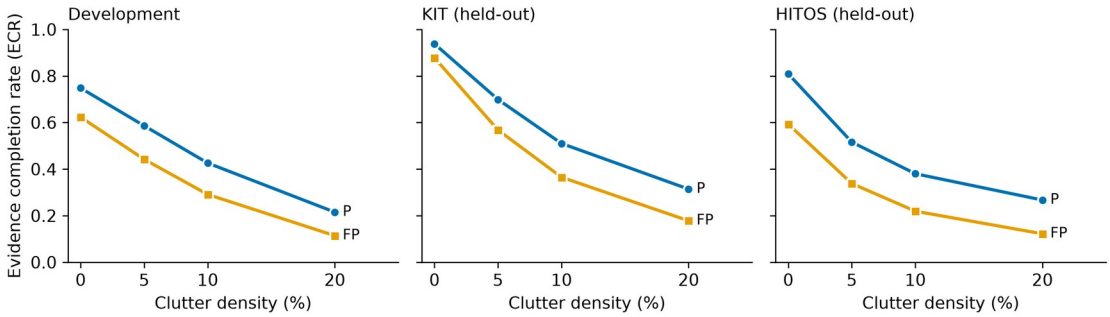


Fig. 6. Evidence completion rate by clutter level for P and FP.

4.5 Re-checking “absent” observations reduces false alarms (H3)

Under noisy and degraded sensing a fixture can be missed by the detector, which a one-shot planner reports as a violation. Planner P re-observes every “absent” result from a second pose. Run-level false alarms of the three-state judge (FP − P, noisy and degraded sensors) were +0.33 per run (95 % CI +0.22 to +0.46), +0.24 (+0.19 to +0.29) and +0.33 (+0.17 to +0.49). In totals for the noisy sensor, three-state false alarms fell from 3,743 (FP) to 545 (P) in the development building, from 1,492 to 207 in KIT and from 2,758 to 416 in HITOS.

The rule-level breakdown (planner P, noisy sensor) shows one exception: for R4 exit signs the three-state and binary judges produced the same number of false alarms in all three buildings (369, 94 and 308). The margin rule that withholds verdicts near a threshold does not apply to a presence/absence observation, so a sign that is missed twice is reported as absent by both judges. More re-checks or a detection-confidence model for presence observations are left to future work.

Exploratory H4. The plan listed H4 (when closing-type attributes are missing, false alarms increase more for the binary judge than for the three-state judge) as exploratory without a test. A post-hoc test with the same paired design and storey-cluster bootstrap (pairs matched on storey, combination, repetition, planner and sensor across the complete and 30 %-missing information conditions; not part of the Holm family) did not support it. The difference in false-alarm increase per run (binary − three-state) was +0.012 (95 % CI −0.003 to +0.027) in the development building, −0.002 (−0.005 to +0.003) in KIT and −0.004 (−0.008 to −0.001) in HITOS. The binary judge’s false alarms rose by at most 0.011 per run when the attribute

was missing, so there was little room for a difference; why the missing attribute so rarely produced false alarms was not examined.

4.6 Why tasks are left indeterminate

Most INDETERMINATE verdicts were caused by reachability, not by sensing. For planner P with ideal sensing (Fig. 7), the reasons were: unreachable 212,410 / constraint unmet 58,374 / not observed 15,572 / missing attribute 850 (development); 70,110 / 16,766 / 6,570 / 0 (KIT); 243,236 / 40,398 / 14,892 / 0 (HITOS). The unreachable count is practically independent of sensor level, while constraint-unmet verdicts grow as sensing degrades (development: 58,374 → 80,937 → 96,539), as intended by the margin rule.

Unreachable tasks were split by a criterion fixed before the held-out runs: a task that is unreachable in every 0 %-clutter run is *structural*. The development building had 31 structural tasks (all in walkable zones without a stair or exit door), KIT had none, and HITOS had 14 (all in zones without an entry door) plus 4 *intermittent* tasks that were unreachable in only some 0 %-clutter runs because of door wedges. All remaining unreachability is caused by clutter: 436,357 (development), 210,322 (KIT) and 632,948 (HITOS) task-runs summed over sensors for planner P. The researcher confirmed the 32 v1.0 structural tasks of the development building against its drawings. The four tasks whose status changed between v1.0 and v1.1 were diagnosed geometrically: two were isolated pockets of walkable space of 1.3 m² (development building) and 0.2 m² (HITOS) that the changed clearance removed or connected; the other two (HITOS storeys 1 and 2, the Gangbro bridges) became unreachable because, in the lightweight world, the bridge connects to the corridor only through a gap of about 0.5 m between a partition end and a curved element, which the 0.297 m clearance closes, although the adjacent doors are 1.99 m wide. HITOS has no drawings independent of its model, so this is reported as a limitation of the lightweight world rather than of the building.

