Research Square

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

Semantic Planning with Large Language Models and Vision-Language Models for Collaborative Robots: A Review

Osama Ali Khan

kosamaali@hotmail.com

Southern University of Science and Technology

Abuzar Ghaffari

University of Engineering and Technology Peshawar

Systematic Review

Keywords: Collaborative robots, Semantic planning, Large language models, Vision-language models, Human-robot collaboration, Uncertainty quantification, Shared autonomy

Posted Date: September 1st, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10643317/v1

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

Additional Declarations: No competing interests reported.

Semantic Planning with Large Language Models and Vision-Language Models for Collaborative Robots: A Review

Osama Ali Khan[1,2,*] **and Abuzar Ghaffari** [3]

[1]*Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, 518055, China*

[2]*Department of Mechatronics Engineering, National University of Sciences and Technology, Islamabad, 44000, Pakistan*

[3]*Department of Mechatronics Engineering, University of Engineering and Technology Peshawar, Peshawar 25000, Pakistan*

**Correspondence: 12431164@mail.sustech.edu.cn*

Abstract

Large Language Models (LLMs) and Vision-Language Models (VLMs) now allow robots to translate natural-language instructions and visual observations into executable task plans. Despite these advances, the resulting plans remain incompletely grounded in the robot's capabilities and affordances, in scene geometry and physical constraints, and in the collaborator's changing state. The deficiency is most damaging over long horizons, where a local planning or execution error propagates through the remaining actions and compromises the task without announcing itself. Reliable collaboration therefore demands not only higher per-step accuracy but the means to detect planning uncertainty and constraint violations and to respond to them. This survey reviews semantic planning for collaborative robotics at the interface of foundation-model reasoning, multimodal perception, task and motion planning, and human-robot interaction. We formalize collaborative semantic planning as a constrained grounding problem relating instruction, observation, robot capability, task state, and human state, and assess surveyed approaches against five explicit conditions: groundedness, feasibility, safety, legibility, and calibrated abstention. We organize the field into a taxonomy of architectures, indexed by the depth at which semantics enter the planning and control hierarchy, and argue that capability and inspectability have moved in opposite directions across successive generations of architectures. We then state a reliability decomposition separating per-step accuracy from detection quality, which gives, under an explicit independent-step model, the condition under which improving detection dominates improving accuracy. We close with the open challenges and a roadmap for incorporating uncertainty estimation, plan verification, and recovery into next-generation collaborative architectures.

Keywords: Collaborative robots, Semantic planning, Large language models, Vision-language models, Human-robot collaboration, Uncertainty quantification, Shared autonomy.

1. Introduction

1.1 Motivation, problem statement, and timeliness

In the industrial era of robotics up until now, safety assurance of the robot has been about separation. This meant that the robot was in a cell, that the cell had a fence, and that the fence was interlocked. The question of the robot's program's correctness was a side issue because the geometry would have spatially confined any incorrectness. By definition, a collaborative robot works in a space shared with a human, so what used to be assured by a fence must now be guaranteed by the robot's own perception, planning, and control. (Haddadin et al., 2009; Lasota et al., 2017). This transition shifted the reliability burden upstream. When a robot executes one specific motion for a year, the cost of programming it accurately becomes negligible. However, when the same robot needs to help with a different assembly variant each week, be redeployed by a line supervisor instead of an integrator, and respond to spoken requests from someone who has never written code, the cost of programming increases significantly. This makes instruction the main hurdle. (Ajoudani et al., 2017; Villani et al., 2018). Surveys of industrial collaborative deployment frequently highlight programming effort and inflexibility as the main barriers to adoption, rather than hardware capability (Bauer et al., 2008; Villani et al., 2018).

Natural language is a clear interface and has been a focus in robotics for over a decade (Matuszek et al., 2013; Tellex et al., 2011, 2020). In 2020, models capable of understanding and generating language without task-specific engineering were introduced. Transformer-based language models trained at scale (Brown et al., 2020; Vaswani et al., 2017), have shown in-context learning and multi-step reasoning (Wei et al., 2022; Yao et al., 2022). This made it possible to turn a vague human request into a structured list of steps for a robot to follow. At the same time, vision-language models connected image and text representations at a large scale (Alayrac et al., 2022; J. Li et al., 2023; H. Liu et al., 2023; Radford et al., 2021). This development enabled open-vocabulary detection and segmentation (Kirillov et al., 2023; S. Liu et al., 2025; Minderer et al., 2022; Zhou et al., 2022) and created the possibility of a perception system in which the object vocabulary was not limited at design time.

The combination of these two abilities created the field this review examines. Systems like SayCan (Ahn et al., 2022), Inner Monologue (W. Huang, Xia, et al., 2022), Code as Policies (Liang et al., 2023), ProgPrompt (Singh et al., 2022), PaLM-E (Driess et al., 2023), and the RT and OpenVLA model families (Brohan et al., 2022, 2023; M. J. Kim et al., 2024; O'Neill et al., 2024; Octo Model Team, 2024), showed that a foundation model can break down a natural-language goal into robot actions on actual hardware. There is a growing focus on predictive world models linked with these systems (F. Wang et al., 2026). Open-source policies are being developed for real-time execution on robots rather than for offline testing (Cai, Guo, et al., 2026). The instruction problem is no longer just a research topic. It has become an engineering challenge with a clear deployment goal, which sharpens the question of reliability that this review addresses. Three application areas have adopted this work quickly: flexible manufacturing and assembly, where collaborative robots handle tasks like fixturing, kitting, and inspection alongside human workers (Villani et al., 2018); logistics and warehouse picking, where there is a lot

of variety in instructions and a wide range of objects; and assistive and domestic robotics, where users typically have less expertise and safety is most critical (J. Wu et al., 2023).

The reliability problem this creates can be stated precisely. Collaborative robots operating in shared spaces alongside humans remain unable to translate open-ended natural-language instructions into safe, reliable, and adaptive actions. The foundation model-based planners create plans that are linguistically coherent and semantically sensible but often disconnected from the robot's capabilities, scene geometry, and human intentions. The particular character of the deficiency needs attention, because it is systematically misrepresented. It is not that LLMs and VLMs make bad plans; all planners do when there is a distribution shift. It is the modality of the mistake. In the case of a classical planner working from an incomplete domain model, there is a clear failure: it will return no plan, or a plan whose preconditions fail at the first action, rendering it inapplicable, and whose failure is easily detected by inspecting the state. A foundation-model planner fails silently. It generates a valid sequence of actions, each semantically well-formed, whose flaw a non-existent referent, an ungraspable grasp, a step assuming an object has been moved, a motion requiring that space be traversed while the human is already moving their arm to reach it is invisible until execution time, and even then, invisible if no part of the pipeline is looking for whether the step was completed successfully (Kambhampati et al., 2024; Z. Liu et al., 2023; Valmeekam, Marquez, et al., 2023).

The semantic planning of a collaborative robot, thus, must achieve four properties concurrently: grounding, a mapping of linguistic referents to physical objects that the robot has indeed perceived; feasibility, a certainty that every action plan falls into the capability of the robot's kinematics and dynamics from its present configuration; safety, which is the fulfillment of the separation, force, and velocity requirements dictated by the operational mode of collaboration in accordance with ISO/TS 15066 (Lasota et al., 2017; Zacharaki et al., 2020), and legibility, the characteristic of the human partner being able to read off the intention of the robot to act in time to cooperate with it (Dragan et al., 2013; Hoffman, 2019). Present-day systems achieve the first property partially, the second fitfully, the third by simply outsourcing it to a reactive safety layer below the planner, and the fourth hardly at all.

Three developments distinguish the present moment from the situation in 2022 and make the question urgent rather than merely open.

1. First, there is architectural convergence. From 2022 to 2026, the trend has been moving from models emitting text, which were then parsed by a separate grounding module (Ahn et al., 2022; W. Huang, Abbeel, et al., 2022), through models that emitted executable code (Liang et al., 2023; Singh et al., 2022), and then to vision-language-action models that emit action tokens (Brohan et al., 2023; M. J. Kim et al., 2024; Octo Model Team, 2024). Each step tightened the coupling between semantics and control, and each step also removed an interface at which a plan could previously be inspected. The most capable systems are now the least transparent, which is precisely the wrong direction for a safety-relevant application.

2. Second is empirical maturity. There is enough experience with real hardware evaluations by now to be able to say that success rates on carefully selected benchmarks cannot withstand contact with messy environments, and the remaining failure modes are those of grounding and long-horizon consistency rather than of low-level control (Firoozi et al., 2023; Valmeekam, Marquez, et al., 2023; Valmeekam, Olmo, et al., 2023). The field has enough negative evidence to reason from, which was not true in 2022.

3. Thirdly, there is a regulatory challenge. The EU AI Act and the NIST AI Risk Management Framework for all high-risk AI applications put forward requirements regarding documentation, risk assessment, and oversight (B. Li et al., 2023), and a foundation model responsible for defining the physical actions of a robot working in proximity to a human is definitely one of them. In addition, the standards for robot safety, ISO 10218 and ISO/TS 15066, assume deterministic and analyzable control algorithms, an assumption not satisfied by a stochastic generation planner (Zacharaki et al., 2020). The gap between what these two regimes require and what current semantic planners can evidence is, at present, unbridged, and it is the practical reason that most LLM-driven collaborative systems remain in laboratories.

1.2 Scope, contributions, and positioning

This survey discusses collaborative systems involving physical embodiments, manipulator arms, mobile manipulators, and dual-arm systems that are co-located with the human in the environment and use natural language as an instruction medium. We exclude disembodied dialogue agents, tele-operation without autonomous planning, and traditional cage-style industrial automation. Navigation is treated as in scope only where it shares the grounding, verification, or human-awareness machinery under discussion. Open-vocabulary spatial memory, semantic anomaly detection, and social motion planning all transfer directly to shared-workspace manipulation, and are excluded where they do not, as in pure metric path planning. We state this asymmetry explicitly because navigation work is cited at several points below, and an undeclared boundary would be inconsistent. Simulation-only work is included where it establishes a method, but is marked as such, because the sim-to-real gap is itself one of our findings.

The contributions are seven in number:

1. Characterization of semantic planning for collaborative robots as a constrained grounding problem, using safety and legibility predicates explicitly (Sect. 3.4, Definition 1)

2. Problem analysis, including causes, population affected, short and long-term consequences of the semantic reliability gap problem, as well as reasons for failure of the remedies already provided to the problem (Sect. 2, Table 2)

3. Classification of architectures of LLM- and VLM-based planning into five classes with comparative analysis along consistent criteria (Sect. 4, Fig. 2, Tables 4 and 5)

4. Proposed five implementable solutions with analysis of each solution, including its strengths, weaknesses, feasibility, resource requirements, initial step, and place within the robotic system pipeline (Sect. 8, Table 8)

5. Differentiation of open challenges based on which challenge requires an engineering approach and which requires a conceptual one (Sect. 9).

6. A reliability decomposition separating per-step accuracy from detection quality, with a proposition giving, under an explicitly stated independent-step model, the condition under which improving detection dominates improving accuracy (Sect. 6.1, Proposition 1, Fig. 4). This converts a claim that the surrounding literature makes qualitatively into one that can be evaluated for a given platform once the constituent quantities are measured.

7. An explicit positioning of this review against the four closest existing surveys, stating what each covers and what remains uncovered (Sect. 1.2, Table 1).

No review stands alone, and the reader is entitled to know what this one adds. Four recent surveys overlap its territory. (Firoozi et al., 2023) survey foundation models in robotics broadly, across perception, planning, and control, without a collaborative or human-in-the-loop focus. The systematic review of vision-language-action models by (Din et al., 2025) catalogs architectures, datasets, and simulators at a scale we do not attempt, but treats the human as absent from the loop and evaluation as a matter of task success. The survey of large language models in multi-robot systems by (P. Li et al., 2026) is the closest to this review in its attention to deployment constraints, but its unit of analysis is the robot team rather than the human-robot dyad. And the human-robot collaboration (Lasota et al., 2017; Villani et al., 2018) surveys that predate the foundation-model era treat safety, fluency, and trust rigorously while necessarily omitting language-driven planning altogether. The gap all four leave open is the one this review occupies: the collaborative dyad, assessed against explicit conditions of groundedness, feasibility, safety, legibility, and abstention, with undetected failure rather than aggregate accuracy as the organizing concern. Table 1 states this comparison in full.

Table 1. Positioning of this review against the four closest existing surveys

Survey	**Unit of analysis**	**Human in the loop**	**Failure detection treated**	**Safety standards treated**	**Quantitative reliability model**
Foundation models in robotics (Firoozi et al., 2023)	Robot capability stack	No	Briefly	No	No

VLA systematic reviews (Din et al., 2025)	Model architecture and data	No	As a benchmark metric	No	No
LLMs for multi-robot systems (P. Li et al., 2026)	Robot team	Intervention only	Briefly	No	No
Pre-foundation-model HRC surveys (Lasota et al., 2017; Villani et al., 2018)	Human-robot dyad	Yes	Fault detection only	Yes	No
This review	**Human-robot dyad**	**Yes**	**Central**	**Yes**	**Yes (Proposition 1)**

1.3 Review methodology and source selection

The sources were identified through systematic searches in IEEE Xplore, Springer Nature Link, Scopus, Web of Science, ACM Digital Library, and arXiv databases from January 2020 to August 2026, along with backward and forward citation chaining from the most-cited sources in each cluster. The main Boolean search string is the following:

("collaborative robot" OR "cobot" OR "human-robot collaboration" OR "manipulation") AND ("large language model" OR "LLM" OR "vision-language model" OR "VLM" OR "foundation model") AND ("task planning" OR "semantic planning" OR "grounding" OR "replanning").

Three further questions focused on areas insufficiently covered by the primary question, namely: safety and standards for collaboration; uncertainty quantification and calibration in language and vision models and shared autonomy and control arbitration. Foundational papers before 2020 have been specifically included when they introduce ideas that are being used in current literature, such as symbolic planning (Fikes & Nilsson, 1971), legibility (Dragan et al., 2013), trust in automation (Hoff & Bashir, 2015; J. D. Lee & See, 2004). Pre-2020 sources account for 50 of the 150 references retained (33 percent). They fall into four groups: the classical planning and task-and-motion-planning literature that semantic planning is contrasted against; the human-robot collaboration literature on legibility, fluency, intent prediction, and trust, which predates the foundation-model era but is not superseded by it; the calibration and uncertainty-quantification literature on which Sect. 7.3 depends; and the safety and human-factors instruments used in Sect. 7.4. No pre-2020 source is used to characterize the current state of LLM- or VLM-based planning.

The screening process involved the examination of records by title and abstract according to four inclusion criteria: the study deals with embodied robotic action; language or vision-language systems are involved in planning, grounding, or monitoring; there is a task-level evaluation reported; and the study has been peer-

reviewed or is a preprint with sufficient independent citations. Exclusions included any linguistic evaluation without embodiment, position papers without technical information, and duplicates in which one record was a preprint and the other its proceedings counterpart, in which case the published paper was kept. The full-text assessment checked whether the evaluation was conducted on physical hardware, whether failures were discussed, and whether humans were involved in the process. **Figure 1** presents the numbers.

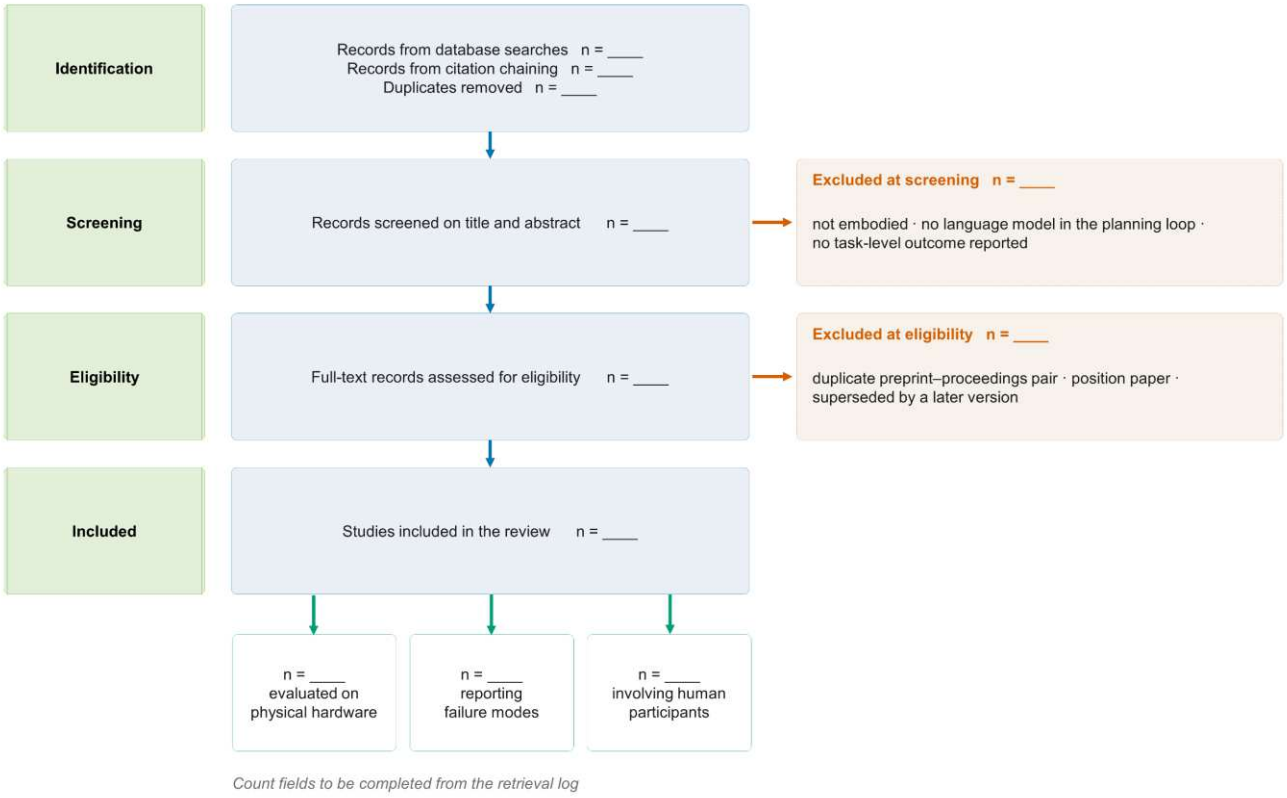


Figure 1*. Source identification, screening, eligibility, and inclusion. Four stages, with exclusion reasons at each screening step and, at the foot, the three sub-counts a reader needs in order to weigh the evidence base: studies evaluated on physical hardware, studies reporting failure modes, and studies involving human participants. Counts are reported for each stage; the three sub-counts at the foot are reported as a fraction of the included set.*

Fig. 1 should be read as an audit trail rather than a summary. The three bars below the inclusion box carry the interpretation, and they are deliberately separated from the headline figure: the number of included studies evaluated on physical hardware rather than in simulation, the number reporting any failure mode at all, and the number involving human participants. The third is substantially lower than the first, so the evidence base supporting the human-centered assertions of Sect. 7 is weaker than the evidence base supporting the architectural assertions of Sect. 4, and the two should not be read as carrying equal weight.

Two limitations of this protocol should be stated at the outset. Only English-language sources were retrieved, and the field's reliance on arXiv preprints means that a portion of the surveyed work has not undergone peer review; such sources are marked in the reference list and are excluded from claims about established results.

The remainder of the paper proceeds from this evidence base to the problem analysis that motivates it.

2. Problem Analysis and Context

2.1 Main causes

The reliability problem set out in Sect. 1.1 is not caused by any single defect. Rather, there are seven contributing factors whose interactions are important, since solutions applied to one problem tend to worsen another.