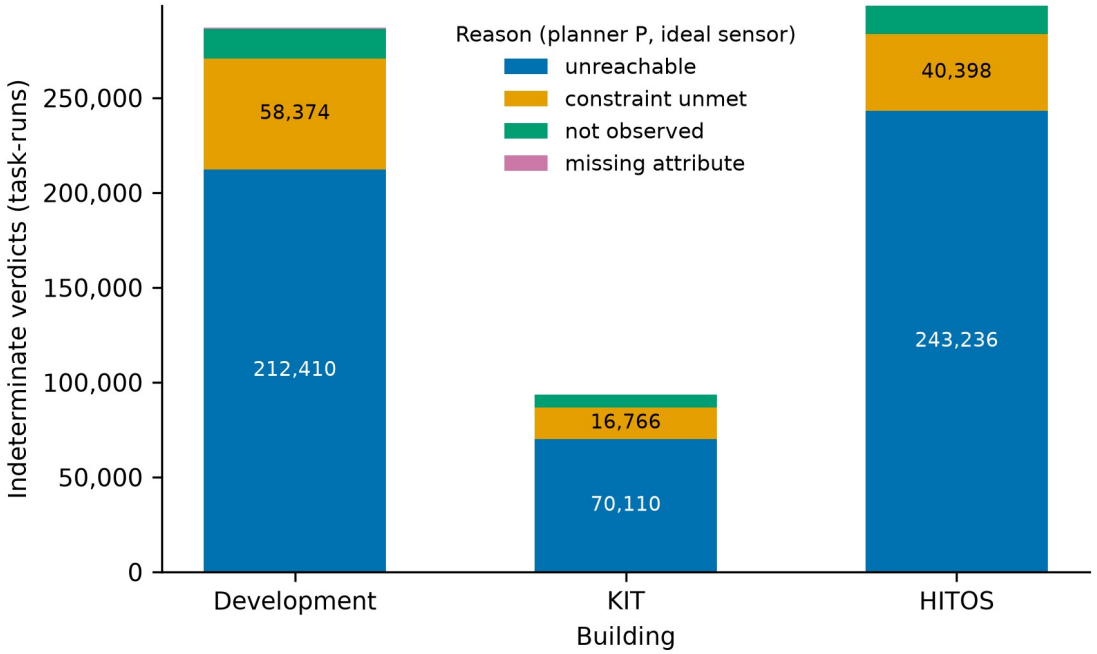


Fig. 7. Reasons for indeterminate verdicts by building.

4.7 Doors with unknown fire rating

For doors whose fire rating is unknown (development 25,920; KIT 23,760; HITOS 86,400 task-runs per sensor × planner cell), the three-state judge abstained in every case with reason missing_attribute. The binary judge returned a verdict in every case — with planner P and ideal sensing, 21,606 compliant and 4,314 non-compliant (development), 19,546 / 4,214 (KIT) and 76,964 / 9,436 (HITOS) — none of which is supported by evidence about the door's legal status.

4.8 Cross-platform reproducibility

In v1.0, the full KIT batch re-run on a machine with a different CPU architecture (NVIDIA DGX Spark, aarch64, vs x86 cloud) differed in 2,703 of 51,840 runs (5.2 %), all from planner P, because tied candidate poses were ordered by an unstable sort; conclusions were unchanged. v1.1 uses a stable sort. The full KIT v1.1 batch run independently on both machines was identical: all 51,840 runs agreed in every recorded field except wall-clock time, and all 1,684,800 task-level decision rows were identical.

4.9 Sensitivity to walking-grid clearance

Table 10 compares the two versions. Doubling the wall clearance increased the number of unreachable tasks per run (planner P: 12.84 → 14.15 development, 6.02 → 8.11 KIT, 24.48 → 24.98 HITOS) and lowered mean ECR (P: 0.523 → 0.494, 0.684 → 0.615, 0.511 → 0.493). The three hypothesis

effects kept their sign and their intervals continued to exclude zero in all buildings:

Table 10. Hypothesis effects in v1.0 and v1.1.

	Development v1.0 → v1.1	KIT v1.0 → v1.1	HITOS v1.0 → v1.1
H1 (%p)	+7.45 → +7.93	+4.26 → +5.47	+4.82 → +4.85
H2 (ECR)	+0.148 → +0.126	+0.142 → +0.118	+0.207 → +0.175
H3 (false alarms / run)	+0.32 → +0.33	+0.25 → +0.24	+0.29 → +0.33

H1 grows slightly because more tasks become unobservable, which the binary judge resolves as compliant; H2 shrinks because some poses that P could reach by replanning are no longer available.

4.10 Comparison with further baselines and ablations (secondary analysis)

The planners below were added after the main analysis to restore the comparisons of the research plan (decision record 5), so this is a secondary analysis, not a pre-specified test; 622,080 further runs were made. Table 11 and Fig. 8 compare all eight planners.

Table 11. Secondary comparison with exhaustive coverage (planner set v1.2).

Building	ECR: P / coverage / fixed-pose	Mean travel per storey mission: P / coverage
Development	0.494 / 0.497 / 0.368	145.8 m / 597.2 m
KIT	0.615 / 0.619 / 0.497	63.6 m / 365.0 m
HITOS	0.493 / 0.471 / 0.318	177.2 m / 652.8 m

Exhaustive coverage did not differ from P in ECR in any building (P − coverage −0.003 [−0.014, +0.007], −0.003 [−0.011, +0.003], +0.022 [−0.002, +0.060]) while travelling 4 to 6 times as far. Replanning accounts for the ECR gain: removing it returned ECR to the fixed-pose value (P − P_noreplan +0.126, +0.118, +0.175), whereas changing only the visiting order (2-opt, checklist order) did not change ECR. Observation-aware alternative poses contribute part of the gain (P − P_noobs +0.063, +0.036, +0.118). Re-checking did not affect ECR but lowered false alarms by 0.549, 0.348 and 0.644 per run, at a cost of 4.4, 3.7 and 5.0 m. P_noreplan had slightly fewer false alarms than P (0.055, 0.032, 0.074 per run). H3 therefore credits re-checking rather than the whole of P; a likely but unexamined reason for the P_noreplan result is that fewer tasks are observed.