1. The foundational cause is the ***symbol-grounding gap*** (Bender & Koller, 2020; Harnad, 1990). The language model's concept of "the blue bracket" does not refer to an actual object in terms of its pose, weight, or grabbable state rather, it is a probability distribution over text tokens. This means that symbol grounding is needed, and this can be provided by an affordance function (Ahn et al., 2022), an open-vocabulary detector (S. Liu et al., 2025; Minderer et al., 2022), a queryable scene representation of the scene (B. Chen et al., 2023; Gu et al., 2024; Shafiullah et al., 2023), or a scene graph (Rana et al., 2023). Such a hybrid architecture will inherit failure modes from both sides, so when the detector detects a non-existing bracket, the planner will have nothing to disbelieve this with.

2. The second cause is ***open-world perception brittleness***. Open-vocabulary detectors become brittle when faced with occlusion, uncommon poses, low light conditions, and the particular visual clutter of a workcell, but without any indication of that brittleness. A benchmark of vision-language-action policy under scene-induced occlusions reveals significant brittleness exactly in the areas of the work cell where there is the most clutter, with no corresponding decrease in reported confidence (T. Li et al., 2026). An analog of collision grounding shows that vision-language models struggle to represent approaching contact between a robot and a human partner, the particular perceptual ability assumed for safe shared workspaces (J. Wang et al., 2026). Vision-language models are also weak in exactly the spatial relations required for planning manipulation actions, relative depth, containment, support, and reachability, as shown repeatedly in benchmarking (Firoozi et al., 2023; Huang et al., 2023).

3. The third and most consequential cause is the ***absence of calibrated uncertainty***. Neural networks are systematically overconfident, and the overconfidence worsens under distribution shift (Ovadia et al., 2019), as proven by (C. Guo et al., 2017), and language models, in particular, generate fluent output at an equal apparent level of certainty irrespective of how well-founded the underlying proposition might be (Ji et al., 2023). Research on confidence elicitation from language models suggests that verbalized confidence estimates are poorly calibrated (Kadavath et al., 2022; Lin et al., 2023; Xiong et al., 2024).

In the absence of calibrated uncertainty, there will be no point beyond which a robot should pause and query, and hence all grounding errors will become actions.

4. The fourth cause is a ***representational mismatch between token-level generation and continuous control***. Plan is discrete; manipulation is a continuous process with kinematic, dynamic, and contact limitations. Intermediate representations fail to cover that gap in an effective manner: natural language subgoals lack metric information, code (Liang et al., 2023; Singh et al., 2022), requires a properly designed and parametrized API, value maps rely on dense geometry of the scene, and action tokens (Brohan et al., 2023; M. J. Kim et al., 2024), neglect the interface where feasibility can be checked.

5. The fifth cause is ***unmodelled human intent***. In most cases, the human is considered to be an unpredictable dynamic obstacle that needs to be avoided, thus planning considers only the current location of the human, but not their next position (Koppula & Saxena, 2016; Mainprice & Berenson, 2013). In collaborative assembly, however, the human's reach can be predicted based on the task, and if it is not done, the motions will be safe but obstructive.

6. The sixth cause is the ***absence of formal guarantees***. Symbolic and temporal logic planning can provide completeness or correctness in light of the given domain model (Kress-Gazit et al., 2009, 2018), while a generative planner lacks this guarantee, which is why the modern approach to dealing with this situation involves putting a safety layer that works reactively under the planning component. This layer protects against harm, but not against semantic errors, and is invoked right when the planner makes an error.

7. The seventh cause is ***data scarcity for physical interaction***. Language and vision models are trained on web-scale corpora; the amount of data for robot interaction is three to five orders of magnitude lower, even when aggregated (O'Neill et al., 2024). Another flawed approach that was proven unreliable is the pooling of robot data, since according to a 2026 scaling experiment, the aggregation of diverse data from robots can lead to negative transfer and harm rather than improve the performance on the target embodiment (Y. Wang et al., 2026). According to another 2026 survey, the bottleneck is now the data infrastructure and evaluation protocol, not the architecture of the model anymore (Z. Wang et al., 2026). The asymmetry implies that the linguistic prior outweighs the physical one, and that is how the model suggests a linguistic action that is physically impossible.

2.2 Who is affected, and with what consequences

Five populations bear distinct costs, and Table 2 provides per-cause details. ***Operators and line workers*** carry the burden of interruption: retraining a task the robot performed incorrectly, reacting during close calls, and partnering with an agent whose behavior they cannot anticipate, a cognitive load that the trust literature identifies as a principal cause of automation underuse (J. D. Lee & See, 2004; Parasuraman & Riley, 1997). ***Assisted users and patients*** in domestic and rehabilitation settings face the same problem in its steepest form, being less able both to recover from a malfunction and to diagnose it. ***Systems integrators*** cannot price a system whose success

rate is unknown. ***Safety assessors and certification bodies*** face an evidentiary problem, because current standards presume an analyzable control program (Zacharaki et al., 2020) and no accepted method exists for evidencing a stochastic planner. ***Researchers*** bear a scientific cost: non-comparable benchmarks and unreported negative results prevent the field from establishing which methods work.

The immediate consequences are functional. A failed task must be restored by hand, and because the failure is silent, restoration begins only after a physical consequence: a dropped part, a displaced fixture, a collision with the workpiece. Throughput falls below the manual baseline against which pilot deployments are judged. Protective stops become frequent, and each activation is evidence that an unsafe plan reached the controller. Operators cannot tell whether the robot has understood, is deliberating, or has failed, and the accumulation of these effects is pilot abandonment.

Over longer horizons the consequences become structural. Calibrated trust degrades in both directions (Hancock et al., 2011; Hoff & Bashir, 2015) operators who have witnessed confident failures monitor redundantly, negating the efficiency the system was installed to deliver, while operators with long successful histories stop monitoring precisely when vigilance matters most. The absence of standardized evaluation leaves an evidence base that accumulates without converging, and the inability to evidence safety impedes the standards development on which deployment depends. That circularity is, in our assessment, the field's principal structural obstacle.

2.3 Existing solutions, and why they fail

Seven kinds of response exist, and each trades one guarantee for another. ***Classical symbolic planning*** and task-and-motion planning (Fikes & Nilsson, 1971; Fox & Long, 2003; Helmert, 2006; Hoffmann & Nebel, 2001) guarantee correctness within a domain model that must be authored and maintained, which is the cost that natural-language instruction was meant to remove. ***Learning from demonstration*** removes the need for programming but generalizes poorly beyond the demonstrated distribution. ***End-to-end vision-language-action models*** (Brohan et al., 2022, 2023; M. J. Kim et al., 2024; Octo Model Team, 2024) fold semantics and control into a single policy at the cost of an inspectable plan. ***LLM-as-planner*** systems (Ahn et al., 2022; W. Huang, Abbeel, et al., 2022; Song et al., 2023) emit an inspectable natural-language plan but provide no means of checking it against the scene or the robot's kinematics. ***Code-as-policy*** approaches (Liang et al., 2023; Singh et al., 2022) add compositional structure and a checkable artifact, bounded by the expressiveness of the underlying API, and offer no correctness guarantee: automated assessment of a generated program requires either exhaustive enumeration of valid outputs or human inspection (Singh et al., 2023), a concession that anticipates the argument of Sect. 8.1. ***Neuro-symbolic hybrids*** (Guan et al., 2023; Kambhampati et al., 2024; B. Liu et al., 2023; Silver et al., 2024) use the language model to solve or populate a formal model, recovering guarantees at the price of maintaining that model. ***Shared-autonomy interfaces*** (Dragan & Srinivasa, 2013; Javdani et al., 2018; Losey et al., 2018; Selvaggio et al., 2021) keep the human in the loop, trading autonomy for reliability. Sect. 4 develops these as an architectural taxonomy.

The most important observation in this review is that these seven approaches fail for methodological rather than algorithmic reasons, and that the same methodological failures recur across otherwise dissimilar methods.

- ***Evaluation on toy tasks*** dominates. Pick-and-place on a tabletop using objects that are clearly distinct from each other and not cluttered is the most common setup for evaluating a model, and it eliminates exactly the kinds of occlusion, clutter, and referential ambiguity that cause grounding to fail.

- ***Open-loop assumptions*** continue to hold true. The majority of the described systems have proven themselves capable of developing an action plan and implementing it without verifying whether the individual actions produce the desired effect. In this situation, the error made in the early stage of implementing the action plan will remain in effect for the remainder of the plan.

- ***Silent failure*** becomes the direct result, and it is the characteristic that differentiates this generation of planners from previous generations of planners. Because there is no signal during a failure, automatic and manual recovery cannot be initiated at the proper time.

- ***Benchmark phrasing overfitting*** affects performance measurement. The commands in the benchmark suite are stereotyped, whereas the commands issued by the real-life operator are elliptical, deictic, fragmented, and context-dependent. The reported success rate does not transfer.

- ***Lack of uncertainty quantification*** makes it impossible to take a pass on the answer, as discussed in Sect. 2.1. This one point alone makes the request for information an assertion of an error, and is the most important and solvable of the errors made herein.

- ***Simulation-only validation*** does not work well in the case of highly tactile manipulation because those differences, which are crucial, like friction, deformable bodies, etc., are the hardest for simulators to model correctly.

- ***Treating the human as an obstacle***. Assuming that the human is a problem to be avoided, not an actor to be partnered with, results in safety and non-use. Cooperation necessitates the prediction and adaptation to a partner (Hoffman, 2019; Nikolaidis et al., 2017). A system that avoids humans cannot cooperate with them.

Table 2 consolidates this analysis by linking each cause to the population affected, its immediate and long-term effects, the solution used to address the problem, and why that solution is inadequate. It summarizes Sect. 2 and supplies the framework against which Sect. 8's five solutions are assessed.

Table 2.Problem analysis summary: causes, affected populations, effects, existing solutions, and failure points

Cause	Who is affected	Short-term effect	Long-term effect	Existing solution	Why it fails	Ref.

Symbol-grounding gap	Operators; integrators	Wrong object manipulated	Loss of calibrated trust	Open-vocabulary detection; queryable scene maps	Detector errors propagate unflagged into the plan	(Ahn et al., 2022; B. Chen et al., 2023; Harnad, 1990)
Open-world perception brittleness	Operators assisted users	Failure under clutter and occlusion	Restriction to structured cells	Larger vision backbones; scene graphs	Degrades without signaling degradation	(Gu et al., 2024; Kirillov et al., 2023; S. Liu et al., 2025)
No calibrated uncertainty	All	Confident errors; no abstention	Over- and under-trust	Verbalized confidence; log-probabilities	Systematically miscalibrated in LLMs and VLMs	(C. Guo et al., 2017; Kadavath et al., 2022; Xiong et al., 2024)
Token-to-continuum mismatch	Integrators; researchers	Infeasible grasps and motions	Reliance on hand-tuned skill APIs	Code-as-policies; action tokens	Feasibility is unchecked before actuation	(Brohan et al., 2023; M. J. Kim et al., 2024; Liang et al., 2023)
Unmodelled human intent	Operators	Obstructive motion; interruption	Reduced fluency; disuse	Reactive obstacle avoidance	Avoids humans rather than anticipating them	(Hoffman, 2019; Koppula & Saxena, 2016; Mainprice & Berenson, 2013)
No formal guarantees	Safety assessors	Unbounded worst case	Certification blocked	Reactive safety layer	Prevents injury only after	(Kress-Gazit et al., 2018;

					planning has failed	Zacharaki et al., 2020)
Interaction-data scarcity	Researchers	Poor generalization	Slow capability growth	Data pooling; simulation	Language prior overwhelms physical prior	(O'Neill et al., 2024; Zhao et al., 2020)

Table 2 illustrates the progression of each cause from its origin through its impact on the population, leading to the applied remedy and reasons for its failure. The fifth column summarizes current practices in terms of better detection, larger backbones, improved scene descriptions, cleaner code interfaces, and a reactive safety layer. These remedies primarily address the magnitude of errors rather than their detectability. Better detection reduces failure frequency without alerting users, while larger backbones encompass more scenarios without indicating when exceptions occur. The safety layer only prevents harm after a planner can no longer adjust. The exception is the row on calibrated uncertainty, which distinguishes itself in kind rather than degree and supports the argument presented in Sect. 8.6.

3. Background and Conceptual Foundations

3.1 Classical symbolic planning versus semantic planning

Classical planning is an extension of STRIPS (Fikes & Nilsson, 1971), in which the world is modeled in terms of sets of ground predicates, actions are represented by operators with preconditions and add and delete effects, and plans correspond to sequences that move the world from an initial configuration that meets certain goal conditions. This way of modeling problems was standardized using PDDL, along with its extensions, allowing temporal reasoning (Fox & Long, 2003) and made scalable with heuristic search planners (Helmert, 2006; Hoffmann & Nebel, 2001) The ability of task planners to incorporate motion planning takes into consideration the dependency of the feasibility of symbolic actions on geometrical reasoning (Garrett et al., 2021; Kaelbling & Lozano-Pérez, 2011; Srivastava et al., 2014).

The characteristics pertinent to this discussion are as follows. Classical planning is a closed-world process, meaning that everything which is not stated explicitly is false and, thus, that the planner understands the scope of his knowledge. Classical planning is sound and complete within the model of the world, making the solution obtained valid if the model itself is correct and the inability to obtain a solution meaningful. Inspection is also possible, as each action within the process is conditioned by specific preconditions that can be validated during execution. Finally, classical planning is brittle in the face of modeling error, since correctness is entirely contingent on a hand-authored model whose maintenance is the dominant cost because of its susceptibility to modeling errors.

With semantic planning, each property is reversed. Semantic planning is open-ended in nature, meaning that the referents for an instruction do not have to be known in advance, and that is exactly what gives it its worth. Semantic planning is neither sound nor complete; instead, it provides a plausibility ranking based on a language prior. Semantic planning can be inspected in that it produces an intermediate output; however, the trend towards action tokens is undermining that.

The tradeoff is thus not one of "better planning" but rather of shifting brittleness from modeling to referent binding. This formulation clarifies the motivation for the repeated occurrence of hybrid systems (Sect. 4.3): seeking open-world referent binding from the language model while preserving the safety guarantees of the symbolic planner.

3.2 LLMs in robotic reasoning and planning

Four capabilities are drawn on, each with a characteristic failure signature.

1. *Hierarchical task decomposition,* splitting up “get the workcell ready for the next variant” into a set of primitives, is the most reliable application of the model, in that it makes use of the experience with procedure-oriented text acquired by the model (Ahn et al., 2022; W. Huang, Abbeel, et al., 2022). A failure mode of this kind is realistic but has a non-executable granularity of the actions.

2. *Common sense priors* are those that provide the default assumptions not stated by an instruction that the container will be opened before being filled, and that the sharp object will be passed handle-first. The problem here is that these priors are cultural and general in nature rather than task-specific, and they silently override the procedure.

3. *Code generation* (Liang et al., 2023; Singh et al., 2022), This process produces a structure of code that can be inspected and composed. The failure pattern in code generation is API hallucination: function calls to non-existent functions or to the correct functions with invalid parameter ranges.

4. *In-context learning and chain-of-thought* (Brown et al., 2020; Wei et al., 2022; Yao et al., 2022) enable fast adaptation without fine-tuning. The signature of failure is that the reasoning path is a post hoc rationalization, not an explanation for why the output happened, and hence that its seeming consistency means nothing. The work from 2026 onwards tried to use these paths as something more than decoration to supervise the chain-of-thought in a dense way, so that reasoning is based on observed states rather than free-form generation (Sun et al., 2026). This is part of a solution to the problem of rationalization but doesn’t solve the problem of ensuring that the reasoning was correct. Critical assessments of language model-based planning (Valmeekam, Marquez, et al., 2023; Valmeekam, Olmo, et al., 2023), showed that performance on benchmark tasks of formal planning collapses when tasks stray beyond common templates, lending credence to the notion that these models are best as generators or heuristics within a verification system (Kambhampati et al., 2024).

3.3 VLMs in perception-grounded control and multimodal fusion

The vision-language models contribute in three areas. The first area is open vocabulary detection and segmentation (Kirillov et al., 2023; S. Liu et al., 2025; Minderer et al., 2022; Zhou et al., 2022). This function uses language models to resolve referring expressions to image regions without relying on a fixed set of classes. The second function is affordance and value grounding. It maps the language on an actionable structure, such as pixel-level pick and place affordances (Shridhar et al., 2022; Zeng et al., 2021), 3D value maps for motion synthesis (W. Huang et al., 2023), or even open-world object manipulation from web-scale priors (Stone et al., 2023). The third area is persistent scene representation, where language-aware features are embedded into a spatial map (Gu et al., 2024; C. Huang et al., 2023; Jatavallabhula et al., 2023; Rana et al., 2023; Shafiullah et al., 2023), so that queries can be answered against accumulated observation rather than the current frame.

Three constraints recur across all three. Spatial reasoning is weak precisely where manipulation depends on it, in relative depth, containment, support, and reachability. Detection confidence is poorly calibrated, so a confidently wrong answer is indistinguishable from a confidently correct one. And predominantly two-dimensional training distributions transfer poorly to metric three-dimensional space.

Where perception and language are fused determines what remains verifiable downstream, and the choice is made at one of four levels. Input-level fusion embeds visual features into the language model's token space (Driess et al., 2023; J. Li et al., 2023; H. Liu et al., 2023), giving strong joint reasoning and no inspectable perception output. Symbolic-level fusion has perception emit a discrete scene description that the planner consumes as text or predicates (B. Chen et al., 2023; Gu et al., 2024; Rana et al., 2023), which is the most auditable and discards the most information. Value-level fusion combines language scores with perceptual affordances (Ahn et al., 2022; W. Huang et al., 2023). Output-level fusion maps straight to action tokens (Brohan et al., 2023; M. J. Kim et al., 2024; Octo Model Team, 2024), maximizing end-to-end performance and eliminating inspection altogether. The failure is the same at every level: the modalities disagree, and no component owns the disagreement. The language model treats the brackets as implied by the instruction, the detector finds none, and with no arbitration mechanism, the architecture defaults to the language prior.

3.4 Formal definition of semantic planning in collaborative robots

Definition 1 (Collaborative semantic planning). Let $\mathcal{L}$ be the space of natural-language instructions, $\mathcal{O}$ the robot's observation space, $\mathcal{S}$ the latent world-state space, and $\mathcal{A} = \{a_1, \dots, a_k\}$ a finite library of parameterized robot skills with parameter spaces Θ_i. Let $\mathcal{H}$ denote the human partner state, comprising pose, predicted reach set over a horizon τ, and inferred task intent. A *grounding function* $g: \mathcal{L} \times \mathcal{O} \to \mathcal{S} \times \Theta$ maps an instruction and an observation onto a believed state and a parameterization. A collaborative semantic planner is a map

$$\Pi: \mathcal{L} \times \mathcal{O} \times \mathcal{H} \to (\mathcal{A} \times \Theta)^n$$

producing a plan $\pi = \langle (a_1, \theta_1), \dots, (a_n, \theta_n) \rangle$ that satisfies, for every i:

- **(P1) Groundedness.** Every referent in θ_i corresponds to an entity currently supported by $\mathcal{O}$, not inferred solely from the language prior; the property originates with the symbol-grounding problem (Harnad, 1990), and its restatement for contemporary language models (Bender & Koller, 2020); the robotics tradition of grounding referring expressions in perceived structure predates the foundation-model era (Matuszek et al., 2013; Tellex et al., 2011, 2020), and current practice discharges the condition through open-vocabulary detection and persistent scene representations (Gu et al., 2024; Jatavallabhula et al., 2023; Kirillov et al., 2023; Minderer et al., 2022). For a recent architecture that enforces it explicitly, see (F. Wu et al., 2026).
- **(P2) Feasibility.** $\mathrm{pre}(a_i)$ holds in the state predicted after $\langle a_1, \ldots, a_{i-1} \rangle$, and θ_i lies within the kinematic and dynamic reachable set from that state. The condition is inherited in two parts: the precondition-and-effect calculus from classical action representation (Fikes & Nilsson, 1971) and the geometric reachability requirement from task-and-motion planning, where symbolic and continuous feasibility must be settled jointly rather than in sequence (Garrett et al., 2021; Kaelbling & Lozano-Pérez, 2011; Srivastava et al., 2014). That preconditions are the operative failure surface in language-driven systems is confirmed by the fact that the most direct repair mechanisms in the recent literature are organized around precondition violation specifically (Raman et al., 2024).
- **(P3) Safety.** The trajectory realizing (a_i, θ_i) satisfies the separation-distance and force-speed constraints of the operative collaborative mode (ISO/TS 15066) with respect to the predicted human reach set over τ. The force and separation limit themselves to injury-biomechanics and collision-detection results (Haddadin et al., 2009; Haddadin & Croft, 2016), the surrounding assurance practice is surveyed in (Lasota et al., 2017; Zacharaki et al., 2020), and the predicted reach set is supplied by human motion prediction and activity anticipation rather than by the planner (Koppula & Saxena, 2016; Mainprice & Berenson, 2013). Note that the condition is stated over the predicted set, so its satisfaction inherits the uncertainty of that prediction, a dependence that current architectures leave implicit (D. Kim et al., 2026).
- **(P4) Legibility.** The partial plan observable at time t admits inference of the goal by the human partner with probability exceeding a threshold λ, in the sense of (Dragan et al., 2013). Legible motion is motion optimized for readability, which is a different objective from motion optimized for efficiency. The threshold is an empirical quantity about the partner rather than about the robot, so establishing it requires the interaction measurement apparatus discussed in Sect. 7.4 (Hoffman, 2019), and adaptation over repeated interaction can shift it (Nikolaidis et al., 2017).
- **(P5) Calibrated abstention.** The planner emits a confidence $c(\pi) \in [0,1]$ that is calibrated against realized success to within a stated tolerance, in the sense that over a reference episode distribution $P(\mathrm{success} \mid c(\pi) = v) = v$ for all v, up to an admissible calibration error in practice a bound on expected calibration error (C. Guo et al., 2017) or a finite-sample coverage guarantee at a stated level (Angelopoulos & Bates, 2022; Vovk et al., 2005), since exact calibration is not attainable on finite data. The planner abstains from acting, requesting clarification or transferring authority to the human,

whenever $c(\pi) < \gamma$. Two properties of γ deserve emphasis. It is set by operator tolerance rather than by statistical convention, because the cost asymmetry between a spurious query and an unsafe action is a property of the deployment and not of the estimator; and it interacts with trust, since a threshold low enough to generate frequent unnecessary queries produces the disuse that the automation literature documents as the dominant failure mode of imperfect aids (Parasuraman & Riley, 1997; Wickens & Dixon, 2007). Confidence in language-model planners can be obtained from token likelihood, verbalized self-report, or conformal calibration, with markedly different reliability (Kadavath et al., 2022; Lin et al., 2023; Ren et al., 2023; Xiong et al., 2024), and all three degrade under the distribution shift that open-world deployment guarantees (Ovadia et al., 2019).

Contemporary systems optimize the plausibility of π under the language prior while enforcing P1 partially, P2 inconsistently, P3 through an external reactive layer rather than within Π itself, and P4 and P5 not at all. The five conditions are not merely a checklist. P1-P3 constrain the per-step reliability of a plan, whereas P5 constrains the probability that a violation of P1-P3 is detected; these are distinct quantities, and the distinction is what makes the decomposition of Sect. 6.1 possible. It is also the distinction that the surrounding literature most often collapses: proposals to improve grounding, feasibility checking and safety are proposals about the first quantity, whereas monitoring, repair and help-seeking are proposals about the second (Y. Guo et al., 2024; W. Huang, Xia, et al., 2022; B. Liu et al., 2023; Raman et al., 2024; Ren et al., 2023), and the two are routinely compared as though they were commensurable. Proposition 1 in Sect. 6.1 gives the condition under which the second dominates the first. Definition 1 is therefore used in two ways in what follows: as the axis along which architectures (Sect. 4, Table 4) and solutions (Sect. 8, Table 8) are assessed, and as the source of the reliability decomposition on which the paper's central quantitative argument rests.

With these foundations in place, we turn to how the field has organized itself architecturally.

Table 3. *Notation used in Definition 1 and in the reliability model of Sect 6.1*

Symbol	Meaning
L, O, S	instruction, observation, and latent world-state spaces
A, Theta	parameterized skill library and its parameter spaces
H	human partner state: pose, predicted reach set over horizon tau, inferred intent
g	grounding function mapping instruction and observation to state and parameterization
Pi, pi	planner and the plan it returns
c(pi), gamma	plan confidence and the abstention threshold
lambda	legibility threshold on goal inference by the partner
p, p'	per-step success probability of two planners under comparison
r, rho	detection, recall, and recovery success probability
f, k	false-positive rate and mean recovery cost in steps

n	plan length in steps

4. Taxonomy and Architectural Overview

These systems are typically categorized by the domains of their applications, thereby overlooking the fact that architecturally different systems share common failure points, whereas those with similar architectures may not. In this regard, we categorize the field by where semantics enter the control chain, because that factor determines what can be validated and what must be trusted. **Fig. 2** presents the emerging taxonomy.

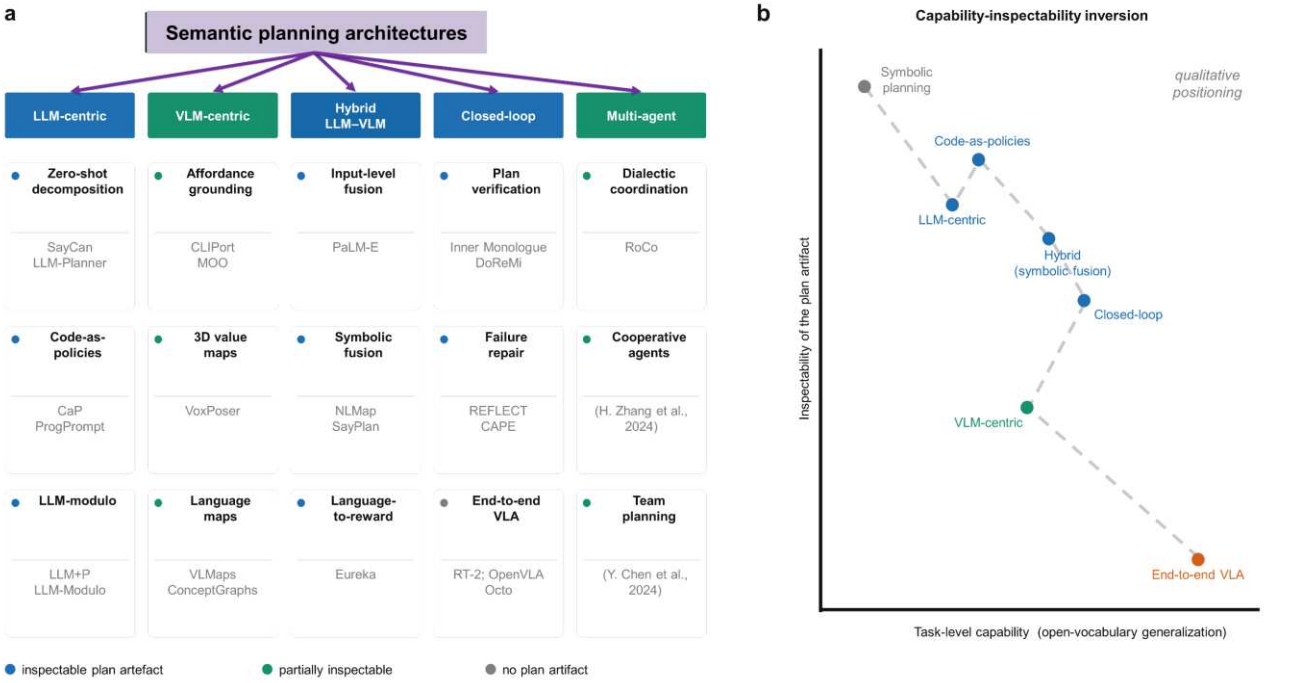


Figure 2. *Taxonomy of semantic planning architectures for collaborative robots that are LLM-centric, VLM-centric, or hybrid.* ***a****, Five architecture classes, each with three sub-approaches and representative systems. Marker color encodes whether the sub-approach emits an inspectable intermediate plan artifact (blue), is partially inspectable (green), or emits none (grey); the grey node marks end-to-end vision-language-action models.* ***b****, Qualitative positioning of the same classes on task-level capability against inspectability of the plan artifact. The monotone decline is the paper's central architectural observation: the systems that generalize best are those on which correctness can be checked least. Positions are ordinal and derived from Sect. 4, not measured*

4.1 LLM-centric architectures

In this class, task decomposition and task sequence planning are performed by a language model, while the perception system is used only to rate the candidate actions suggested by the model. Examples of such systems include the zero-shot planners (W. Huang, Abbeel, et al., 2022) SayCan (Ahn et al., 2022), LLM-Planner (Song

et al., 2023), TidyBot (J. Wu et al., 2023), and the neuro-symbolic LLM-modulo approaches arrangements of (Guan et al., 2023; Kambhampati et al., 2024; B. Liu et al., 2023; Silver et al., 2024).

The defining property of such systems is compositional generalization over the task space, since the ability to decompose does not depend on an enumerated domain but on a language prior and thus works even when faced with new ways of phrasing instructions and new sequences of tasks. The defining weakness is that the system queries a language before obtaining information it cannot provide. It refers to objects that have not been established through perception, and it orders steps according to how such tasks are typically performed, rather than how they are performed in this particular workcell. In the terminology of Definition 1, this category solves none of P1: groundedness is outsourced to a downstream filter.

4.2 VLM-centric architectures

Here, perception-based computations are at work, and language serves to condition or question these computations. CLIPort (Shridhar et al., 2022) and its prior versions (Zeng et al., 2021) map instructions to affordances for pick-and-place tasks, whereas MOO (Stone et al., 2023) uses manipulation over objects belonging to open vocabularies; VoxPoser (W. Huang et al., 2023), generates 3D value maps that are used for motion optimization, while VLMaps (C. Huang et al., 2023), CLIP-Fields (Shafiullah et al., 2023), ConceptFusion (Jatavallabhula et al., 2023), and ConceptGraphs (Gu et al., 2024) build persistent language-aligned spatial representations.

The advantage is grounding fidelity: the results are linked to structures in the world by construction, which solves the failure of Sect. 4.1 in that the system makes references to things that don’t exist. The typical disadvantage is horizon (Zeng et al., 2021). These approaches are great at one- or two-step-ahead instructions, but they lack the symbolic tools to reason about an action whose precondition was set up four steps earlier.

4.3 Hybrid LLM-VLM architectures

Hybrid systems aim to integrate decomposition with grounding. In PaLM-E (Driess et al., 2023) and instruction-tuned multimodal models (J. Li et al., 2023; H. Liu et al., 2023), the fusion happens at the input stage. NLMap (B. Chen et al., 2023) and SayPlan (Rana et al., 2023) perform symbolic fusion by generating a queryable scene representation from the perceptual system that the language model can plan on. Compositional systems (Socratic Models) (Zeng et al., 2022) combine independent systems using language as the interlingua. The language-to-reward approach (Ma et al., 2024; W. Yu et al., 2023) involves using the language model to define the optimizer's objective.

The advantage of the approach is its ability to handle both P1 and P2 within a single framework, while the hybrid model delivers the best performance on open-vocabulary and long-horizon problems. The typical problem of the model is the modality disagreement discussed in Sect. 3.3: When there is a conflict between scene abstractions

and language priors, the lack of an arbitration mechanism will make the architecture default to language priors. The symbolic fusion model is the most auditable method and is our default recommendation.

4.4 Closed-loop perception-action systems

This class conditions the process on the observed outcome, which is central to this class of approaches. The Inner Monologue (W. Huang, Xia, et al., 2022) approach integrates success, failure detection, scene descriptions, and human feedback back into the planning prompt. The REFLECT (Z. Liu et al., 2023) approach summarizes execution traces to diagnose and fix failures. DoReMi (Y. Guo et al., 2024) checks for the mismatch between expected and observed effects of the plan. CAPE (Raman et al., 2024) derives corrections for the action plan from precondition violations. The RePLan (Skreta et al., 2024) integrates perception monitoring and planning into a single framework, and integrates such architectures with learned world models to predict the effect of an action beforehand (F. Wang et al., 2026). This represents the closest thing to an internal validation approach within the end-to-end paradigm. The end-to-end vision-language-action models (Brohan et al., 2022, 2023; M. J. Kim et al., 2024; O'Neill et al., 2024; Octo Model Team, 2024) are closed-loop at the control level, in that each action is conditioned on the current observation.

The strength of this class is that it is the only structural remedy for silent failure: a process that can verify the accuracy of its operations will spot the problem described in Sect. 2.3. The critical issue is that the verification process is a calibration-free model by nature. A VLM which asks "did the grasp succeed?" is just as prone to overconfidence as the model which provided the referent; moreover, the false negative causes unnecessary replanning, while the false positive causes the very silent failure that the loop should prevent.

4.5 Multi-robot and multi-agent collaborative planning

Language-based multi-agent systems employ language to coordinate among robots or reasoning modules. RoCo (Mandi et al., 2024) involves several robotic arms in negotiating a common strategy in a dialectical manner; modular cooperative embodied agents (H. Zhang et al., 2024) and scalable multi-robot setups (Y. Chen et al., 2024) both generalize RoCo. The key advantage is that coordination via dialogue is observable in a way that is not the case with policy optimization. The most recent survey of LLMs for multi-robot systems organizes applications along high-level task allocation, mid-level motion planning, low-level action generation, and human intervention, and identifies mathematical reasoning, hallucination, latency, and the absence of benchmarking as the principal barriers to adoption (P. Li et al., 2026). These barriers correspond closely to the causes cataloged in Sect. 2.1. The distinctive problem of this class, however, is cumulative: because agents are independently calibrated, the likelihood that at least one hallucinated commitment makes its way into the collaborative plan increases with team size, and negotiation adds unwarranted confidence to the outcome. Latency increases with the number of communications, too, contrary to Sect. 9.2.

4.6 Comparative assessment

Assessed on consistent axes, the classes trade off against one another rather than dominating. Grounding fidelity rises from LLM-centric through hybrid to VLM-centric while horizon and compositional reasoning fall in the same order; inspectability peaks for symbolic hybrids and code generation and bottoms out for end-to-end action models; human-awareness is treated as a first-class concern by essentially none of them; and run-time cost is highest for closed-loop verification, which is the capability the argument of this review most requires. Table 4 gives the comparison per class and Table 5 per system.

Table 4. Comparison of LLM-centric, VLM-centric, and hybrid approaches across consistent architectural axes

Axis	**LLM-centric**	**VLM-centric**	**Hybrid LLM–VLM**	**Closed-loop**	**Multi-agent**	**Ref.**
Primary input	Text	Image + text	Text, image, scene graph	Text, image, execution feedback	Text between agents	(Ahn et al., 2022; W. Huang, Xia, et al., 2022; Rana et al., 2023; Shridhar et al., 2022); (Mandi et al., 2024)
Reasoning horizon	Long	Short	Long	Medium–long	Long	(W. Huang, Abbeel, et al., 2022; W. Huang et al., 2023; Rana et al., 2023)
Grounding fidelity (P1)	Low	High	Medium–high	Medium	Low	(Ahn et al., 2022; Gu et al., 2024; Stone et al., 2023)
Feasibility checking (P2)	Weak	Partial	Partial	Explicit	Weak	(W. Huang et al., 2023; B. Liu et al., 2023; Raman et al., 2024)
Inspectable plan artifact	Yes	Partial	Yes (symbolic fusion)	Yes	Yes	(Z. Liu et al., 2023; Rana et al., 2023; Singh et al., 2022)
Human-awareness	Absent	Absent	Rare	Rare	Absent	(Dragan et al., 2013; Hoffman, 2019)

(P3-P4)						
Uncertainty handling (P5)	Rare	Absent	Rare	Rare	Absent	(Ren et al., 2023; Xiong et al., 2024)
Compute cost at run time	Medium	Low–medium	High	High	Very high	(Firoozi et al., 2023; Leviathan et al., 2023)
Typical failure mode	Ungrounded referent	Horizon failure	Unadjudicated modality conflict	Uncalibrated verifier	Compounded hallucination	(Z. Liu et al., 2023; Mandi et al., 2024; Valmeekam, Marquez, et al., 2023)

Table 4 illustrates that the five architecture types are arranged along a single set of axes; three rows are noteworthy because they mark where the different architecture types diverge. The rows related to grounding-fidelity and reasoning-horizon differ in that VLM-centric architectures link their output to observed structure and thus never refer to non-existent objects, but lack the symbolic machinery to reason about a condition four actions ago, whereas LLM-centric architectures are the reverse. Hybrid architectures lie between the two mentioned above, which is why they now yield the best performance on long-horizon open-vocabulary tasks. The inspectable-plan-artifact row is the one that has the practical engineering impact, as the class of agents producing no artifact closes off the verification described in Sect. 8.1, no matter what its capability is. The last two rows are the ones that should surprise the reader: human awareness is lacking across all classes, and uncertainty handling is rare in all classes except the system explicitly designed for this purpose. These are not weaknesses of a specific architecture; these are weaknesses of the field as a whole.

Table 5. Representative methods, capabilities, and limitations

System	**Yr**	**Class**	**Platform**	**Domain**	**Key capability**	**Principal limitation**	**Eval. setting**	**Ref.**
Zero-shot planners	2022	LLM	Sim	Household	Decomposition without task data	No grounding to scene	Sim	(W. Huang, Abbeel, et al., 2022)
SayCan	2022	LLM + affordance	Mobile manip.	Kitchen	Affordance-weighted step scoring	Filter can reject, not revise	Real	(Ahn et al., 2022)

Inner Monologue	2022	Closed-loop	Manipulator	Tabletop	Feedback re-entering the prompt	Uncalibrated success detector	Real	(W. Huang, Xia, et al., 2022)
Code as Policies	2023	LLM code	Manipulator	Tabletop	Composable, inspectable programs	API hallucination	Real	(Liang et al., 2023)
ProgPrompt	2023	LLM code	Manipulator	Household	Situated program synthesis	Fixed primitive vocabulary	Sim + real	(Singh et al., 2022)
PaLM-E	2023	Hybrid	Mobile manip.	Multi-domain	Input-level multimodal fusion	No separable perception audit	Real	(Driess et al., 2023)
VoxPoser	2023	VLM	Manipulator	Tabletop	3D value maps for motion synthesis	Needs dense geometry; short horizon	Real	(W. Huang et al., 2023)
SayPlan	2023	Hybrid	Mobile robot	Building-scale	3D scene-graph grounded planning	The scene graph must be maintained	Real	(Rana et al., 2023)
KnowNo	2023	LLM + CP	Manipulator	Tabletop	Conformal abstention and help-seeking	Coverage under distribution shift	Real	(Ren et al., 2023)
REFLECT	2023	Closed-loop	Manipulator	Household	Failure explanation from traces	Post-hoc; latency	Sim + real	(Z. Liu et al., 2023)
RT-2	2023	VLA	Manipulator	Tabletop	Web knowledge transferred to control	No inspectable plan	Real	(Brohan et al., 2023)

OpenVLA	2024	VLA	Manipulator	Multi-domain	Open generalist policy	No inspectable plan; compute	Real	(M. J. Kim et al., 2024)
RoCo	2024	Multi-agent	Dual-arm	Assembly	Dialectic multi-robot coordination	Compounded hallucination; latency	Sim + real	(Mandi et al., 2024)
DoReMi	2024	Closed-loop	Manipulator	Tabletop	Plan–execution misalignment detection	Constraint set authored manually	Real	(Y. Guo et al., 2024)
CAPE	2024	Closed-loop	Manipulator	Household	Correction from precondition errors	Requires precondition model	Sim + real	(Raman et al., 2024)
Concept Graphs	2024	VLM map	Mobile manip.	Indoor	Open-vocabulary 3D scene graphs	Build cost; staleness	Real	(Gu et al., 2024)
TacVLA	2026	VLA + tactile	Manipulator	Contact-rich	Tactile fusion into the policy	Sensor coverage; calibration	Real	(K. Zhang et al., 2026)
Fast-ThinkAct	2026	VLA	Manipulator	Tabletop	Verbalizable latent planning at low latency	Compressed trace is harder to audit	Real	(C.-P. Huang et al., 2026)
Xiaomi-Robotics-0	2026	VLA	Manipulator	Multi-domain	Open policy with real-time execution	No inspectable plan artifact	Real	(Cai, Guo, et al., 2026)

Legible shared autonomy	2026	Shared control	Manipulator	Assistive	Belief communicated through motion	Legibility-efficiency trade-off	Real	(J. Liu et al., 2026)
Semantic anomaly detection	2023	LLM monitor	Vision policy	Driving/ manip.	System-level edge-case recognition	The monitor itself is uncalibrated	Real	(Elhafsi et al., 2023)

Table 5 focuses on individual systems, highlighting that real-hardware evaluation has become standard and that the quality of negative results has improved since 2022. It outlines a transition from LLM-centric approaches to vision-language-action policy design in 2024 systems, which are bimodal in inspectability. The table identifies four principal limitations: a non-revisable filter, a horizon that cannot reach earlier conditions, an uncalibrated verifier, and the absence of any plan artifact. Only one entry, KnowNo, addresses calibrated abstention, but it has its own limitation related to coverage degradation under distribution shift, which must be addressed in Sect. 8.2.