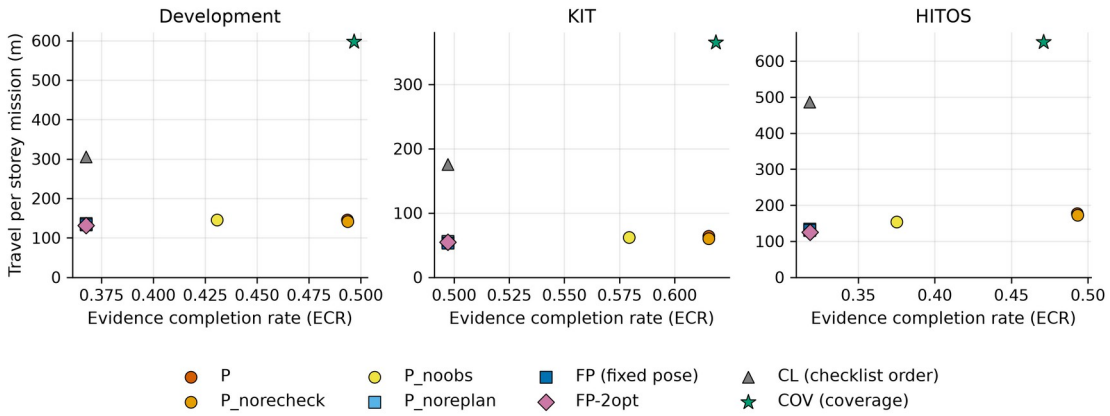


Fig. 8. Evidence completion rate and travel distance of the eight planners in the three buildings.

4.11 Physical driving fidelity (E4)

Routes were exported from the lightweight simulation and driven by an NVIDIA Nova Carter in Isaac Sim (Fig. 9) with a pure-pursuit controller without obstacle avoidance. Table 12 and Fig. 10 give completion by route set. Doors were made passable and furniture removed; clutter was placed at the same positions. Driving tests route following only: no sensing was simulated in Isaac, so this experiment does not validate evidence collection. Repetitions at 0 % clutter are identical, so completion is given both for all routes and for distinct routes.

KIT. v1.0 routes were almost never completed (3 of 129), which motivated v1.1 (Section 4.9). With v1.1, 91 of 129 routes (70.5 %; distinct 67 of 98, 68.4 %) were completed and 89.0 % of 2,564 stops were reached. At all 38 stuck poses the front of the body was within 0.05 m of a wall or clutter box (nearest object clutter 21, wall 17) although the planned routes kept at least 0.30 m, with a mean deviation of 0.19 m at the stop. These are geometric results from the last pose, not contact records. Two sensitivity sets fixed in advance (decision record 6) support tracking error against a narrow margin as the cause: slower control (0.4 m/s, lookahead 0.4 m) completed 125 of 129 routes (distinct 94 of 98) with 98.5 % of stops reached, and a clearance of 0.397 m completed 123 of 129 (distinct 89 of 95) with 97.0 % reached, at the cost of about 10 % of the stops (2,306). Routes of planner P, driven with default control, gave a similar completion to the baseline (90 of 129; distinct 69 of 101) with 82.7 % of 4,494 stops reached.

HITOS. KIT informed the v1.1 change, so HITOS is the check on a building not used for that change. Here completion was much lower: 40 of 150 routes (26.7 %; distinct 23 of 101) with 33.8 % of stops reached; 10 of 160 routes had no stop because clutter blocked the start zone and were not driven. At the 110 stuck poses the nearest objects in the Isaac scene were walls (77), stairs (25), clutter (6), a ramp (1) and a column (1). The lightweight world

does not subtract columns, stairs or ramps from walkable space. A post-hoc world including them (decision record 7) removed the stair contacts and raised completion to 66 of 150 (44.0 %; distinct 45 of 97, four further routes failed an automatic check when the robot was pushed more than 5 m off the route and count as not completed); slow control on that world reached 83 of 150 (55.3 %; distinct 55 of 96, five check failures). The hypothesis effects in HITOS were stable under this change (H1 +4.85 → +4.92 %p, H2 +0.175 → +0.180, H3 +0.329 → +0.301; ECR P 0.493 → 0.481). To identify the remaining causes directly, 52 distinct stuck routes on that world were driven again with PhysX contact reporting, which did not change the driving values on the route tested. Of 78 stuck route segments (Fig. 11), 33 involved a floor-level problem (the robot more than 2 cm from the storey floor height or touching the ground plane below), all on the u.etg storey; 31 were wall contacts; 12 were clutter contacts; 1 was a slab edge and 1 other. The lightweight world assumes a flat floor wherever walls and rooms enclose space and does not model slab extent or level, which HITOS exposes and KIT does not. Of the 31 wall contacts, 17 were at one wall (AV11T), part of which lies in the walkable space because a door opening is cut there, next to a double door; the exact mechanism was not confirmed. These causes were not corrected in a further version; they are reported as limitations (Section 5.7).

Table 12. Route completion of physical driving in Isaac Sim (Nova Carter). Routes from planner FP unless stated.

Set	Routes	Completed	Completed %	Distinct routes	Distinct completed	Stops %
KIT v1.1 route, default control (0.6 m/s)	129	91	70.5	98	67	89.0
KIT v1.1 route, slow control (0.4 m/s, lookahead 0.4 m)	129	125	96.9	98	94	98.5
KIT route with 0.397 m clearance, default control	129	123	95.3	95	89	97.0
KIT planner P route, default control	129	90	69.8	101	69	82.7
HITOS v1.1 route, default control	150	40	26.7	101	23	33.8
HITOS world with columns, stairs, ramps, default control	150	66	44.0	97	45	44.8
HITOS world with columns, stairs, ramps, slow control	150	83	55.3	96	55	48.0

These results show that lightweight routes were largely drivable in one building and only partly in the other. The ECR values of Sections 4.4–4.6 should not be read as what a physical robot would collect.