Having characterized the architectures, we now examine the planning functions they must perform.

5. Semantic Planning for Collaborative Robots

Fig. 3 shows the reference pipeline to which the functions presented in this section are compared. The figure is placed at the beginning of Sect. 5 because each subsection corresponds to a step in the pipeline.

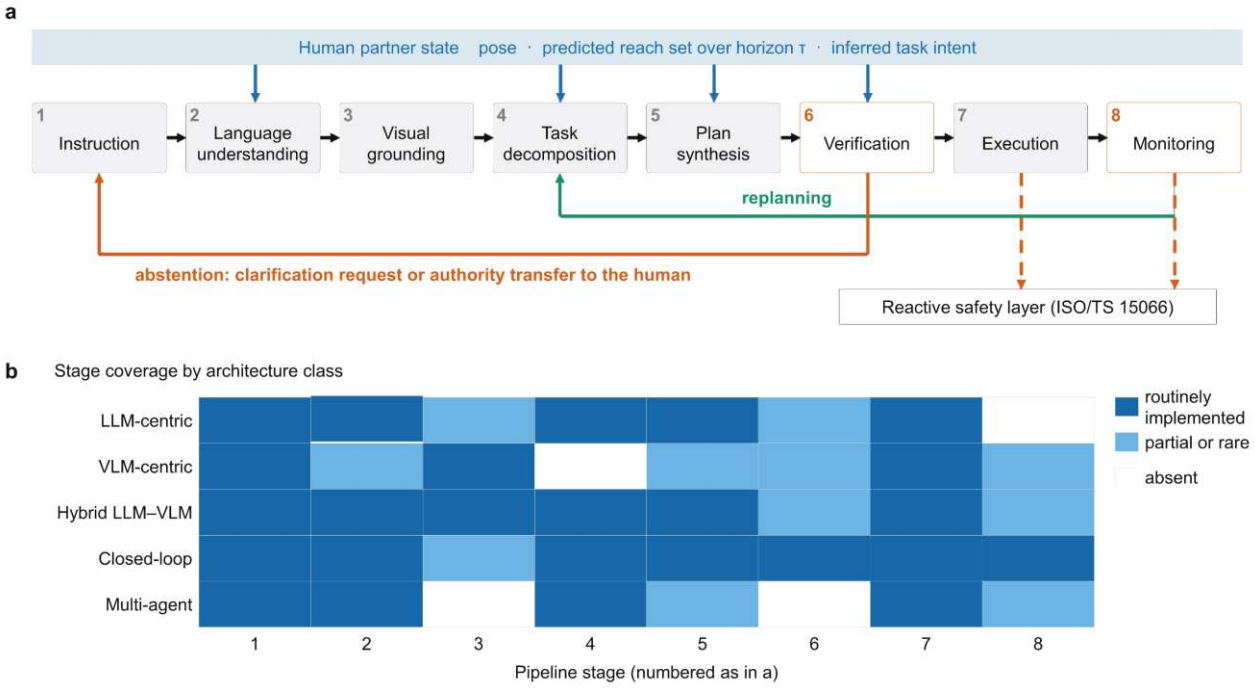


Figure 3. *Semantic planning pipeline and its coverage by the architecture class.* **a***, Eight-stage pipeline from natural-language instruction to monitoring. Stages 6 and 8 (orange) are the verification and monitoring stages, absent or weak in most surveyed systems; their omission removes the entire error-detection path. The dashed band is the human partner state, entering only at stages 3, 5, and 6. Green return path, replanning; orange return path, abstention from the human on low confidence. The band beneath stages 7 and 8 is the reactive safety layer, which protects the operator from a bad plan but never inspects the plan.* **b,** *Coverage of each pipeline stage by the five architecture classes of Sect. 4, coded from the assessment corpus of Table 4. Stages 6 and 8 are the only columns not routinely implemented by any class*

Three features of Fig. 3 carry the argument. The two highlighted stages, verification at 6 and monitoring at 8, are the ones most often absent from surveyed systems, and the only two whose function is to establish whether the preceding stages were performed correctly; omitting them removes the entire error-detection path, which is the structural form of the compounding argument of Sect. 6.1. The two return paths are this review's highest-leverage proposals: replanning, from monitoring back to decomposition and plan synthesis (Sect. 6), and abstention, from verification back to the instruction (Sect. 8.2). Abstention terminates at the human rather than inside the robot, which is the point: asking is a planning action. The reactive safety layer is drawn beneath stages 7 and 8 rather than within the pipeline, because it shields the operator from a bad plan without ever inspecting it (Sect. 7.1). The human-state band enters at only three points, and Sect. 5.3 finds even that integration is almost universally absent.

5.1 Task decomposition and hierarchical reasoning

Decomposition transforms an underspecified objective into an ordered sequence of executable components, and it is the function which foundation models excel at the most (Ahn et al., 2022; W. Huang, Abbeel, et al., 2022; Song et al., 2023). There are three key design choices that make decomposition useful. First, granularity alignment: the decomposition should end in the components corresponding to the entries in the robot's library of primitives, and granularity mismatch is the most frequent reason why the produced plans look correct but cannot be executed. This problem is automatically avoided by systems that generate plans, provided the primitive actions are explicitly listed (Liang et al., 2023; Singh et al., 2022). The second choice is linear order vs. dependency: a linear order imposes too many constraints on a task when its components are only partially dependent, and the ability to choose an order is essential in collaborative environments to accommodate human action without giving up the plan. Hierarchies, or dependency graphs, are much more useful representations in this case than the plain lists most systems generate. The third choice is to update the plan based on new information, as discussed in Sect. 6.

The key problem is that it is impossible to validate the quality of decomposition based on the decomposition process alone. Fluency and correctness do not correlate, and experiments on formal planning domains (Valmeekam, Marquez, et al., 2023; Valmeekam, Olmo, et al., 2023), confirm poor performance on anything beyond well-known templates. This makes it reasonable to regard the model as a distribution over plans, which

should then be filtered through some sound process, instead of a planner, as proposed in (Kambhampati et al., 2024) and adopted in Sect. 8.1.

5.2 Grounding language into executable actions

Grounding relies on four approaches, each distinct through the artifact it generates. ***Skill libraries with affordance filtering*** (Ahn et al., 2022) scores every potential primitive based on the product of linguistic and physical plausibility, and this filter works precisely as well as the affordance function does; it cannot prevent a poor move, but can catch one. ***Code generation*** (Liang et al., 2023; Singh et al., 2022), outputs a code base, whose method calls can be verified against the actual API, something underutilized, as signature and parameter-range checking provides cheap protection from hallucinating an incorrect API at no expense. ***Value and affordance maps*** (W. Huang et al., 2023; Shridhar et al., 2022; Zeng et al., 2021), ground into a detailed geometrical structure ready for movement planning at the expense of reliable geometry. ***Action tokenization*** (Brohan et al., 2023; M. J. Kim et al., 2024; Octo Model Team, 2024) skips this intermediate step and achieves the strongest reported generalization at the expense of eliminating the artifact on which P1 and P2 would be verified.

Referencing expression resolution needs to be considered separately, as this is the point at which the collaboration departs the most from the benchmark. Operators rely on deixis ("that one"), ellipsis ("and now the other side"), and common knowledge ("just like before"), none of which exist in benchmark instructions, and all of which necessitate dialogue state and a long-term scene memory (Gu et al., 2024; Rana et al., 2023) rather than single-frame perception.

5.3 Human-aware planning in shared spaces

The planner for a collaborative system needs to reason about humans as agents. Four criteria ensue. The ***Proxemic and separation constraints*** criteria arise from the operative collaborative modality defined in ISO/TS 15066 and from the social distancing literature (Sisbot et al., 2007). They are dynamic in nature, since the allowable speed depends on the current separation distance; therefore, there is coupling rather than layering in the planning and safety process. ***Intent inference and motion prediction*** (Koppula & Saxena, 2016; Mainprice & Berenson, 2013), enable the robot to plan around the position of the human in the near future (τ) rather than their current position, and this is the distinction between a robot that is simply safe and one that is non-obstructive. ***Legibility*** (Dragan et al., 2013) requires that the trajectory performed allows early goal prediction, and conflicts with efficiency, as the most legible trajectory is not usually the shortest. ***Fluency*** (Hoffman, 2019) encompasses the timing of the joint task and is the best predictor of whether the operator experiences the system as a collaborator.

Some early counterexamples exist and should be mentioned. A deployable social robot navigation system explicitly combines semantic reasoning about vision-language relationships with low-level control reasoning regarding surrounding humans (Cai, Yamasaki, et al., 2026). This proves that the integration is feasible, rather

than merely desirable. However, the difference is evident. No existing system of LLMs or VLMs for planning incorporates human motion models or aims to optimize legibility; this is the biggest difference between the literature on semantic planning and that on human-robot collaboration.

5.4 Shared autonomy and control handover

Shared autonomy is a combination of human and autonomous contributions as opposed to an alternation between them (Dragan & Srinivasa, 2013; Javdani et al., 2018; Losey et al., 2018; Selvaggio et al., 2021). Semantic planning introduces one key distinction in the framing of the issue: arbitration can take place in a semantic fashion as well, not just a kinematic one. Rather than combining velocity commands, the task of arbitration can be transferred at a finer level of resolution subtask, such as the robot doing what it is sure about and passing on the delicate insertion to the human operator.

The legible shared autonomy provides the belief of the robot implicitly by its action, so that the person can figure out what the system believes now, without being informed afterward (J. Liu et al., 2026). It is the first approach in our review to connect the legibility literature (Sect. 5.3) with the arbitration problem. The transparency of confidence does not necessarily require additional explicit interaction. There are three conditions that define whether a handover will be performed correctly. First, it should be clear to both the person and the robot who is in charge. Ambiguity about control is a known cause of automation errors (J. D. Lee & See, 2004). Second, the handover has to be predictable and based on some explicit threshold, not hidden inside the system. Third, it should be confidence-driven, which is related directly to P5 and the proposed solution in Sect. 8.2: a calibrated confidence estimate should serve as the trigger.

5.5 Personalization and adaptive behavior

The speed, handedness, tolerance to robot closeness, and division of work among the operators vary, and a system that can accommodate these differences has been found to be more fluid (Nikolaidis et al., 2017; J. Wu et al., 2023). The adaptation process could be explicit, using preferences, or inferred based on the history of interaction; the latter can be done at reduced cost with language models because history can be recorded using context in natural language (J. Wu et al., 2023).

There are three such dangers that are underappreciated in the literature. ***Preference drift*:** a preference inferred from its consequences will tend to drive the system away from safety or efficiency, even if each decision is correct in itself. ***Privacy*:** interaction history in a workplace constitutes information about the behavior of an identifiable person, and is therefore protected personal data. ***Over-adaptation*:** a robot optimized to work for one employee will be unpredictable when controlled by the next shift's employee, turning efficiency into a safety risk. These risks motivate the bounded, inspectable memory design proposed in Sect. 8.4.

We now turn to the mechanism that determines whether any of this survives contact with a real environment.

6. Closed-Loop Replanning, Failure Detection, and Recovery

6.1 Why open-loop plan generation is insufficient

An open-loop planner relies on the premise that performing action a_i yields the state its model predicts. That premise is exactly the one classical action representation makes explicit and discharges through preconditions and effects (Fikes & Nilsson, 1971), and every source of error cataloged in Sect. 2.1 contravenes it. The consequence is not specific to language models. Wherever a policy is executed open-loop over a horizon, small per-step deviations compound because each one moves the system into states the model characterizes less well, a result established for sequential decision making well before the current architectures existed (Ross et al., 2011). What foundation-model planners add is not a new failure mechanism but the removal of the interface through which the mechanism was previously observable. To make the consequence precise, and to expose the assumptions on which it rests, we state the reliability model explicitly rather than appealing to it informally.

Let each of the n steps of a plan succeed with probability p, and assume for the moment that step outcomes are independent. The plan then succeeds with probability p^n, so $p = 0.95$ yields 0.60 over sixteen steps and 0.36 over twenty, horizons that collaborative assembly routinely exceeds. Fig. 4a plots this collapse. Two qualifications must be attached to the independent model, because it is the assumption most open to challenge. First, manipulation failures are in practice, positively dependent: an ungrounded referent contaminates every subsequent step that refers to the same entity, and adverse illumination degrades all perception steps together. For positively dependent Bernoulli outcomes $P(\bigcap_i S_i) \geq \prod_i P(S_i)$, so p^n is a conservative lower bound on plan success rather than an estimate of it. Second, and pulling in the opposite direction, dependence concentrates failure early in the sequence, because the steps that establish referents precede the steps that consume them. The same asymmetry makes early intervention disproportionately valuable in sequential policies generally (Ross et al., 2011). The independent model therefore understates total success while also understating the value of *early* detection, and both distortions should be borne in mind when reading the figures below. The value $p = 0.95$ is used illustratively; per-step reliabilities are seldom reported separately from task success in the surveyed literature, with the exception of systems that explicitly decompose plan generation from plan execution (Ahn et al., 2022), and this omission is itself one of the reporting gaps identified in Sect. 7.4.

Now admit detection and recovery. Let a failed step be detected with probability r, the detection recall of the monitoring system, and let a detected failure be recovered within the attempt budget with probability ρ. An undetected failure propagates and the plan fails. The two quantities are separately instantiated in the closed-loop literature: detection by success verification, precondition monitoring, and plan-execution misalignment checks (Y. Guo et al., 2024; W. Huang, Xia, et al., 2022; Khalastchi & Kalech, 2019), and recovery by corrective-action synthesis and explanatory replanning (Z. Liu et al., 2023; Raman et al., 2024; Skreta et al., 2024). Per-step effective reliability becomes

$$p_{\text{eff}} = p + (1 - p)\, r\rho,$$

and plan success becomes p_{eff}^{n}. This model is what licenses the paper's central claim, and it also bounds it, which the informal version did not.

Proposition 1 (detection dominates accuracy above a threshold, under independence). *Assume steps fail independently, as in the model above. Consider a planner A satisfying P1-P3 of Definition 1 with per-step reliability p but violating P5, so that $r = 0$; and a planner B with lower reliability $p' < p$ that satisfies P5 with detection recall r and recovery success ρ. Then B attains higher plan success than A at every horizon $n \geq 1$ if and only if*

$$r\rho \;>\; \frac{p - p'}{1 - p'}.$$

Proof. $p_{\text{eff}}^{n} > p^{n}$ for $n \geq 1$ if $p_{\text{eff}} > p$, since $x \mapsto x^{n}$ is strictly increasing on $[0,1]$. Substituting $p_{\text{eff}} = p' + (1 - p')r\rho$ and rearranging gives $(1 - p')r\rho > p - p'$, and dividing by $1 - p' > 0$ gives the result.

Three consequences follow, and they are more informative than the qualitative claim they replace. First, within the independent-step model the threshold is *independent of horizon*: whether the plan has five steps or fifty, the detection quality required to offset a given accuracy deficit is the same, so the trade-off can in principle be evaluated once for a platform rather than per task. This invariance is a property of the geometric form and should not be carried outside it; Sect. 6.1.1 states what survives when the independence assumption is relaxed. Second, the required detection quality is $\Delta/(1 - p')$ where $\Delta = p - p'$ is the accuracy deficit, which is punishing at high reliability. Offsetting a seven-point deficit from $p' = 0.90$ against $p = 0.97$ demands $r\rho > 0.70$; at a plausible recovery success of $\rho = 0.8$ this requires detection recall $r > 0.875$. Fig. 4b shows this crossover and Fig. 4c the full iso-performance surface. Third, and consequently, the widely repeated intuition that detection simply matters more than accuracy is *false in general and true only above the threshold*. A monitoring system with mediocre recall can be worse than no monitoring at all once its false-positive cost is counted. This is not a novel observation about robots so much as a rediscovery: the human-factors literature on imperfect diagnostic aids reached the same structural conclusion from experimental data, and a meta-analysis across twenty studies located the crossover below which an unreliable aid is worse than no aid at approximately 0.70 reliability (Wickens & Dixon, 2007). That the same figure appears in our worked example is coincidental in magnitude and instructive in form; the quantity being thresholded is different, but the existence of a threshold, and the consequence of operating beneath it, are the same. The honest form of the field's guidance is therefore conditional: invest in detection when, and only when, the achievable $r\rho$ exceeds $\Delta/(1 - p')$ for the accuracy you would otherwise have bought.

False positives do not enter Proposition 1 because a spurious recovery from a successful step does not corrupt the plan, but they dominate the throughput cost. With false-positive rate f and mean recovery cost $\bar{k}$ steps, expected effort per nominal step is $1 + [(1 - p)r + pf]\bar{k}$, which grows linearly in f and is multiplied by p rather

than $(1-p)$ so at high reliability the monitoring overhead is driven almost entirely by false positives on steps that never failed. The cost is not only throughput. False alarms and misses degrade operator behavior along separate channels: false alarms erode compliance with the aid, while misses erode reliance on it (Dixon et al., 2007), so a monitor tuned for recall alone can produce a system that is statistically safer yet operationally abandoned (Parasuraman & Riley, 1997). This is why Sect. 7.4 treats the false-positive rate as the reporting omission with the largest practical consequence.

6.1.1 Scope and limitations of the reliability model

Four assumptions bound Proposition 1. We state them because the proposition is this review's central quantitative claim and because two of them are in tension with arguments made elsewhere in this review.

First, independence and recovery. The derivation treats step outcomes as independent, whereas the preceding paragraph argues that manipulation failures are positively dependent. For plan success the independent model is conservative, since positive dependence raises the probability that all steps succeed; for the threshold no such guarantee holds, because once dependence is admitted plan success is no longer geometric over the horizon and the invariance of the crossover is not preserved. Under a contamination model in which a grounding error at step i corrupts every subsequent step that refers to the same entity, the sequential analog of the compounding-under-shift mechanism analyzed by (Ross et al., 2011), early detection is strictly worth more than late detection, so Proposition 1 understates the value of detection while losing its horizon-independence. The threshold is therefore exact under independence, a useful approximation under weak dependence, and requires horizon-specific evaluation under strong dependence; deriving the contamination-model condition is left open rather than assumed to be covered. The recovery term is treated similarly optimistically: recovery success is modeled as a constant, independent of per-step reliability and of failure type, with a recovered step returning execution to nominal at no residual cost. Sect. 6.5 argues the opposite for the case that matters most, and the argument is now supported from two directions. A recovery generated by the model that produced the failed plan preserves the original misconception, because language models do not reliably correct their own reasoning without an external error signal (J. Huang et al., 2024), and self-critique of plans in particular has been shown not to improve them (Valmeekam, Marquez, et al., 2023), which is precisely the argument for placing the verifier outside the generator (Kambhampati et al., 2024). Recovery success is therefore correlated with the failure cause and lowest exactly where it is most needed: the recovery mechanisms that do work in practice are those driven by an external signal, whether a precondition check, a physical observation, or a human report (Y. Guo et al., 2024; Z. Liu et al., 2023; Raman et al., 2024). Conditioning ρ on failure type is the correct refinement, and it raises the detection quality required to clear the threshold.

Second, false positives and measurability. Proposition 1 governs only the plan's success probability. A spurious recovery from a successful step does not corrupt the plan, so false positives are absent from the criterion, yet they dominate throughput cost at high reliability, as the expression above shows, and they act on operator compliance through a channel the success criterion does not represent at all (Dixon et al., 2007; Wickens & Dixon, 2007).

The rule stated in the third consequence is therefore necessary but not sufficient for detection to be the better investment, and the success criterion and the throughput expression must be evaluated jointly. The criterion is also expressed in quantities the surveyed literature almost never reports separately: per-step reliability, detection recall, recovery success, false-positive rate, and mean recovery cost. We were consequently unable to instantiate it for any system in Table 5, and we record that as a limitation rather than present the proposition as an applied result. These quantities are not hard to measure, merely unreported, the same diagnosis that motivated the construction of dedicated planning benchmarks when aggregate scores proved uninformative about planning competence (Valmeekam, Olmo, et al., 2023), and that the 2026 literature restates from the data side in arguing that evaluation infrastructure now constrains progress more than architecture does (Z. Wang et al., 2026). It is the gap identified in Sect. 7.4, and the reason reporting reform is proposed in Sect. 8.5 rather than treated as secondary. Until they are reported, Proposition 1 is a criterion awaiting data, not a verdict on any existing system.

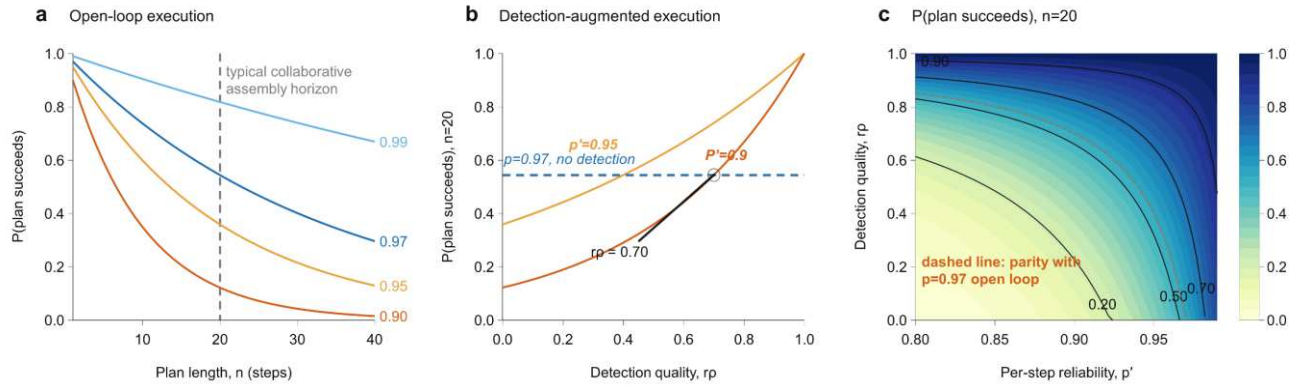


Figure 4. *Reliability of plan execution under open-loop and detection-augmented models.* ***a****, Probability that an n-step plan succeeds under independent per-step failure, for four per-step reliabilities p; the dashed line marks the twenty-step horizon typical of collaborative assembly.* ***b****, Plan success at n=20 as a function of detection quality rp for two per-step reliabilities p', against the open-loop baseline at p=0.97 (dashed); the marked point is the crossover threshold rp=0.70 given by Proposition 1.* ***c****, Iso-performance surface over per-step reliability and detection quality at n=20; solid contours are plan success, and the dashed contour traces parity with the p=0.97 open-loop baseline. Points above and to the left of the dashed contour are regimes in which investing in detection outperforms investing in accuracy*

Provisional corroboration consistent with the model appeared in 2026, when long-horizon manipulation failures were reported to persist after finetuning on large teleoperated datasets (Choe et al., 2026). That is the direction the model predicts, since additional data raises p but leaves the exponent untouched. The source is a preprint and the reported failures are not decomposed into the per-step quantities the model requires, so the observation is consistent with Proposition 1 rather than a test of it. On the model's own terms, no attainable increase in p substitutes for detection once the horizon is long.

6.2 Failure detection mechanisms

Six mechanisms appear in the literature, with complementary coverage and cost. Their combination matters more than any one of them, because their blind spots differ.

1. ***Vision-language success verification*** asks a VLM whether the intended effect obtains (W. Huang, Xia, et al., 2022; Z. Liu et al., 2023). It is a general approach that requires no task-specific engineering and shares the calibration challenge discussed in Sect. 2.1. The verifier's confident errors are indistinguishable from its confident correct assessments.

2. ***Precondition and effect monitoring*** checks explicit logical conditions before and after each action (Y. Guo et al., 2024; Raman et al., 2024). It is precise and observable where the conditions are correctly authored, and it thereby reintroduces a simplified version of the domain model, a trade-off we believe is worthwhile, as the required model is significantly less extensive than an entire planning domain.

3. ***Force and proprioceptive signatures*** detect contact events, missed grasps, and jams with low latency and high reliability (Haddadin & Croft, 2016). They are the most reliable detection channel available and the most neglected: they are almost entirely absent from the systems of Sect. 4, and a 2026 vision-language-action framework that fuses tactile sensing directly into the policy for contact-rich manipulation (K. Zhang et al., 2026) is the first substantial attempt we have seen to give a foundation-model policy access to the modality that most reliably reveals a failed manipulation.

4. ***Anomaly detection*** over execution traces (Khalastchi & Kalech, 2019) flags departure from a learned norm without requiring that norm to be defined semantically; it is effective for hardware faults and weak for semantic errors, since a wrong action executed correctly is not anomalous. Semantic anomaly detection is the exception, using a language model's contextual breadth to recognize system-level edge cases that no individual component of the autonomy stack would report as a fault (Elhafsi et al., 2023). It is the only mechanism in the surveyed literature aimed directly at the semantic hazard class of Sect. 7.1, and it deserves more attention than it has had.

5. ***Self-consistency*** and log-probability signals use the planner's own output distribution, either sampling several plans and scoring their agreement or thresholding on token likelihood (Kadavath et al., 2022; Lin et al., 2023). They are inexpensive and correlate weakly with accuracy.

6. ***Human report*** is still the most accurate detector in a collaborative environment and is often not utilized to its full potential. The operator is actively observing and can promptly identify any mistakes made by the robot; a system that enables easy reporting of these errors and actions can achieve nearly flawless detection with minimal engineering expense.

6.3 Recovery strategies

Identifying an issue without a recovery strategy changes a quiet failure into a complete halt, which is safer but not necessarily more effective. We suggest a four-level escalation, arranged by expense and by the extent of the commitment adjusted, as illustrated in Fig. 5.

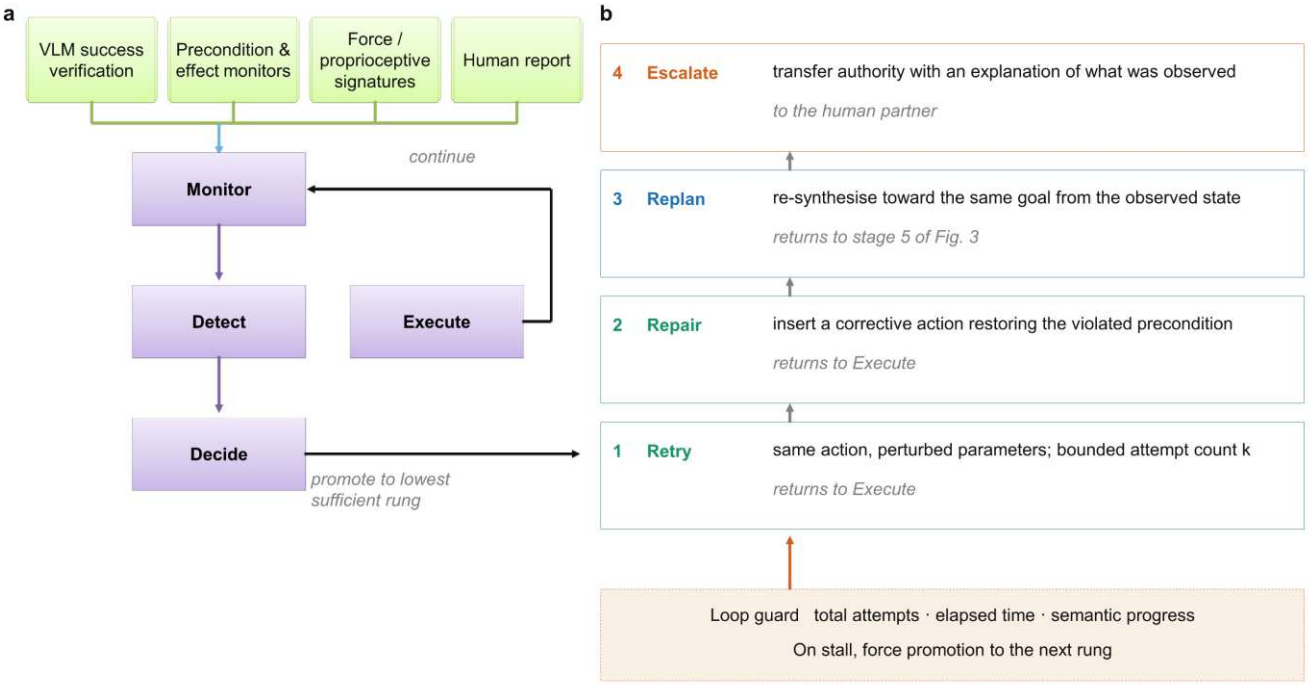


Figure 5. *Closed-loop detection and the recovery escalation hierarchy. a, Execution cycle, with four complementary detection channels entering the Monitor node. b, Four-rung escalation ladder; each rung revises a strictly larger commitment than the one below, from the parameters of one action, to the state that action assumed, to the remainder of the plan, to the question of who holds authority. The Decide node promotes to the lowest sufficient rung. The dashed guard monitors total attempts, elapsed time and semantic progress, forcing promotion when a rung stalls; its absence produces the repetition-loop pathology observed in deployment*

Retry attempts the same task again with perturbed parameters and is suitable when execution fails stochastically, such as due to a slipped grip; it must be constrained by the number of attempts, because unlimited retries of the same plan in the face of systematic error is a livelock condition. ***Repair*** involves an additional action that fixes a violated precondition and opens up the gripper; it is the level considered by (Raman et al., 2024). ***Replan*** involves abandonment of the rest of the old plan and generation of a new plan towards the same goal, starting from the current state, which is the proper response to a state that has become divergent from local repair (Y. Guo et al., 2024; Skreta et al., 2024). ***Escalate*** transfers authority to the human with an explanation of what was attempted and what was observed (Z. Liu et al., 2023).

Two characteristics of this hierarchy are more significant than its particular levels. It needs to be monitored: without tracking total attempts, elapsed time, and semantic advancements, a system could endlessly oscillate

between retrying and replanning, which is the most frequent issue observed in documented closed-loop systems. Additionally, it must be able to be audited: every transition should have a log detailing the evidence that triggered it, both because this serves as the foundational material for the failure-case reporting suggested in Sect. 8.5 and because it will be the documentation required by regulators (B. Li et al., 2023).

Table 6. *Failure taxonomy, detection mechanisms, recovery strategies, and residual risk*

Failure type	Example	Detection mechanism	Recovery tier	Residual risk	Ref.
Grounding error	Wrong object selected	VLM verification; human report	Replan	Verifier shares the grounding error	(W. Huang, Xia, et al., 2022; Z. Liu et al., 2023),
Referential ambiguity	Two candidate brackets	Confidence threshold; conformal set size	Escalate (clarify)	Over-querying degrades fluency	(Ren et al., 2023)
Infeasible parameterisation	Grasp outside workspace	Kinematic pre-check	Repair or replan	Check omitted for VLA action tokens	(W. Huang et al., 2023; M. J. Kim et al., 2024)
API hallucination	Call to non-existent skill	Static signature and range check	Replan	Valid call, invalid semantics	(Liang et al., 2023; Singh et al., 2022)
Execution slip	Grasp fails; object dropped	Force/proprioceptive signature	Retry (bounded)	Undetected without force sensing	(Haddadin & Croft, 2016)
Precondition violation	Container still closed	Precondition monitor	Repair	Requires authored conditions	(Y. Guo et al., 2024; Raman et al., 2024)
Plan-order error	Step assumes prior action	Effect monitoring	Replan	Detected late in the sequence	(Y. Guo et al., 2024)
Human-state violation	Motion into the predicted reach set	Separation monitoring; intent prediction	Retry after yield	Prediction horizon too short	(Mainprice & Berenson, 2013;

					Zacharaki et al., 2020)
Semantic drift	Task completed as described, not as intended	Human report	Escalate	No automatic detector exists	(J. D. Lee & See, 2004)

Table 6 makes one argument: detection coverage is uneven, and the unevenness is not accidental. The upper rows grounding errors, referential ambiguity, infeasible parameterisation, API hallucination - are all detectable before or at actuation using mechanisms that already exist, and the residual-risk column records only how imperfectly those mechanisms work. The lower rows differ in kind. Execution slips are detectable quickly and reliably through force and proprioceptive sensing, a channel largely absent from the systems of Sect. 4, so failures a 1980s assembly cell would have caught pass unnoticed in a 2026 language-driven system. Plan-order errors surface only late, after the compounding of Sect. 6.1 has done its damage. Semantic drift has no automatic detector at all: the robot performs the task exactly as instructed but not as intended, no monitor is violated, no force threshold crossed, and the only evidence is the person standing beside it. That row is the strongest argument for treating human reports as a primary detection channel rather than a fallback.

6.4 Robust execution in dynamic environments

Dynamism in a collaborative workspace primarily stems from humans, who handle objects, obstruct the view, and alter the task while the robot is still in the planning phase. There are three characteristics that set apart systems capable of adapting. ***Perceptual persistence***: a scene representation that accumulates across observations (Gu et al., 2024; Shafiullah et al., 2023) permits reasoning about objects currently occluded, where single-frame perception cannot. ***Plan flexibility***: partial-order plans (Sect. 5.1) allow reordering in response to the human's current action without full replanning. ***Reactive layering***: Rapid safety and compliance mechanisms must operate beneath the semantic planner at the required control rates, since no foundation-model loop can respond to contact at the necessary speed. This leads to an architectural implication of a three-rate system, semantic planning operating on the scale of seconds, execution monitoring at the scale of tenths, and safety and compliance at the scale of milliseconds, and misunderstanding among these rates is a frequent cause of design mistakes.

6.5 Limits of current methods

There are four fundamental, tangible constraints.

1. *Detection latency*: VLM verification increases the time taken for each check by hundreds of milliseconds to several seconds, necessitating a decision between verifying each individual step and checking at broader intervals, where broader checkpoints permit error propagation.

2. *False-positive cost*: A verifier that indicates success as failure can prompt unnecessary recovery efforts, which reduce throughput and, more critically, undermine operator trust. Systems that report detection accuracy seldom provide the false-positive rate separately, even though this statistic is crucial for determining usability.

3. *Unbounded loops*: as noted, guards are frequently absent.

4. *Unverified recovery*: The recovery policy is created by the same model that generated the unsuccessful plan, meaning that a consistent misunderstanding remains intact during recovery. Additionally, no evaluated system assesses its recovery actions against any standards other than those it initially used.

A benchmark set for 2026 on an affordable platform clarifies the practical implications. Assessing vision-language-action policies amidst realistic embodiment variations reveals typical failure patterns, such as grasp instability, repetitive loops, state mismatches, and precision misalignment, which are often overlooked in controlled benchmarks. The authors emphasize that failure behavior, execution stability, and recovery capability should be evaluated alongside overall task success (Y. Yu & Qiu, 2026). The repetition-loop failure reflects the unchecked escalation issue noted in Sect. 6.3, as observed in real-world scenarios. These constraints prompt an inquiry into the extent to which safety and trust can actually be demonstrated.

7. Safety, Trust, Robustness, and Evaluation

Fig. 6 outlines the framework used in this section, linking hazard categories to the mitigation strategies that address them, as well as the criteria used to evaluate each category.

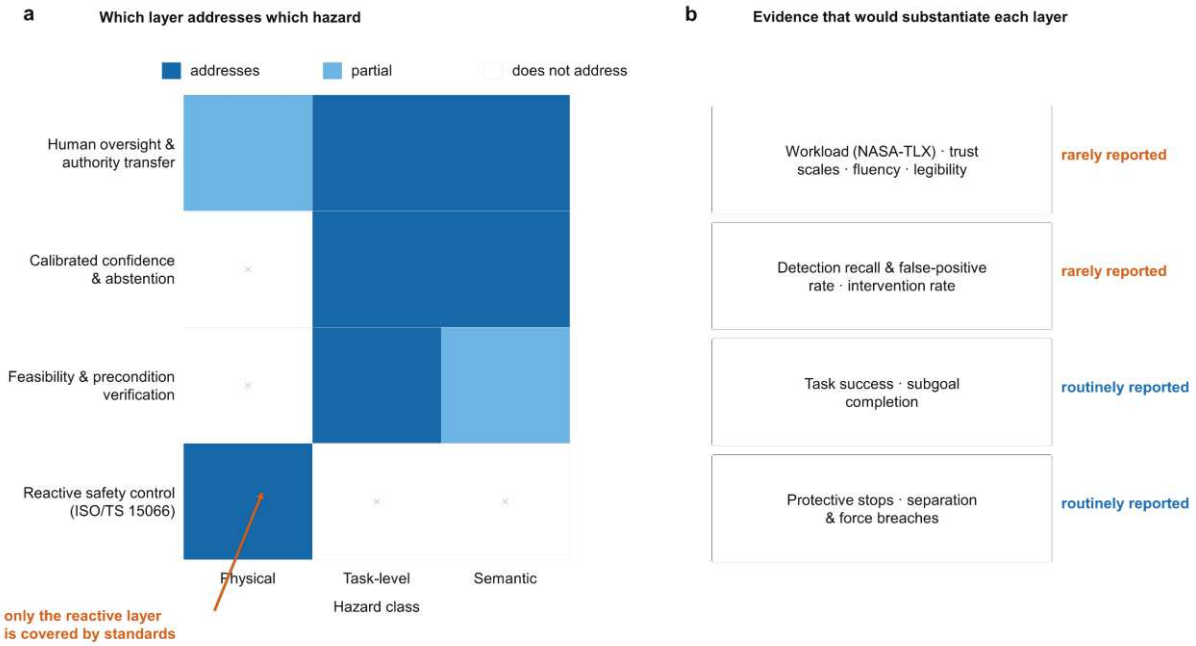


Figure 6. *Hazard coverage and the evidence that would substantiate each mitigation layer.* ***a****, which of the four mitigation layers addresses which of the three hazard classes? The reactive safety layer, governed by ISO/TS 15066, addresses only the physical class, so task-level and semantic hazards fall entirely to the three layers that current systems implement weakly or not at all.* ***b****, The evaluation evidence corresponding to each layer, with observed reporting practice. Evidence for the two upper layers is rarely reported, and the quantitative form of the gap is analyzed in Sect. 7.4*

Fig. 6 is built to display a negative result. The hazard classes on the left are ordered by distance from physical contact, and the mitigation layers in the center by increasing scope. The pattern of filled cells is the section's central argument: the lowest layer, reactive safety control, is populated only for the physical class. Every assurance that the collaborative-robot safety standards were written to provide sits in that one cell. Task-level hazards, in which the robot applies correct force to the wrong object, and semantic hazards, in which it completes the instruction while misreading the intent, produce neither force exceedance nor separation breach, so the reactive layer misses both by design rather than by defect. The three layers that would address them are precisely the three that current systems implement weakly or not at all. The right-hand column names the evidence that would substantiate each layer, which is how Fig. 6 connects to Table 7: a system reporting only protective stops and task success has validated the bottom layer and nothing above it.