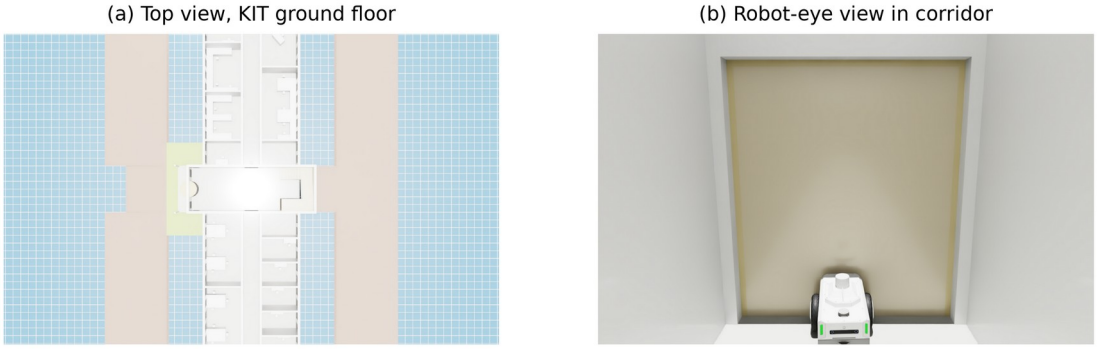


Fig. 9. The KIT scene in Isaac Sim: view from above and robot eye level.

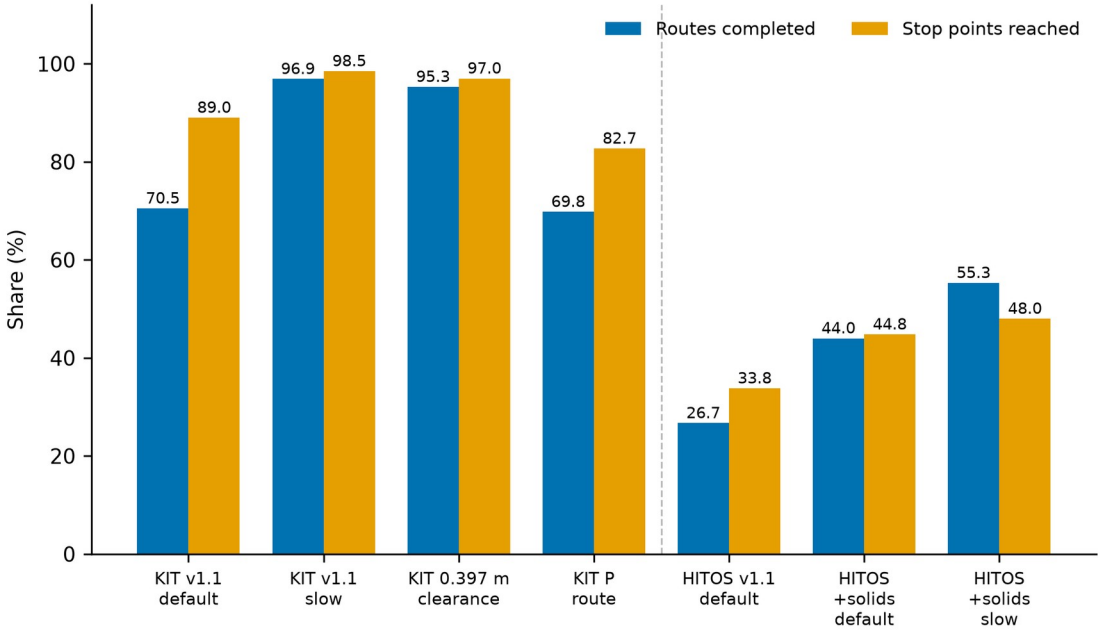


Fig. 10. Route completion and stop reach rate of physical driving in Isaac Sim, by route set.

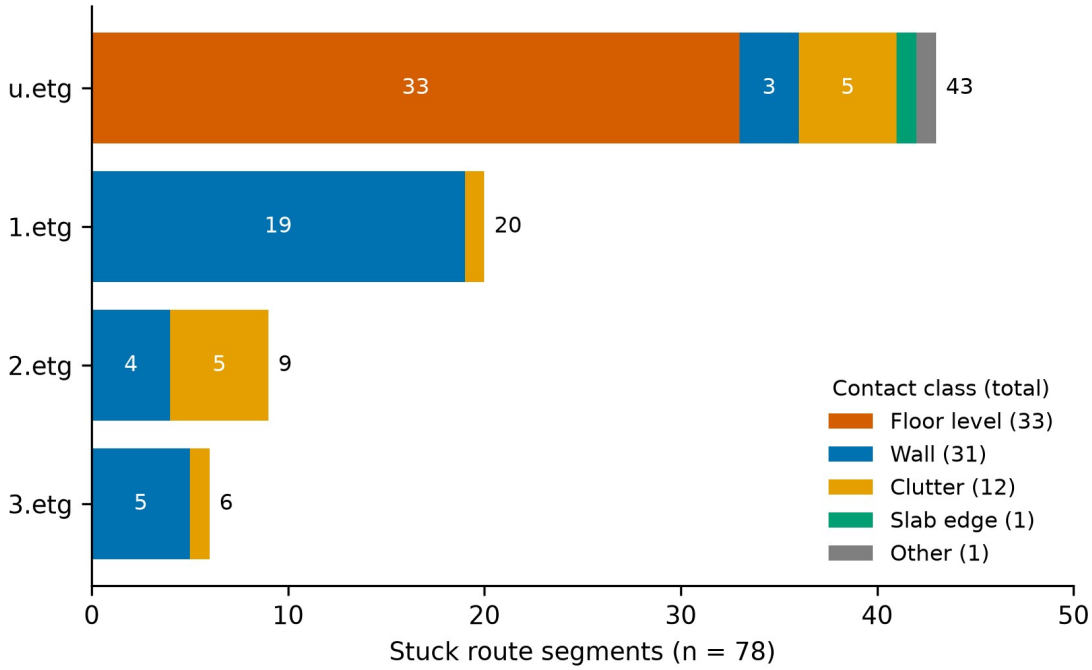


Fig. 11. Contact types at the stalls of driving routes in the HITOS building, by storey.