7.1 Safety risks and failure modes

The collaborative operation can be defined in accordance with the ISO 10218 series of standards and, more specifically, for collaborative modes, ISO/TS 15066. These include safety-rated monitored stop, hand guiding, speed and separation monitoring, power and force limiting, and biomechanical limits by body region. The approach is well developed with respect to physical hazards and is based on a presupposition that a semantic planner ignores: that the robot's task program is deterministic and analyzable.

It is essential to identify three distinct hazard categories. The reactive layer deals with physical hazards such as impact, clamping, and force exceedance, and the relevant standards are applied appropriately and effectively (Haddadin et al., 2009; Zacharaki et al., 2020). *Task-level hazards* - the application of correct force to the wrong object, the execution of actions in the wrong order, and the silent completion of a task that was never the one intended - result in no force exceedance and are undetected by the reactive layer; however, they lead to damage, scrap, and, in the clinical or assisting context, they inflict physical harm. *Semantic hazards* are the most unique: the system performs the task according to the request, but violates its intention. Requesting to clean up the work surface, which results in the part being removed from its fixture, is a semantic hazard and represents a robotic version of the specification problem described in (Amodei et al., 2016). No current standard addresses the second or third class, and no current architecture detects them without human report.

7.2 Trustworthiness and Explainability

Trust research clearly indicates that the goal is to calibrate trust to match actual reliability, rather than to maximize trust (Hancock et al., 2011; Hoff & Bashir, 2015; J. D. Lee & See, 2004). Both over-trusting and under-trusting are failures, and a system that shows consistent confidence can lead to both issues over time.

Three explanation functions are required, and they vary in timing. ***Anticipatory*** Explanation conveys purpose prior to acting, which is the role that clear motion fulfills non-verbally (Dragan et al., 2013). The current language now explicitly permits this. ***Concurrent*** explanation reports provide information during execution and clarify mode confusion described in Sect. 2.2. In contrast, a ***post-hoc*** explanation explains a failure after it occurs, which is the main difference (Z. Liu et al., 2023) provides. The fact that language models can cheaply generate explanations has led to a particular risk that is not addressed in any surveyed literature: a fluent explanation carries a lot of persuading power even when it is inaccurate, and therefore, an elaborate explanation for a failure will calibrate user trust less accurately than no explanation at all. Evaluation of explanation quality should be done on ground-truth information about the system state (Anjomshoae et al., 2019; Sakai & Nagai, 2022).

7.3 Robustness under uncertainty

Uncertainties that occur in the semantic planner are heterogeneous in nature, and mixing the types of uncertainties can be considered a material mistake rather than a terminology issue. By using the traditional classification (Kendall & Gal, 2017), ***aleatoric*** uncertainty includes variability, sensor noise, contact stochasticity, and human variability, and should be addressed with robust and compliant control. ***Epistemic*** uncertainty arises from the model's ignorance of the unknown object or distribution, and it can be reduced by gaining information; in the case of a collaborative robot, this means asking for information.

Four families of techniques are available. ***Conformal prediction*** (Angelopoulos & Bates, 2022; Vovk et al., 2005) offers distribution-free coverage guarantees under exchangeability and is the most viable approach for maintaining these guarantees when integrated with a foundation model. KnowNo (Ren et al., 2023) applies LLM planning to generate predictions that determine whether the robot should act or ask questions; thus, this method is the clearest illustration of P5 currently found in the literature. The weakness of this approach is its reliance on exchangeability, because altering the configuration of the workcell breaks this assumption and reduces coverage. ***Ensembles and Bayesian approximations*** (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017) estimate epistemic uncertainty from disagreement, but at a computational cost that is prohibitive for large models. ***Post-hoc calibration*** (C. Guo et al., 2017) corrects systematic overconfidence using held-out data, which must be representative of deployment conditions - a demanding requirement in the open world. ***Verbalized and internal-state confidence*** (Kadavath et al., 2022; Lin et al., 2023; Xiong et al., 2024) is cheap and, on current evidence, insufficiently calibrated to act on alone.

7.4 Evaluation metrics

The evaluation practice identified in the surveyed literature is its most significant flaw. The metrics reported tend to focus mainly on task success rates, which are essential but fundamentally inadequate: they conflate different types of failures, obscure the false-positive rate of any detection mechanism, and offer no insight into human involvement. Table 7 outlines the constructs that a collaborative semantic planner ought to report.

Table 7. *Evaluation metrics for collaborative semantic planning, with definitions and observed reporting frequency in the surveyed literature*

Metric	Definition	What it captures	Reporting frequency	Ref.
Task success rate	Fraction of episodes achieving the goal	Aggregate capability	Near-universal	(Ahn et al., 2022; Brohan et al., 2023)
Subgoal completion	Fraction of plan steps achieving the intended effect	Where in the horizon failure occurs	Occasional	(Mees et al., 2022; Shridhar, Thomason, et al., 2020)
Detection recall	Fraction of true failures detected	Silent-failure exposure	Rare	(Y. Guo et al., 2024; Z. Liu et al., 2023)
Detection false-positive rate	Fraction of successes flagged as failures	Throughput and the cost of monitoring	Very rare	(W. Huang, Xia, et al., 2022)
Intervention rate	Human interventions per task	Real autonomy level	Rare	(Ren et al., 2023)
Time to recovery	Elapsed time from failure to resumed progress	Practical cost of failure	Very rare	(Raman et al., 2024)
Safety violations	Protective stops, separation, or force breaches	Physical hazard exposure	Rare	(Zacharaki et al., 2020)
Fluency measures	Robot and human idle time; concurrent activity; functional delay	Quality of joint action	Rare in LLM/VLM work	(Hoffman, 2019)
Legibility	Probability of correct goal	Predictability to the partner	Almost absent	(Dragan et al., 2013)

	inference from partial trajectory			
Workload (NASA-TLX)	Subjective workload across six dimensions	Cognitive cost to the operator	Rare in LLM/VLM work	(Hart & Staveland, 1988)
Trust (validated scale)	Trust perception on a validated instrument	Calibration of reliance	Rare in LLM/VLM work	(Jian et al., 2000; Muir, 1994; Schaefer, 2016)
Situation awareness	Operator's model of robot state and intent	Mode confusion exposure	Almost absent	
Failure-behavior taxonomy	Distribution over grasp instability, repetition loops, state mismatch, and precision misalignment	Which failure mode dominates, not merely how often	Newly proposed (2026)	(Y. Yu & Qiu, 2026)

Table 7 is read through its second-to-last column. The definitions are uncontroversial; the informative content is the observed reporting frequency, which records our qualitative judgment over the included set rather than a systematic count, since reporting is often partial or buried in appendices. We treat the ordering it induces, not the individual labels, as the finding. On that reading, the metrics fall into two groups: quantities derivable from the robot's own logs are reported almost universally, and quantities depending on an honest report from the monitor or on a human participant are rarely reported at all. The detection false-positive rate is the most consequential omission, because it determines whether a monitor is usable at all: a verifier with excellent recall and a twenty percent false-positive rate will stop the line often enough to be abandoned, and a paper reporting recall alone tells the reader nothing about this. Intervention rate and time to recovery reveal what the autonomy is worth in operation, while fluency, legibility, workload, and situation awareness characterize the collaboration rather than the robot; their near-absence is the quantitative form of the disconnect identified in Section. 5.3.

7.5 Benchmarking and validation gaps

Existing benchmarks were built for adjacent but different tasks. ALFRED and ALFWorld (Shridhar, Thomason, et al., 2020; Shridhar, Yuan, et al., 2020) test simulated instruction following without a collaborator; RLBench, Meta-World, and CALVIN (James et al., 2020; Mees et al., 2022; T. Yu et al., 2020) test manipulation skill and language conditioning without collaboration; BEHAVIOR (C. Li et al., 2023; Srivastava et al., 2022) offers realistic household activity without collaboration; PlanBench (Valmeekam, Olmo, et al., 2023) evaluates formal

planning without embodiment; and Habitat 3.0 and Watch-And-Help (Puig et al., 2020, 2024) come closest to a collaborative setting but remain simulated.

This is beginning to change. A collaborative assembly benchmark built around a high-precision planetary gearbox task provides both a simulation and a physical dual-arm implementation (H. Liu et al., 2026), the closest approximation yet to the physical collaborative benchmark this review argues is missing, and a 2026 survey of vision-language-action datasets and benchmarks argues that evaluation infrastructure should be treated as a primary research problem rather than a secondary concern (Z. Wang et al., 2026). Neither places a human participant in the loop, which leaves four gaps. There is no established benchmark for collaborative semantic planning with a human participant on physical hardware. Human-subject protocols are absent, so the participant-dependent metrics of Table 7 are not routinely collected. Simulation-only results predominate in contact-rich tasks, where transfer is weakest (Hofer et al., 2021; Zhao et al., 2020). And negative results go unpublished, leaving the field's error taxonomy anecdotal rather than empirical, which a requirement to log failures would remedy at negligible cost. What follows is how these changes might be implemented.

8. Five Practical Solutions

The five proposals listed below have been chosen based on three key criteria: each one tackles a cause identified in **Table 2** rather than merely addressing a symptom; each can be implemented using current technology; and each has an initial action step that a single research group or integrator can initiate within a matter of weeks. They are arranged according to how quickly they may impact the silent-failure issue. Each proposal is set out in prose under four headings - what it is, how it addresses the problem, its benefits, and its risks and limitations - followed by the first action step a group can take. Difficulty, resource requirements, and position within the pipeline of **Fig. 3** are tabulated for all five solutions in Table 8 rather than repeated for each.

8.1 Verifier-in-the-loop neuro-symbolic planning

What it is. A framework where the foundational model is relegated from the role of planner to that of a proposal generator, while a reliable verifier possesses the authority to veto any action before it is executed. The language model generates potential plans; a symbolic layer verifies each action’s preconditions against a simplified domain model, a kinematic layer assesses reachability and collision based on the predicted state, and a temporal-logic monitor ensures that safety invariants are upheld throughout the duration of the plan (Guan et al., 2023; Kambhampati et al., 2024; Kress-Gazit et al., 2009, 2018; B. Liu et al., 2023; Silver et al., 2024). Only approved plans are sent to the controller; rejected plans are returned to the generator with detailed feedback on the specific violation.

How it addresses the problem. It addresses the discrepancy between tokens and continuous representations, as well as the lack of direct formal assurances. Instead of only addressing silent failures after actuation, it transforms them into noticeable failures at the earliest stage possible. According to Definition 1, it explicitly enforces P2 and

adds to P3. Additionally, it maintains the open-world referent resolution that was the reason for utilizing a language model in the first place, as the model continues to make proposals.

Benefits. Guarantees are obtained without having to reinstate the full costs of domain authoring, since the model required for precondition verification is considerably smaller than that required for planning. Violations are formatted so that machines can read them, facilitating an automatic feedback loop to the generator. The verifier generates exactly the audit artifact needed for regulatory documentation (B. Li et al., 2023). Additionally, the architecture is resilient; a less robust model merely necessitates more rounds of proposals.

Risks and limitations. The soundness of the verifier is limited by its domain model, meaning that any unmodeled precondition goes unchecked, leading to a false sense of security in the architecture. Rejection loops introduce additional latency, and a model that continually suggests plans that violate constraints can become stalled. Most crucially, verification ensures feasibility and safety, yet it does not account for intent: a plan may be completely viable while still pursuing an incorrect objective, so this approach fails to tackle semantic hazards (Sect. 7.1) and should be used in conjunction with Sect. 8.2.

First action step. Take the existing skill library and, for each primitive, write explicit precondition and effect specifications in PDDL or an equivalent formalism; then instrument the current planner to check each generated step against these specifications and log violations without blocking them. The distribution of logged violations over one week of normal operation quantifies the problem and justifies the remaining work.

8.2 Calibrated uncertainty with principled abstention

What it is. An estimate of confidence in both the language understanding and the visual grounding associated with each plan, based on actual success, along with a policy that uses low confidence as grounds for requesting clarification or for delegating authority but not for performing an action. Conformal prediction (Angelopoulos & Bates, 2022; Vovk et al., 2005) offers the most robust current foundation, producing prediction sets in which the set size determines the action: a single-element set indicates acting, while a set with more elements suggests inquiring, following (Ren et al., 2023).

How it addresses the problem. It targets the root cause identified as the most significant and most solvable in Sect. 2.1. It is the only recommendation in this document that deals with failure before it happens and meets Condition P5 from Definition 1 that none of the other solutions meet. It also provides an arbitration signal demanded by Sects. 5.4 and 8.4.

Benefits. Coverage guarantees are independent of the distribution under exchangeability, meaning they hold without any assumptions about the underlying model. Inquiry is significantly less costly than making mistakes in nearly all collaborative scenarios. Confidence that is calibrated and presented to the operator fosters a reliable sense of trust (J. D. Lee & See, 2004), addressing Sect. 7.2 as a side effect. And the mechanism is model-agnostic, so it survives model upgrades.

Risks and limitations. Exchangeability can break down when there is a distribution shift, leading to a silent degradation in coverage, a failure mode that replicates the initial issue at a higher level, which means that shift detection should be included during deployment. Over-querying poses a real risk: a robot that frequently asks questions is more problematic than one that occasionally makes mistakes, and the threshold γ serves as a usability factor, not simply a statistical one. Proper calibration requires a representative held-out dataset, which is challenging to obtain in the open world and must be updated regularly.

First action step. Monitor the current planner to record, for each decision made, its top-k potential outputs, their scores, and the actual outcome. Logging this data over two hundred episodes produces a reliability diagram that measures current miscalibration, and this same log serves as the calibration set for the conformal method.

8.3 Grounded execution monitoring with a structured recovery library

What it is. Compulsory confirmation of the desired outcome following each planning stage should use multiple detection channels: vision-language verification, precondition monitoring, force and proprioceptive signatures, and a consistently accessible human report linked to the guarded four-tier recovery hierarchy outlined in Sect. 6.3 and illustrated in **Fig. 5**

How it addresses the problem. This text serves as a straightforward response to silent failure and the cumulative reasoning presented in Sect. 6.1. It transforms the rapid decrease in open-loop reliability into a manageable form and generates the execution logs necessary for Sects. 7.5 and 8.5.

Benefits. Failure detection identifies the specific step that triggered the failure instead of the point at which the effects were observed, enabling both automatic recovery and effective diagnosis. Recovery maintains throughput, while a protective stop interrupts it. Multi-channel detection is resilient to the miscalibration of any single verifier, and force sensing, in particular, offers a channel that is not reliant on the models that generated the error. The logs serve as a valuable research resource on their own.

Risks and limitations. The main expense is verification latency, and checking at each step might not be practical within the necessary cycle times, necessitating checkpoints to allow some propagation. False positives negatively impact both trust and throughput, making it essential to explicitly report and manage the false-positive rate. The recovery strategy is produced by the same model that encountered the failure, meaning that systematic misconceptions persist even after recovery. Loops without safeguards can enter a livelock state.

First action step. For the three most commonly performed skills, specify clear, observable success criteria and set up a single detection method for each, usually the most cost-effective option, such as a force threshold or a fixed-view image assessment. Evaluate the detection recall and, independently, the false-positive rate across one hundred executions before expanding the coverage.

8.4 Human-aware and personalized planning with bounded memory

What it is. Incorporation of the human partner as a first-class element of the planning state: a predicted reach set over horizon τ (Koppula & Saxena, 2016; Mainprice & Berenson, 2013), entering as a constraint, a legibility term (Dragan et al., 2013) entering the trajectory objective, and a bounded, inspectable, editable interaction memory supporting personalization (Nikolaidis et al., 2017; J. Wu et al., 2023). The memory is capped in size, expressed in natural language so the operator can read it, and is directly editable and erasable by that operator.

How it addresses the problem. It addresses the gap mentioned in Sect. 5.3 between the literature on semantic planning and that concerning human-robot collaboration, while fulfilling P3 and P4 of Definition 1, which the architectures discussed in Sect. 4 largely overlook. It transforms a safe robot into a collaborative partner.

Benefits. Anticipating the human's next reach eliminates the obstructive-but-safe motion that operators experience as interference, improving fluency (Hoffman, 2019) and reducing protective stops. Legible motion reduces the monitoring load and supports situation awareness (Endsley, 1995). Bounded, readable memory delivers personalization while satisfying the transparency and data-protection expectations of workplace deployment. An operator who can read and correct what the robot believes about them serves as an additional detection channel.

Risks and limitations. Motion forecasting is not dependable beyond brief timeframes and diminishes in quality when there is occlusion, so caution is required when confidence in the prediction is low this relates to Sect. 8.2. The balance between legibility and efficiency is task-specific and varies in importance. Personalization can lead to drift, unpredictable behavior when operators change shifts, and exposure of private information; measures such as bounding, inspectability, and individual operator limitations help reduce these issues but do not completely resolve them.

First action step. Monitor the workcell to capture human pose data during standard collaborative activities over a two-week period, then train a task-conditioned reach-set predictor using the data. Assess it offline against reserved sessions before allowing it to affect any planning.

8.5 A standardized collaborative benchmark and reporting protocol

What it is. A benchmarking and reporting framework for collaborative semantic planning within the community, including a physical task suite that involves a human collaborator, a set of required metrics based on Table 7, which encompasses human-centered measures, an obligatory human-subject protocol, the necessary publication of failure logs, and a clear statement of the conditions for simulation versus real-world evaluations.

How it addresses the problem. The method tackles the shortcomings found in Sect. 2.3, but does not focus on any specific algorithm. This is because the fundamental issue we have identified within the discipline is that the challenges are primarily methodological.

Benefits. The ability to compare different methods is essential for understanding which techniques are effective, something that is currently unclear in the field. Requiring reporting of false-positive detection rates, intervention rates, and workload would quickly highlight the gaps illustrated in Table 7. Publishing failure logs transforms an anecdotal classification of errors into a data-driven one. Additionally, a standardized evaluation protocol is a necessary step in the certification process mentioned in Sect. 9.5 (B. Li et al., 2023).

Risks and limitations. Benchmarks can skew the research they evaluate, and an inadequately selected task suite can lead to overfitting, as outlined in Sect. 2.3. Physical benchmarks are costly and challenging to replicate across different locations, making hardware variability a potential source of confusion in comparisons. Protocols involving human subjects need ethics approval and significantly increase costs. Moreover, widespread adoption requires a collective effort that no single entity can achieve, which is why the initial step mentioned below is intentionally one-sided.