5. Discussion

5.1 Abstaining changes what an inspection report means

The binary judge's wrong verdicts come mainly from treating missing observations as compliant. In all three buildings, more than half of violations under ideal sensing (with planner P) were reported as compliant by the binary judge and were not by the three-state judge, which returned INDETERMINATE. The cost is that many tasks stay undecided: at 20 % clutter, ECR fell to 0.215–0.314 even for planner P. A three-state report does not make the inspection complete; it says what remains to be checked. In practice this means a person must follow up on the undecided tasks, and the workload this creates was not measured here.

5.2 What the reasons tell an inspector

An undecided task is useful to an inspector only if it says what must be done next, and the four reasons point to different actions. *Unreachable* means the robot could not get to a pose from which the evidence could be collected; the follow-up is a person going there, and the cause is part of the finding. In our runs it was the most frequent reason in all three buildings (Section 4.6), and apart from a few structural and intermittent tasks all of it was caused by clutter. Clutter that stops the robot is not the same as an obstruction that

narrows an escape route below its legal width, because the robot's clearance is not the legal width, but it marks where an R1 obstruction is likely and is worth reporting as such rather than discarding. Structural unreachability (31 tasks in the development building and 14 in HITOS, none in KIT) points to the robot's movement assumption or to the walkable-space model, not to the building's condition, and should be listed once per building rather than in every report. *Constraint unmet* means the robot observed the target but not well enough for the rule: the measurement lay near the threshold or the view did not meet the rule's standoff or line-of-sight condition. The follow-up is a direct measurement, such as a door opening angle or a corridor width, and it is the reason that grew as sensing degraded (Section 4.6). *Not observed* means that some or all of the task's targets were never viewed, and the task can be repeated on the next run. *Missing attribute* cannot be resolved by any observation: a door whose fire rating or closing type is unknown remains undecided however often it is visited, and the follow-up is a request for documents, addressed to the owner or the model author rather than to the inspector.

The reasons therefore separate what a site visit can settle from what only records can settle. For a regulator or building owner who receives a robot's report, this separation matters more than the share of decided tasks: a binary report that lists an unobserved door as compliant certifies a condition that no one saw, and nothing in the report shows which entries are of that kind. For the doors with unknown fire rating, the binary judge returned a verdict in every case (Section 4.7), none supported by evidence about the door's legal status. Whether the inspection forms used in Korean practice can record an abstention with its reason, or only a pass, a deficiency and "not applicable", was not examined here; if they cannot, the reasons would have to be carried in the remarks of the form.

5.3 Replanning, not ordering, produces the gain

The ablations separate the components of P. Replanning accounts for the ECR gain; re-checking accounts for fewer false alarms and has no effect on ECR. Visiting order had no effect on ECR. Exhaustive coverage matched P's ECR but travelled 4 to 6 times as far. The practical reading is that an inspection robot benefits more from reacting to failed observations than from a better tour, and that coverage is a valid, expensive alternative when battery or time allows. Removing replanning slightly reduced false alarms, so the statement that P reduces false alarms should be attributed to re-checking; the cause of the small reduction without replanning was not examined.

5.4 What the driving test does and does not support

The driving test changed the study: it showed that the first simulation allowed routes the robot body could not pass, which led to v1.1. After that

change, KIT routes were largely drivable, including those of P, and failures there were mostly explained by tracking error against a 0.05 m margin. HITOS, the building not used for that change, did much worse. Adding stairs, ramps and columns to the world removed the stair contacts and raised completion from 40 to 66 of 150, and slow control to 83, but failures remained. Contact records attribute them mainly to floor levels the lightweight world does not model and to walls, including one location where the world cut an opening. We therefore do not claim that the lightweight model transfers generally. The contrast between KIT and HITOS is itself a finding: a walkable-space model that performed well in one building failed in another, and a physics check on a second building was needed to reveal it. The hypothesis effects in HITOS did not change materially when the world was corrected, which suggests that the relative comparisons are more robust than the absolute ECR values, though this was tested for one correction in one building.

5.5 Comparison with the closest robotic system

Tuomisto et al. [30] did what this study did not: a real robot with real perception, a vision-language model answering questions on presence and installation, and a test in a real building with BIM–reality discrepancies. This study adds what theirs did not: tasks compiled from statutory clauses and evaluated on held-out buildings, a judgment that abstains with a reason, comparison with ordering, coverage and ablation baselines, and a contact-level physics check. The two designs meet at two cases from their study that show how the handling of missing evidence changes the report.

First, in their system a viewpoint reported as unreachable is removed from the inspection queue, so the target receives no outcome. Here such a task stays in the report as INDETERMINATE with reason *unreachable* and stays in the denominator of ECR. Because unreachability was the most frequent reason for abstaining in all three buildings (Section 4.6), the difference is not marginal: a report that dropped these tasks would show a higher completion on a smaller set of tasks without stating which tasks were removed, and the clutter that caused most of them (Section 5.2) would not appear in the report.