First action step. Implement the reporting procedure independently in the group's upcoming publication: disclose the full range of metrics from Table 7, list all failures in detail, and clarify the evaluation criteria. A single well-cited paper that does this establishes the norm more effectively than a consortium announcement, and it requires no external agreement to begin.

8.6 Comparative synthesis

These five solutions cannot be considered to be independent of each other; the sequence in which they are adopted is significant. The calibrated uncertainty (8.2) is the enabling condition: it provides the triggering device for verification (8.1), the confidence indicator that will determine at what point the hierarchy of recovery would move up (8.3), the arbitration indicator for authority delegation (8.4), and the confidence reporting requirement that a benchmark (8.5) would impose. Calibration of uncertainty is also the most economical and quickest to adopt; hence, the suggested order is 8.2, followed by 8.3, then 8.1, then 8.4, while 8.5 can be pursued in parallel since it takes several years.

There are two dependencies that need to be emphasized. First, verification (8.1) and monitoring (8.3) complement each other but do not replace each other since one detects unrealistic plans before action, while the other detects realistic plans that nevertheless failed. Second, human-aware planning (8.4) is the only approach that improves the quality of cooperation rather than its reliability; a system designed according to 8.1-8.3 will be reliable but unpleasant to cooperate with. Table 8 below compares the approaches discussed above.

Table 8. *Five practical solutions compared on benefits, risks, difficulty, resources, first action step, and pipeline stage*

#	Solution	Principal benefit	Principal risk	Difficulty	Resources	First action step	Pipeline stage	Ref.

1	Verifier-in-the-loop neuro-symbolic planning	Guarantees without full domain authoring	Unmodelled preconditions give false assurance	Moderate	2 engineers, 3–6 months; planner and collision checker	Write preconditions for the existing skill library and log violations without blocking	Stage 6 (verification)	(Kambhampati et al., 2024; Kress-Gazit et al., 2009; B. Liu et al., 2023)
2	Calibrated uncertainty with principled abstention	Prevents errors before they are made; enables abstention	Coverage degrades silently under a distribution shift	Low–moderate	1 engineer, 1–3 months; ~200 labeled episodes	Log top-k candidates and outcomes; plot a reliability diagram	Stages 3 and 6 (confidence)	(Angelopoulos & Bates, 2022; Ren et al., 2023; Vovk et al., 2005)
3	Grounded execution monitoring with recovery library	Converts a silent failure into a detected, recoverable failure	Verification latency and false positives	Moderate–high	2 engineers, 6–9 months; force sensing; low-latency VLM	Define success conditions for the three most-used skills; measure recall and false-positive rate	Stage 8 (monitoring)	(Y. Guo et al., 2024; W. Huang, Xia, et al., 2022; Z. Liu et al., 2023; Raman et al., 2024)
4	Human-aware and personalized planning with	Fluency and legibility, not merely safety	Prediction unreliable at long horizon; personalization drift and privacy	Moderate	2 researchers, 4–6 months; human tracking;	Record human pose for two weeks; fit and offline-	Human-state band (stages 3, 5, 6)	(Dragan et al., 2013; Hoffman, 2019; Koppula & Saxena, 2016; J.

	bounded memory				ethics approval	validate a reach-set predictor		Wu et al., 2023)
5	Standardized benchmark and reporting protocol	Comparability; empirical error taxonomy	Benchmark overfitting requires collective action	High (organizational)	Consortium; reference hardware; ethics infrastructure	Adopt the full metric set and publish failure logs in the next paper	External (governs all stages)	(Mees et al., 2022; Puig et al., 2024; Shridhar, Thomason, et al., 2020)

Table 8 is designed to be utilized rather than simply read. Its columns are arranged so that a group looking to determine where to start can navigate from left to right through the decision: what the intervention provides, the risks involved, the effort required, the human and equipment costs, and what can be initiated on Monday. Two comparisons across the rows hold practical significance. First, the difficulty and resource columns do not correlate with the benefit column: the second row merges the lowest difficulty and minimal resource need with a benefit that no other row offers, which is the prevention of errors before they occur rather than detecting them after the fact. Second, the pipeline-stage column indicates that the five proposals do not vie for the same position. In the stack, verification is at stage 6, monitoring is at stage 8, uncertainty spans stages 3 and 6, human-awareness is integrated through the human-state band, and the benchmark lies entirely outside the pipeline, which explains why they enhance rather than replace each other. The first-action-step column was crafted to be actionable without the need for funding, new hardware, or ethics approval in four of the five rows, based on the belief that any proposal a group cannot start this quarter is one it will likely never initiate.

9. Open Challenges and Future Directions

9.1 Grounding and hallucination

Hallucination in a language model is an annoyance; in an embodied planner, it is a physical act. Retrieval grounding, affordance filtering, and scene-graph constraint (Ahn et al., 2022; Gu et al., 2024; Rana et al., 2023) reduce its incidence without altering its nature, because they filter outputs rather than changing the mechanism by which a language model makes confident assertions. The research program we would advocate is to make grounding provenance-carrying: every referent in a plan tagged with the observation that supports it and with an estimate of the reliability of the channel that supplied it, so that an ungrounded referent becomes a structurally detectable condition rather than a statistical outlier. Conformal prediction (Sect. 7.3) quantifies uncertainty about the referent given an exchangeable calibration sample, and is the correct instrument for the abstention decision itself. What it does not quantify is confidence in the channel, which is precisely the quantity that varies when a detector operates under occlusion or degraded illumination, and precisely what fails when exchangeability breaks.

Representing channel reliability explicitly alongside the referent is, in our assessment, the most consequential open representational problem in this area, and no surveyed system attempts to do so.

9.2 Real-time constraints

The three-rate architecture structure presented in Sect. 6.4 is merely a way of addressing latency in a foundation model, not a resolution. A semantic planner that is incapable of revising itself in accordance with the timing of the person's activity cycle, which is about one to two seconds in collaborative construction, cannot take part in fluency (Hoffman, 2019). Quantization, low-rank adaptation, distillation, and speculative decoding (Dettmers et al., 2023; Hu et al., 2022; Leviathan et al., 2023) narrow the gap. Hierarchical architectures, in which the large model is used only for occasional planning while the small distilled model handles action-level decision-making, seem to be more successful than uniform acceleration. In 2026, there have been some practical advancements in this area. Verbalizable latent planning has condensed the reasoning step, enabling the vision-language-action model to operate at much lower latency (C.-P. Huang et al., 2026); on-device planning reduces unnecessary inference to stay within an embedded compute budget (J. Lee et al., 2026). The combination of these factors indicates that the latency barrier is more of an engineering limitation than a core issue. The unresolved question is which choices truly necessitate a large model, and no systematic study of this has been conducted.

9.3 Sim-to-real transfer

Simulation allows for cost-effective collaborative evaluation, but the transfer of contact-rich manipulation falls short (Hofer et al., 2021; Zhao et al., 2020). While domain randomization (Tobin et al., 2017) effectively addresses visual and dynamic discrepancies, it does not tackle the critical issue: simulated humans do not act like real ones, and a collaboration policy tested against a scripted avatar has yet to be validated. Habitat 3.0 (Puig et al., 2024) signifies progress in human simulation, yet the more challenging task of replicating the variability, errors, and adaptability of a real partner remains unresolved. The immediate recommendation is procedural rather than technical: report results from simulations and real-world scenarios separately and avoid combining them. We also narrow one challenge that the 2026 literature has partly overtaken. Multimodal grounding beyond vision, often listed as future work, is now being attempted through tactile fusion into vision-language-action policies (Sect. 6.2); the residual and harder problem is integrating force and tactile channels into the verification path of Fig. 3 rather than only into the policy, so that contact evidence can contradict a language-derived plan before it is executed.

9.4 Human factors, datasets, and benchmarks

Acceptability does not follow from capability. Perceived surveillance arising from the human tracking that human-aware planning requires, and the erosion of calibrated trust described in Sect. 2.2, both bear on whether a system is used as intended (Hoff & Bashir, 2015; J. D. Lee & See, 2004). Yet the LLM- and VLM-driven robotics literature contains very little human-subject research, and what exists is small-scale, unblinded, and often run

with the developers as participants. The decisive absence is a longitudinal field study with real operators, for which there is no methodological obstacle.

Beyond the benchmark proposed in Sect. 8.5, the field lacks a failure dataset. Aggregated demonstration data has markedly accelerated policy learning (O'Neill et al., 2024); pooling recorded failure incidents, each stating what was instructed, what was intended, what was observed, what went wrong, and how it was resolved, would make detection and recovery models trainable that currently are not. The 2026 literature has converged on the same view from the data side, arguing that data engines and settled evaluation protocols now matter more to progress than further architectural change (Z. Wang et al., 2026). The failure-log proposal is the collaborative-robotics form of that argument, and it is the highest-value dataset the community could assemble: it requires only that failures be recorded and shared rather than discarded.

9.5 Formal guarantees and trustworthy deployment

The deepest challenge is that the safety case for a collaborative robot is currently built around a deterministic control program, whereas a foundation-model planner is not. Three routes exist. *Architectural containment* places the model outside the safety-critical envelope, allowing it to propose while a verified component disposes; this is Sect. 8.1, it is achievable today, and it bounds the guarantee of what the verifier covers. *Statistical assurance* accepts a probabilistic claim with quantified coverage, as conformal methods provide (Angelopoulos & Bates, 2022; Ren et al., 2023); this requires regulators to accept a form of evidence that current robot safety standards do not contemplate. *Runtime verification* (Bartocci et al., 2018) monitors temporal-logic properties during execution (Kress-Gazit et al., 2018), guaranteeing detection rather than prevention. None of the three is sufficient alone, and the field has not yet articulated what a composite safety case would look like, which is, in our assessment, the most important unaddressed question in the area, and the one whose resolution most directly determines whether these systems leave the laboratory.

10. Conclusion

Collaborative robots can now be instructed in natural language, but they cannot yet be relied upon to act on that instruction. The obstacle is narrower than a general shortfall in capability, and identifying it precisely has been the purpose of this review. Foundation-model planners do not fail conspicuously, as a classical planner does when its domain model is incomplete; they emit fluent, well-formed action sequences whose defects, a non-existent referent, an unreachable grasp, a step presupposing an effect that never occurred, become observable only at execution, and only if some component is looking. Definition 1 states the five conditions such a planner must satisfy, and the assessment in Sects. 4 and 8 find groundedness partially enforced, feasibility inconsistently enforced, safety delegated to a reactive layer outside the planner, and legibility and calibrated abstention essentially unaddressed.

Two consequences follow. The first is quantitative. Separating per-step reliability from detection quality yields a criterion, stated as Proposition 1, under which a less accurate planner that detects and recovers outperforms a more accurate one that does not: the product of detection recall and recovery success must exceed the accuracy deficit normalized by the residual failure rate. Under the independent-step model, the threshold does not depend on plan length, so that it can be evaluated once for a platform; under the positive dependence that manipulation actually exhibits, early detection is worth more than the model credits, while horizon invariance is lost. Either way, the field's prevailing assumption that monitoring is always worth adding holds only above a threshold, and the criterion cannot presently be applied to any surveyed system, because per-step reliability, detection recall, recovery success, and the false-positive rate are rarely reported separately from aggregate task success. Reporting them is the cheapest available intervention and the largest obstacle to cumulative progress.

The second consequence is architectural and runs counter to the direction of travel. Between 2022 and 2026, architectures moved from emitting text to emitting code to emitting action tokens. Generalization improved at every step, and at each step, an interface through which a plan could be inspected was removed, so the systems that generalize best are those on which correctness can be checked least. If that trajectory continues, capability and verifiability will diverge further, and the deployment gap will widen even as benchmark performance improves.

None of the remedies this review identifies requires a research breakthrough. Verification can be reintroduced without surrendering open-vocabulary generalization by demoting the foundation model from planner to proposal generator and giving a sound component the veto. Calibrated abstention requires no retraining and is the single intervention that prevents an error rather than detecting it afterward. The tactile fusion appearing in 2026 policies restores the sensing channel that most reliably reveals a failed manipulation, and which has been available since the 1980s. What is missing is not method but evidence: a collaborative benchmark that places a person in the loop on physical hardware, without which the legibility and abstention conditions of Definition 1 can be argued but not measured; and a composite safety case, assembling architectural containment, statistical assurance, and runtime verification into an argument no current standard yet contemplates.

We therefore expect the decisive results of the coming years to come not from larger models but from architectures that treat their own outputs as hypotheses, and from the reporting practices that would let anyone tell whether they do. A robot that knows when it does not know is a collaborator; one that does not is a hazard with good manners.

Declarations

Funding: No funding was received for conducting this study.

Competing interests: The authors have no relevant financial or non-financial interests to disclose.

Ethics approval: This review synthesizes previously published work and did not involve human participants, their data, or biological material; ethics approval was not required.

Data availability: No datasets were generated or analyzed.

Author contributions *Osama Ali Khan:* Conceptualization; Validation; Visualization; Writing – review & editing; Writing – original draft; Investigation; Methodology. *Abuzar Ghaffari:* Visualization; Writing – review & editing; Writing – original draft.

Statement on AI Writing Assistance

Claude and Grammarly were utilized to enhance grammatical accuracy and refine sentence structure; all AI-generated revisions were thoroughly reviewed and edited for relevance.

References

Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R. J., Jeffrey, K., … Zeng, A. (2022). Do as I can, not as I say: Grounding language in robotic affordances. *Proceedings of Machine Learning Research*, *205*, 287–318. https://arxiv.org/abs/2204.01691

Ajoudani, A., Zanchettin, A. M., Ivaldi, S., Albu-Schäffer, A., Kosuge, K., & Khatib, O. (2017). Progress and prospects of the human–robot collaboration. *Autonomous Robots*, *42*(5), 957–975. https://doi.org/10.1007/s10514-017-9677-2

Alayrac, J. B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., … Simonyan, K. (2022). Flamingo: A visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, *35*, 23716–23736. https://doi.org/10.52202/068431-1723

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). *Concrete problems in AI safety* [Preprint]. arXiv. https://arxiv.org/abs/1606.06565

Angelopoulos, A. N., & Bates, S. (2022). *A gentle introduction to conformal prediction and distribution-free uncertainty quantification* [Preprint]. arXiv. https://arxiv.org/abs/2107.07511

Anjomshoae, S., Najjar, A., Calvaresi, D., & Främling, K. (2019). Explainable agents and robots: Results from a systematic literature review. *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. https://arodes.hes-so.ch/record/4175

Bartocci, E., Falcone, Y., Francalanza, A., & Reger, G. (2018). Introduction to runtime verification. *Lectures on Runtime Verification*, *10457*, 1–33. https://doi.org/10.1007/978-3-319-75632-5_1

Bauer, A., Wollherr, D., & Buss, M. (2008). Human–robot collaboration: A survey. *International Journal of Humanoid Robotics*, *5*(1), 47–66. https://doi.org/10.1142/s0219843608001303

Bender, E. M., & Koller, A. (2020). Climbing towards NLU: On meaning, form, and understanding in the age of data. *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 5185–5198. https://doi.org/10.18653/v1/2020.acl-main.463

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., … Zitkovich, B. (2023). RT-2: Vision-language-action models transfer web knowledge to robotic control. *Proceedings of Machine Learning Research*, *229*. https://arxiv.org/abs/2307.15818

Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jackson, T., Jesmonth, S., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., … Zitkovich, B. (2022). RT-1: Robotics transformer for real-world control at scale. *Robotics: Science and Systems*. https://doi.org/10.15607/rss.2023.xix.025

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Cai, R., Guo, J., He, X., Jin, P., Li, J., Lin, B., Liu, F., Liu, W., Ma, F., Ma, K., Qiu, F., Qu, H., Su, Y., Sun, Q., Wang, D., Wang, D., Wang, Y., Wu, R., Xiang, D., … Zhou, Q. (2026). *Xiaomi-Robotics-0: An open-sourced vision-language-action model with real-time execution* [Preprint]. arXiv. https://arxiv.org/abs/2602.12684

Cai, R., Yamasaki, T., & Xiao, L. (2026). *Vision-language models for deployable social robot navigation: Bridging semantic reasoning and low-level control* [Preprint]. arXiv. https://arxiv.org/abs/2606.28760

Chen, B., Xia, F., Ichter, B., Rao, K., Gopalakrishnan, K., Ryoo, M. S., Stone, A., & Kappler, D. (2023). Open-vocabulary queryable scene representations for real world planning. *Proceedings of the IEEE International Conference on Robotics and Automation*, 11509–11522. https://doi.org/10.1109/icra48891.2023.10161534

Chen, Y., Arkin, J., Zhang, Y., Roy, N., & Fan, C. (2024). Scalable multi-robot collaboration with large language models: Centralized or decentralized systems? *Proceedings of the IEEE International Conference on Robotics and Automation*, 4311–4317. https://doi.org/10.1109/icra57147.2024.10610676

Choe, D. B., Sangeetha, S. V., Coogan, S., & Kousik, S. (2026). *Make your VLA more robust without more data by interleaving motion planning* [Preprint]. arXiv. https://arxiv.org/abs/2606.00985

Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). QLoRA: Efficient finetuning of quantized LLMs. *Advances in Neural Information Processing Systems*, *36*, 10088–10115. https://doi.org/10.52202/075280-0441

Din, M. U., Akram, W., Saoud, L. S., Rosell, J., & Hussain, I. (2025). *Vision language action models in robotic manipulation: A systematic review* [Preprint]. arXiv. https://arxiv.org/abs/2507.10672

Dixon, S. R., Wickens, C. D., & McCarley, J. S. (2007). On the independence of compliance and reliance: Are automation false alarms worse than misses? *Human Factors*, *49*(4), 564–572. https://doi.org/10.1518/001872007x215656

Dragan, A. D., Lee, K. C. T., & Srinivasa, S. S. (2013). Legibility and predictability of robot motion. *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, 301–308. https://doi.org/10.1109/hri.2013.6483603

Dragan, A. D., & Srinivasa, S. S. (2013). A policy-blending formalism for shared control. *International Journal of Robotics Research*, *32*(7), 790–805. https://doi.org/10.1177/0278364913490324

Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., … Florence, P. (2023). PaLM-E: An embodied multimodal language model. *Proceedings of Machine Learning Research*, *202*, 8469–8488.