Second, in their real-world test an extinguisher installed on the far side of a column from its BIM position was recorded as not present from an image counted as a successful capture. In compliance terms this is a false violation. The method here would not necessarily avoid it. The observation gate tests the line of sight to the position in the prior, and from the planned pose that line of sight was clear, so the gate would accept the view, the sensor would report the fixture absent, and both judges would report non-compliance unless the re-check from a second pose happened to see the fixture. This is the mechanism of the R4 exception in Section 4.5, where the two judges produced the same number of false alarms for exit signs. Our scenarios

removed fixtures but did not move them, so this case was not tested. What the three-state formulation adds is that “absent” and “not seen” become separate outcomes that the gate must decide between; deciding between them for a displaced fixture requires searching around the prior position, which neither system did.

The two systems also differ in what triggers replanning. Tuomisto et al. replan when view quality falls below the planned score; P replans when the rule’s observation constraints (standoff range, line of sight, share of a corridor segment in view) are not met, so the trigger is what the verdict needs rather than image quality. Their comparison was with ablated variants of their own system; here, changing only the visiting order did not change ECR, and coverage matched P only at 4 to 6 times the travel (Section 4.10), which indicates where the gain of replanning comes from.

5.6 Task generation depends on model information

Recall on HITOS was limited by name-based rules: exterior doors without an “Entrance” keyword were not recognised as exits, and doors between non-corridor rooms treated as egress by the expert were missed. IFC models contained no extinguishers or signs and no fire ratings, which is why R2 tasks stayed undetermined. Better results here would depend as much on model information requirements as on the compiler. Four requirements follow directly from the results, and they could be written into the information requirements an owner sets for the model handed over for operation:

- **Exits marked as exits.** An exit door should carry an explicit property or classification rather than a type name. This would have removed the third cause of the HITOS misses (Section 4.2). The first two causes, doors between rooms and doors on small corridors that the expert counted as part of the escape route, need the escape route itself in the model, for example as a group of spaces.
- **Fire rating on doors.** Pset_DoorCommon.FireRating exists in the schema but carried no value in the three IFC models. With a value from the applicable fire-door classes, a stair-adjacent door could be decided as a fire door or not; without it, all such doors became undetermined tasks (KIT 11, HITOS 40).
- **Closing type on fire doors.** Whether a fire door is kept closed or closes automatically on detection decides which observation counts as a violation, and IFC has no standard property for it, so R2 reads it from an overlay (Section 3.2). A property for it would remove one source of *missing attribute*.
- **Fire safety equipment as elements.** Extinguishers and signs placed in the model with their positions would let the compiler check the installed equipment instead of generating it, and would remove the

synthetic-placement threat in Section 5.7. Because the models contain neither these elements nor explicit escape routes, the IFC-only baseline produced no R1, R3 or R4 task in any building.

Where a model lacks these items, the three-state output names which attribute is missing for which door, and that list can serve as a model-checking report for the owner.

5.7 Limitations and threats to validity

Fixture placement (internal validity). Extinguisher and sign positions were generated by the compiler's own R3 and R4 logic, because none of the models contains them. They were not placed or checked by an expert, they meet R3 and R4 by construction, and violations enter only through removal. Because the same rule logic informs both placement and inspection, these results may be favourable to the method: equipment placed by a designer may stand where it is harder to see, and displaced equipment (Section 5.5) was not simulated.

Single reference standard (construct validity of E1). Precision and recall were scored against one reference, A2's. The second annotator agreed closely on route spaces (kappa 0.84 and 0.95) and on doors needing an exit sign (0.93, 0.85) but recorded door-front checks at a different granularity (door-level kappa 0.00, 0.14). R1 scores at door level therefore depend on A2's choice to list every door, and no adjudicated reference was formed.

Storey clusters (statistical conclusion validity). There are only 4 to 7 storey clusters per building (7, 4 and 5). A cluster bootstrap with so few clusters can understate uncertainty, and the Holm-adjusted p-values reach the resolution limit of the bootstrap ($p = 0.0003$). The support for H1–H3 rests more on the same sign in all three buildings and in both world versions (Section 4.9) than on the p-values. The planner comparison is secondary, and H4 was exploratory and not supported.

Lightweight world (construct validity of ECR and travel). The lightweight world assumes a flat floor and does not model slab extent or level, columns, stairs or ramps, and door openings may not match the physical scene. In HITOS, contact records attributed 33 of 78 stuck route segments to floor level and 31 to walls. These were identified but not corrected, so absolute ECR in that building is less certain than in KIT; the hypothesis effects changed little under the one correction tested (Section 4.11). Reachability and path length are computed with the clutter of the scenario, standing in for local navigation; a robot that finds a blocked passage only on arrival would travel further. This applies to all planners alike, and its effect on the comparisons was not tested.

Perception. Measurement is an oracle with Gaussian noise. No detector was tested, and Isaac runs include no sensing, so evidence collection has not been validated physically.

Buildings (external validity). Three buildings, two of them foreign and judged under an assumption of Korean law. KIT informed the post-hoc v1.1 change, so only HITOS is unused for that change, and the world correction for HITOS was itself post hoc.

Scene and control. In E4, doors are passable and furniture is removed; doors were not operated by the robot. Pure pursuit without obstacle avoidance is deliberately simple. A local planner may raise completion, but that would be a different test.

Preregistration. None on an external registry; the plan, code and reference were fixed by hash before the held-out runs.

Rules. The 24 legal-basis entries were checked against the original texts (article numbers and excerpts all matched; tag rules-v1.0). Two exemptions for exit and corridor signs (NFTC 303 2.8.1.3 and 2.8.2.2) are not encoded, so the rules may require a sign where the standard would exempt it; this errs towards non-compliance and does not create missed violations. Stair-landing signs (2.3.1.3) and roof doors are outside the scope of this paper.

5.8 Future work

Modelling slab extent and level in the walkable world, and checking door openings against the physical scene, come first. After that: a real detector in the loop, displaced as well as removed equipment in the scenarios, operating doors, field tests in a real building, an adjudicated reference from more than one annotator, a model that carries the information listed in Section 5.6, and a detection-confidence model for exit signs, whose false alarms the margin rule cannot withhold.

6. Conclusion

Inspection tasks were generated from Korean fire regulations and IFC models, with precision of at least 0.83 and recall of at least 0.60 across rules on two held-out buildings. In simulation, a three-state judgment reduced wrong verdicts compared with a binary judgment, replanning raised evidence completion, re-checking reduced false alarms, and exhaustive coverage matched the replanning planner’s completion only at 4 to 6 times the travel. The reasons attached to undecided tasks separate what a further site visit can settle from what only building records can settle. A physics-based test showed that routes were largely drivable in one building but only partly in another. The relative findings held under the corrections tested, while

absolute evidence completion should not be taken as what a physical robot would achieve.

CRediT authorship contribution statement

[Author name]: Conceptualization, Methodology, Validation, Investigation, Writing – review & editing, Supervision. **[Confirm roles before submission.]**

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. **[Confirm.]**

Funding

[Funding source and grant number, or "This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors."]

Data availability

[Choose one before submission.]

[A. Data shared:] The code, rule files, configuration, simulation results and decision records of this study are available in a public repository: [repository name, DOI or URL]. The IFC models of the two held-out buildings are available from their original publishers: [sources].

[B. Data not shared:] [Reason the data cannot be shared, e.g. the development building's IFC model and its derivatives are private.] The code, rule files, configuration, simulation results and decision records for the two public held-out buildings are available in [repository / on request to the corresponding author].

The development building's model and its derivatives cannot be shared because they are private.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the authors used Claude (Anthropic) to draft text, write analysis code and check numbers against the result files, and three AI deep-research tools **[tool names]** to search the literature. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article. The second reference annotator used a large language model only to format a summary of their own judgments.

Acknowledgements

The authors thank the licensed fire protection professional engineer and the fire safety practitioner who wrote the expert references.

References

[1] Home Office, Fire prevention and protection statistics, england, april 2025 to march 2026, (2026). https://www.gov.uk/government/statistics/fire-prevention-and-protection-england-year-ending-march-2026/fire-prevention-and-protection-statistics-england-april-2025-to-march-2026.

[2] National Fire Agency of Korea, 2023 상반기 소방사범 일제단속 결과 (results of the national fire-law enforcement campaign, first half of 2023), (2023). https://m.korea.kr/briefing/pressReleaseView.do?newsId=156581309.

[3] Korean Fire Protection Association, 제천 스포츠센터 화재 (jecheon sports centre fire, 21 december 2017), 2017. https://www.kfpa.or.kr/mem/pdf_file/B/14/20171221_%EC%A0%9C%EC%B2%9C_%EC%8A%A4%ED%8F%AC%EC%B8%A0%EC%84%BC%ED%84%B0_%ED%99%94%EC%9E%AC.pdf.

[4] M. Moore-Bick, Grenfell tower inquiry: Phase 2 report, volume 1, Grenfell Tower Inquiry, 2024. https://assets.publishing.service.gov.uk/media/66d817aa701781e1b341dbd3/CCS0923434692-004_GTI_Phase_2_Volume_1_BOOKMARKED.pdf.

[5] C. Eastman, J. Lee, Y. Jeong, J. Lee, Automatic rule-based checking of building designs, Automation in Construction 18 (2009) 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002.

[6] W. Solihin, C. Eastman, Classification of rules for automated BIM rule checking development, Automation in Construction 53 (2015) 69–82. https://doi.org/10.1016/j.autcon.2015.03.003.

[7] J. Zhang, N.M. El-Gohary, Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking, Automation in Construction 73 (2017) 45–57. https://doi.org/10.1016/j.autcon.2016.08.027.

[8] L. Jiang, J. Shi, C. Wang, Multi-ontology fusion and rule development to facilitate automated code compliance checking using BIM and rule-based reasoning, Advanced Engineering Informatics 51 (2022). https://doi.org/10.1016/j.aei.2021.101449.

[9] F. Yang, J. Zhang, Prompt-based automation of building code information transformation for compliance checking, Automation in Construction 168 (2024) 105817. https://doi.org/10.1016/j.autcon.2024.105817.

[10] O. Iversen, L. Huang, Leveraging large language models for BIM-based automated compliance checking, Automation in Construction 182 (2026) 106707. https://doi.org/10.1016/j.autcon.2025.106707.

[11] H. Lee, J.-K. Lee, S. Park, I. Kim, Translating building legislation into a computer-executable format for evaluating building permit requirements, Automation in Construction 71 (2016) 49–61. https://doi.org/10.1016/j.autcon.2016.04.008.

[12] K. Kincelova, C. Boton, P. Blanchet, C. Dagenais, Fire safety in tall timber building: A BIM-based automated code-checking approach, Buildings 10 (2020) 121. https://doi.org/10.3390/buildings10070121.

[13] S. Fischer, C. Schranz, H. Urban, D. Pfeiffer, Automation of escape route analysis for BIM-based building code checking, Automation in Construction 156 (2023) 105092. https://doi.org/10.1016/j.autcon.2023.105092.

[14] Y. Wang, Y. Liu, H. Cai, J. Wang, X. Zhou, An automated fire code compliance checking jointly using building information models and natural language processing, Fire 6 (2023) 358. https://doi.org/10.3390/fire6090358.

[15] D. Chen, L. Chen, Y. Yang, Q. Lu, C. Hancock, R. Lock, S. Sølvsten, A knowledge-driven approach for automated fire safety compliance checking in operational buildings, Knowledge-Based Systems 336 (2026) 115156. https://doi.org/10.1016/j.knosys.2025.115156.

[16] T. Beach, J. Yeung, A. Ghoroghi, Y. Rezgui, Moving automated compliance checking to the operational phase of the building life-cycle: Analysis and feasibility study in the UK, International Journal of Construction

Management 25 (2024) 840–849.
https://doi.org/10.1080/15623599.2024.2366727.

[17] S. Malsane, J. Matthews, S. Lockley, P.E.D. Love, D. Greenwood, Development of an object model for automated compliance checking, Automation in Construction 49 (2015) 51–58.
https://doi.org/10.1016/j.autcon.2014.10.004.

[18] T. Bloch, R. Sacks, Clustering information types for semantic enrichment of building information models to support automated code compliance checking, Journal of Computing in Civil Engineering 34 (2020).
https://doi.org/10.1061/(asce)cp.1943-5487.0000922.

[19] P. Schönfelder, A. Aziz, F. Bosché, M. König, Enriching BIM models with fire safety equipment using keypoint-based symbol detection in escape plans, Automation in Construction 162 (2024) 105382.
https://doi.org/10.1016/j.autcon.2024.105382.

[20] S. Bigdeli, P. Pauwels, S. Verstockt, N. Van de Weghe, B. Merci, Advancing automated fire safety code compliance: Automatic annotation of BIM door elements, Fire Technology 62 (2026).
https://doi.org/10.1007/s10694-026-02010-1.

[21] E. Hjelseth, B. Li, Use of "reasoner" for automated building code compliance assessment, in: Proceedings of the 38th International Conference of CIB W78, Luxembourg, 2021: pp. 138–147.
https://cumincades.scix.net/pdfs/w78-2021-paper-014.pdf.

[22] S. Kim, M. Peavy, P.-C. Huang, K. Kim, Development of BIM-integrated construction robot task planning and simulation system, Automation in Construction 127 (2021). https://doi.org/10.1016/j.autcon.2021.103720.

[23] K. Kim, M. Peavy, BIM-based semantic building world modeling for robot task planning and execution in built environments, Automation in Construction 138 (2022). https://doi.org/10.1016/j.autcon.2022.104247.

[24] S. Park, S. Yoon, S. Ju, J. Heo, BIM-based scan planning for scanning with a quadruped walking robot, Automation in Construction 152 (2023).
https://doi.org/10.1016/j.autcon.2023.104911.

[25] Z. Chen, C. Song, B. Wang, X. Tao, X. Zhang, F. Lin, J.C.P. Cheng, Automated reality capture for indoor inspection using BIM and a multi-sensor quadruped robot, Automation in Construction 170 (2025).
https://doi.org/10.1016/j.autcon.2024.105930.

[26] J. Chen, W. Lu, Y. Fu, Z. Dong, Automated facility inspection using robotics and BIM: A knowledge-driven approach, Advanced Engineering Informatics 55 (2023) 101838. https://doi.org/10.1016/j.aei.2022.101838.

[27] A. Tandon, J. Stührenberg, K. Smarsly, Automated building inspections coupling building information modeling and behavior trees, in: Computing in Construction, 2025. https://doi.org/10.35490/EC3.2025.360.

[28] A. Aziz, P. Herbers, H. Bayer, M. König, Fully autonomous fire safety equipment inspection missions on a legged robot, in: Computing in Civil Engineering 2023, 2024: pp. 804–812. https://doi.org/10.1061/9780784485224.097.

[29] A. Aziz, N. Gard, M. König, P. Eisert, J.H. Heinbach, L. Trost, Visual fire safety inspection framework using computer vision algorithms, Journal of Computing in Civil Engineering 39 (2025). https://doi.org/10.1061/JCCEE5.CPENG-6492.

[30] J. Tuomisto, Y. Zheng, I. Chauhan, O. Seppänen, J. Pajarinen, Automating on-site object inspection with a quadruped robot and BIM, Automation in Construction 181 (2026). https://doi.org/10.1016/j.autcon.2025.106571.

[31] Z. Chen, H. Wang, K. Chen, C. Song, X. Zhang, B. Wang, J.C.P. Cheng, Improved coverage path planning for indoor robots based on BIM and robotic configurations, Automation in Construction 158 (2024) 105160. https://doi.org/10.1016/j.autcon.2023.105160.

[32] T. Duraimany, F. Bosché, Automated planning, execution, and re-planning of terrestrial laser scanning in the built environment, in: Lecture Notes in Civil Engineering, 2025: pp. 74–85. https://doi.org/10.1007/978-3-032-09040-9_7.

[33] M.A. Gopee, S.A. Prieto, B. García de Soto, Improving autonomous robotic navigation using IFC files, Construction Robotics 7 (2023) 235–251. https://doi.org/10.1007/s41693-023-00112-8.

[34] J. Stührenberg, A. Tandon, K. Smarsly, GPP-BIM — global path planning for robot navigation using building information models, Advanced Engineering Informatics 74 (2026). https://doi.org/10.1016/j.aei.2026.104809.

[35] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, D. Batra, Sim2Real predictivity: Does evaluation in simulation predict real-world performance?, IEEE Robotics and Automation Letters 5 (2020) 6670–6677. https://doi.org/10.1109/LRA.2020.3013848.