Elhafsi, A., Sinha, R., Agia, C., Schmerling, E., Nesnas, I. A. D., & Pavone, M. (2023). Semantic anomaly detection with large language models. *Autonomous Robots*, *47*(8), 1035–1055. https://doi.org/10.1007/s10514-023-10132-6

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. *Human Factors*, *37*(1), 32–64. https://doi.org/10.1518/001872095779049543

Fikes, R. E., & Nilsson, N. J. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, *2*(3–4), 189–208. https://doi.org/10.1016/0004-3702(71)90010-5

Firoozi, R., Tucker, J., Tian, S., Majumdar, A., Sun, J., Liu, W., Zhu, Y., Song, S., Kapoor, A., Hausman, K., Ichter, B., Driess, D., Wu, J., Lu, C., & Schwager, M. (2023). *Foundation models in robotics: Applications, challenges, and the future* [Preprint]. arXiv. https://arxiv.org/abs/2312.07843

Fox, M., & Long, D. (2003). PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, *20*, 61–124. https://doi.org/10.1613/jair.1129

Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *Proceedings of Machine Learning Research*, *48*, 1050–1059. https://proceedings.mlr.press/v48/gal16.html

Garrett, C. R., Chitnis, R., Holladay, R., Kim, B., Silver, T., Kaelbling, L. P., & Lozano-Pérez, T. (2021). Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, *4*, 265–293. https://doi.org/10.1146/annurev-control-091420-084139

Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K. M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., Gan, C., de Melo, C. M., Tenenbaum, J. B., Torralba, A., Shkurti, F., & Paull, L. (2024). ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. *Proceedings of the IEEE International Conference on Robotics and Automation*, 5021–5028. https://doi.org/10.1109/icra57147.2024.10610243

Guan, L., Sreedharan, S., Valmeekam, K., & Kambhampati, S. (2023). Leveraging pre-trained large language models to construct and utilize world models for model-based task planning. *Advances in Neural Information Processing Systems*, *36*, 79081–79094. https://doi.org/10.52202/075280-3459

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. *Proceedings of Machine Learning Research*, *70*, 1321–1330. https://proceedings.mlr.press/v70/guo17a.html

Guo, Y., Wang, Y. J., Zha, L., & Chen, J. (2024). DoReMi: Grounding language model by detecting and recovering from plan-execution misalignment. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 12124–12131. https://doi.org/10.1109/iros58592.2024.10802284

Haddadin, S., Albu-Schäffer, A., & Hirzinger, G. (2009). Requirements for safe robots: Measurements, analysis and new insights. *International Journal of Robotics Research*, *28*(11–12), 1507–1527. https://doi.org/10.1177/0278364909343970

Haddadin, S., & Croft, E. (2016). Physical human–robot interaction. In B. Siciliano & O. Khatib (Eds.), *Springer handbook of robotics* (2nd ed., pp. 1835–1874). Springer. https://doi.org/10.1007/978-3-319-32552-1_69

Hancock, P. A., Billings, D. R., Schaefer, K. E., Chen, J. Y. C., de Visser, E. J., & Parasuraman, R. (2011). A meta-analysis of factors affecting trust in human-robot interaction. *Human Factors*, *53*(5), 517–527. https://doi.org/10.1177/0018720811417254

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena*, *42*(1–3), 335–346. https://doi.org/10.1016/0167-2789(90)90087-6

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in Psychology*, *52*, 139–183. https://doi.org/10.1016/s0166-4115(08)62386-9

Helmert, M. (2006). The fast downward planning system. *Journal of Artificial Intelligence Research*, *26*, 191–246. https://doi.org/10.1613/jair.1705

Höfer, S., Bekris, K., Handa, A., Gamboa, J. C., Mozifian, M., Golemo, F., Atkeson, C., Fox, D., Goldberg, K., Leonard, J., Liu, C. K., Peters, J., Song, S., Welinder, P., & White, M. (2021). Sim2Real in robotics and automation: Applications and challenges. *IEEE Transactions on Automation Science and Engineering*, *18*(2), 398–400. https://doi.org/10.1109/tase.2021.3064065

Hoff, K. A., & Bashir, M. (2015). Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors*, *57*(3), 407–434. https://doi.org/10.1177/0018720814547570

Hoffman, G. (2019). Evaluating fluency in human-robot collaboration. *IEEE Transactions on Human-Machine Systems*, *49*(3), 209–218. https://doi.org/10.1109/thms.2019.2904558

Hoffmann, J., & Nebel, B. (2001). The FF planning system: Fast plan generation through heuristic search. *Journal of Artificial Intelligence Research*, *14*, 253–302. https://doi.org/10.1613/jair.855

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR 2022)*.

Huang, C., Mees, O., Zeng, A., & Burgard, W. (2023). Visual language maps for robot navigation. *Proceedings of the IEEE International Conference on Robotics and Automation*, 10608–10615. https://doi.org/10.1109/icra48891.2023.10160969

Huang, C.-P., Man, Y., Yu, Z., Chen, M.-H., Kautz, J., Wang, Y.-C. F., & Yang, F.-E. (2026). *Fast-ThinkAct: Efficient vision-language-action reasoning via verbalizable latent planning*. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). https://arxiv.org/abs/2601.09708

Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X., & Zhou, D. (2024). Large language models cannot self-correct reasoning yet. *International Conference on Learning Representations (ICLR 2024)*. https://openreview.net/forum?id=IkmD3fKBPQ

Huang, W., Abbeel, P., Pathak, D., & Mordatch, I. (2022). Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. *Proceedings of Machine Learning Research*, *162*, 9118–9147. https://proceedings.mlr.press/v162/huang22a.html

Huang, W., Wang, C., Zhang, R., Li, Y., Wu, J., & Fei-Fei, L. (2023). VoxPoser: Composable 3D value maps for robotic manipulation with language models. *Proceedings of Machine Learning Research*, *229*. https://arxiv.org/abs/2307.05973

Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., Sermanet, P., Brown, N., Jackson, T., Luu, L., Levine, S., Hausman, K., & Ichter, B. (2022). Inner monologue: Embodied reasoning through planning with language models. *Proceedings of Machine Learning Research*, *205*, 1769–1782. https://arxiv.org/abs/2207.05608

James, S., Ma, Z., Arrojo, D. R., & Davison, A. J. (2020). RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, *5*(2), 3019–3026. https://doi.org/10.1109/lra.2020.2974707

Jatavallabhula, K. M., Kuwajerwala, A., Gu, Q., Omama, M., Chen, T., Maalouf, A., Li, S., Iyer, G., Saryazdi, S., Keetha, N., Tewari, A., Tenenbaum, J. B., de Melo, C. M., Krishna, M., Paull, L., Shkurti, F., & Torralba, A. (2023). ConceptFusion: Open-set multimodal 3D mapping. *Robotics: Science and Systems*. https://doi.org/10.15607/rss.2023.xix.066

Javdani, S., Admoni, H., Pellegrinelli, S., Srinivasa, S. S., & Bagnell, J. A. (2018). Shared autonomy via hindsight optimization for teleoperation and teaming. *International Journal of Robotics Research*, *37*(7), 717–742. https://doi.org/10.1177/0278364918776060

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys*, *55*(12). https://doi.org/10.1145/3571730

Jian, J.-Y., Bisantz, A. M., & Drury, C. G. (2000). Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics*, *4*(1), 53–71. https://doi.org/10.1207/s15327566ijce0401_04

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., … Kaplan, J. (2022). *Language models (mostly) know what they know* [Preprint]. arXiv. https://arxiv.org/abs/2207.05221

Kaelbling, L. P., & Lozano-Pérez, T. (2011). Hierarchical task and motion planning in the now. *Proceedings of the IEEE International Conference on Robotics and Automation*, 1470–1477. https://doi.org/10.1109/icra.2011.5980391

Kambhampati, S., Valmeekam, K., Guan, L., Verma, M., Stechly, K., Bhambri, S., Saldyt, L., & Murthy, A. (2024). *LLMs can't plan, but can help planning in LLM-Modulo frameworks* [Preprint]. arXiv. https://arxiv.org/abs/2402.01817

Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, *30*, 5575–5585.

Khalastchi, E., & Kalech, M. (2019). On fault detection and diagnosis in robotic systems. *ACM Computing Surveys*, *51*(1). https://doi.org/10.1145/3146389

Kim, D., Park, D., Lee, S., Kim, J., Oh, Y., Shin, J., & Yoon, S. (2026). *Safe embodied AI for long-horizon tasks: A cross-layer analysis of robotic manipulation* [Preprint]. arXiv. https://arxiv.org/abs/2606.05660

Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., & Finn, C. (2024). OpenVLA: An open-source vision-language-action model. *Proceedings of Machine Learning Research*, *270*, 2679–2713. https://arxiv.org/abs/2406.09246

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W. Y., Dollár, P., & Girshick, R. (2023). Segment anything. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3992–4003. https://doi.org/10.1109/iccv51070.2023.00371

Koppula, H. S., & Saxena, A. (2016). Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(1), 14–29. https://doi.org/10.1109/tpami.2015.2430335

Kress-Gazit, H., Fainekos, G. E., & Pappas, G. J. (2009). Temporal-logic-based reactive mission and motion planning. *IEEE Transactions on Robotics*, *25*(6), 1370–1381. https://doi.org/10.1109/tro.2009.2030225

Kress-Gazit, H., Lahijanian, M., & Raman, V. (2018). Synthesis for robots: Guarantees and feedback for robot behavior. *Annual Review of Control, Robotics, and Autonomous Systems*, *1*, 211–236. https://doi.org/10.1146/annurev-control-060117-104838

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, *30*.

Lasota, P. A., Fong, T., & Shah, J. A. (2017). A survey of methods for safe human-robot interaction. *Foundations and Trends in Robotics*, *5*(4), 261–349. https://doi.org/10.1561/2300000052

Lee, J. D., & See, K. A. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors*, *46*(1), 50–80. https://doi.org/10.1518/hfes.46.1.50_30392

Lee, J., Shin, H., Kim, H., Zhang, P., & Ko, J. (2026). *On-device robotic planning: Eliminating inference redundancy for efficient decision-making* [Preprint]. arXiv. https://arxiv.org/abs/2605.31460

Leviathan, Y., Kalman, M., & Matias, Y. (2023). Fast inference from transformers via speculative decoding. *Proceedings of Machine Learning Research*, *202*, 19274–19286. https://proceedings.mlr.press/v202/leviathan23a.html

Li, B., Qi, P., Liu, B., Di, S., Liu, J., Pei, J., Yi, J., & Zhou, B. (2023). Trustworthy AI: From principles to practices. *ACM Computing Surveys*, *55*(9). https://doi.org/10.1145/3555803

Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martín-Martín, R., Wang, C., Levine, G., Lingelbach, M., Sun, J., Anvari, M., Hwang, M., Sharma, M., Aydin, A., Bansal, D., Hunter, S., Kim, K.-Y., Lou, A., Matthews, C. R., … Fei-Fei, L. (2023). BEHAVIOR-1K: A benchmark for embodied AI with 1,000 everyday activities and

realistic simulation. *Proceedings of Machine Learning Research*, *205*, 80–93. https://proceedings.mlr.press/v205/li23a.html

Li, J., Li, D., Savarese, S., & Hoi, S. (2023). BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Proceedings of Machine Learning Research*, *202*, 19730–19742. https://proceedings.mlr.press/v202/li23q.html

Li, P., An, Z., Abrar, S., & Zhou, L. (2026). Large language models for multi-robot systems: A survey. *Autonomous Robots*, *50*(3), 30. https://doi.org/10.1007/s10514-026-10257-4

Li, T., Zhang, J., Wang, S., Huang, X., & Wei, Z. (2026). *LIBERO-Occ: Evaluating and improving vision-language-action models under scene-induced occlusion via viewpoint imagination* [Preprint]. arXiv. https://arxiv.org/abs/2606.10862

Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., & Zeng, A. (2023). Code as policies: Language model programs for embodied control. *Proceedings of the IEEE International Conference on Robotics and Automation*, 9493–9500. https://doi.org/10.1109/icra48891.2023.10160591

Lin, Z., Trivedi, S., & Sun, J. (2023). Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*. https://arxiv.org/abs/2305.19187

Liu, B., Jiang, Y., Zhang, X., Liu, Q., Zhang, S., Biswas, J., & Stone, P. (2023). *LLM+P: Empowering large language models with optimal planning proficiency* [Preprint]. arXiv. https://arxiv.org/abs/2304.11477

Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual instruction tuning. *Advances in Neural Information Processing Systems*, *36*. https://doi.org/10.52202/075280-1516

Liu, H., Zhou, Y., Wu, Z., Ji, Z., Shan, Z., Wang, Q., Liu, R., Yang, Z., Gu, Y., Khan, S., Yan, S., Liu, J., Zhu, H., Liu, C., Yang, J., Zhang, J., & Wang, Z. (2026). *RoCo challenge at AAAI 2026: Benchmarking robotic collaborative manipulation for assembly towards industrial automation* [Preprint]. arXiv. https://arxiv.org/abs/2603.15469

Liu, J., Li, P., Chen, S., Wang, T., & Zhao, Y.-B. (2026). *Legible shared autonomy: Implicit communication of robot belief through motion* [Preprint]. arXiv. https://arxiv.org/abs/2606.29846

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., & Zhang, L. (2025). Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *Lecture Notes in Computer Science*, *15105*, 38–55. https://doi.org/10.1007/978-3-031-72970-6_3

Liu, Z., Bahety, A., & Song, S. (2023). REFLECT: Summarizing robot experiences for failure explanation and correction. *Proceedings of Machine Learning Research*, *229*. https://arxiv.org/abs/2306.15724

Losey, D. P., McDonald, C. G., Battaglia, E., & O'Malley, M. K. (2018). A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction. *Applied Mechanics Reviews*, *70*(1). https://doi.org/10.1115/1.4039145

Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., & Anandkumar, A. (2024). Eureka: Human-level reward design via coding large language models. *International Conference on Learning Representations (ICLR 2024)*, 26516–26560.

Mainprice, J., & Berenson, D. (2013). Human-robot collaborative manipulation planning using early prediction of human motion. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 299–306. https://doi.org/10.1109/iros.2013.6696368

Mandi, Z., Jain, S., & Song, S. (2024). RoCo: Dialectic multi-robot collaboration with large language models. *Proceedings of the IEEE International Conference on Robotics and Automation*, 286–299. https://doi.org/10.1109/icra57147.2024.10610855

Matuszek, C., Herbst, E., Zettlemoyer, L., & Fox, D. (2013). Learning to parse natural language commands to a robot control system. *Experimental Robotics*, 403–415. https://doi.org/10.1007/978-3-319-00065-7_28

Mees, O., Hermann, L., Rosete-Beas, E., & Burgard, W. (2022). CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, *7*(3), 7327–7334. https://doi.org/10.1109/lra.2022.3180108

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., Mahendran, A., Arnab, A., Dehghani, M., Shen, Z., Wang, X., Zhai, X., Kipf, T., & Houlsby, N. (2022). Simple open-vocabulary object detection. *Lecture Notes in Computer Science*, *13670*, 728–755. https://doi.org/10.1007/978-3-031-20080-9_42

Muir, B. M. (1994). Trust in automation: Part I. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics*, *37*(11), 1905–1922. https://doi.org/10.1080/00140139408964957

Nikolaidis, S., Hsu, D., & Srinivasa, S. (2017). Human-robot mutual adaptation in collaborative tasks: Models and experiments. *International Journal of Robotics Research*, *36*(5–7), 618–634. https://doi.org/10.1177/0278364917690593

Octo Model Team, Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Kreiman, T., Xu, C., Luo, J., Tan, Y. L., Chen, L. Y., Sanketi, P., Vuong, Q., Xiao, T., Sadigh, D., Finn, C., & Levine, S. (2024). Octo: An open-source generalist robot policy. *Robotics: Science and Systems*. https://doi.org/10.15607/rss.2024.xx.090

O'Neill, A., Rehman, A., Maddukuri, A., Gupta, A., Padalkar, A., Lee, A., Pooley, A., Gupta, A., Mandlekar, A., Jain, A., Tung, A., Bewley, A., Herzog, A., Irpan, A., Khazatsky, A., Rai, A., Gupta, A., Wang, A., Singh, A., … Lin, Z. (2024). Open X-Embodiment: Robotic learning datasets and RT-X models. *Proceedings of the IEEE*

International Conference on Robotics and Automation, 6892–6903. https://doi.org/10.1109/icra57147.2024.10611477

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J. V., Lakshminarayanan, B., & Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, *32*.

Parasuraman, R., & Riley, V. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, *39*(2), 230–253. https://doi.org/10.1518/001872097778543886

Puig, X., Shu, T., Li, S., Wang, Z., Liao, Y. H., Tenenbaum, J. B., Fidler, S., & Torralba, A. (2020). Watch-and-help: A challenge for social perception and Human-AI collaboration. *International Conference on Learning Representations (ICLR 2021)*. https://arxiv.org/abs/2010.09890

Puig, X., Undersander, E., Szot, A., Dallaire-Côté, M., Yang, T.-Y., Partsey, R., Desai, R., Clegg, A., Hlavac, M., Min, S. Y., Vondruš, V., Gervet, T., Berges, V.-P., Turner, J., Maksymets, O., Kira, Z., Kalakrishnan, M., Malik, J., Chaplot, D., … Mottaghi, R. (2024). Habitat 3.0: A co-habitat for humans, avatars, and robots. *International Conference on Learning Representations (ICLR 2024)*, 15306–15336.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *Proceedings of Machine Learning Research*, *139*, 8748–8763. https://proceedings.mlr.press/v139/radford21a.html

Raman, S. S., Cohen, V., Idrees, I., Rosen, E., Mooney, R., Tellex, S., & Paulius, D. (2024). CAPE: Corrective actions from precondition errors using large language models. *Proceedings of the IEEE International Conference on Robotics and Automation*, 14070–14077. https://doi.org/10.1109/icra57147.2024.10611376

Rana, K., Haviland, J., Garg, S., Abou-Chakra, J., Reid, I., & Sünderhauf, N. (2023). SayPlan: Grounding large language models using 3D scene graphs for scalable robot task planning. *Proceedings of Machine Learning Research*, *229*. https://arxiv.org/abs/2307.06135

Ren, A. Z., Dixit, A., Bodrova, A., Singh, S., Tu, S., Brown, N., Xu, P., Takayama, L., Xia, F., Varley, J., Xu, Z., Sadigh, D., Zeng, A., & Majumdar, A. (2023). Robots that ask for help: Uncertainty alignment for large language model planners. *Proceedings of Machine Learning Research*, *229*. https://arxiv.org/abs/2307.01928

Ross, S., Gordon, G. J., & Bagnell, J. A. (2011). A reduction of imitation learning and structured prediction to no-regret online learning. *Proceedings of Machine Learning Research*, *15*, 627–635. https://proceedings.mlr.press/v15/ross11a.html

Sakai, T., & Nagai, T. (2022). Explainable autonomous robots: A survey and perspective. *Advanced Robotics*, *36*(5–6), 219–238. https://doi.org/10.1080/01691864.2022.2029720

Schaefer, K. E. (2016). Measuring trust in human robot interactions: Development of the "trust perception scale-HRI." In R. Mittu, D. Sofge, A. Wagner, & W. F. Lawless (Eds.), *Robust intelligence and trust in autonomous systems* (pp. 191–218). Springer. https://doi.org/10.1007/978-1-4899-7668-0_10

Selvaggio, M., Cognetti, M., Nikolaidis, S., Ivaldi, S., & Siciliano, B. (2021). Autonomy in physical human-robot interaction: A brief survey. *IEEE Robotics and Automation Letters*, *6*(4), 7989–7996. https://doi.org/10.1109/lra.2021.3100603

Shafiullah, N. M. M., Paxton, C., Pinto, L., Chintala, S., & Szlam, A. (2023). CLIP-Fields: Weakly supervised semantic fields for robotic memory. *Robotics: Science and Systems*. https://doi.org/10.15607/rss.2023.xix.074

Shridhar, M., Manuelli, L., & Fox, D. (2022). CLIPort: What and where pathways for robotic manipulation. *Proceedings of Machine Learning Research*, *164*, 894–906. https://proceedings.mlr.press/v164/shridhar22a.html

Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., & Fox, D. (2020). ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10737–10746. https://doi.org/10.1109/cvpr42600.2020.01075

Shridhar, M., Yuan, X., Côté, M. A., Bisk, Y., Trischler, A., & Hausknecht, M. (2020). ALFWorld: Aligning text and embodied environments for interactive learning. *International Conference on Learning Representations (ICLR 2021)*. https://arxiv.org/abs/2010.03768

Silver, T., Dan, S., Srinivas, K., Tenenbaum, J. B., Kaelbling, L., & Katz, M. (2024). Generalized planning in PDDL domains with pretrained large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, *38*(18), 20256–20264. https://doi.org/10.1609/aaai.v38i18.30006

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., & Garg, A. (2022). *ProgPrompt: Generating situated robot task plans using large language models* [Preprint]. arXiv. https://arxiv.org/abs/2209.11302

Singh, I., Blukis, V., Mousavian, A., Goyal, A., Xu, D., Tremblay, J., Fox, D., Thomason, J., & Garg, A. (2023). ProgPrompt: Program generation for situated robot task planning using large language models. *Autonomous Robots*, *47*(8), 999–1012. https://doi.org/10.1007/s10514-023-10135-3

Sisbot, E. A., Marin-Urias, L. F., Alami, R., & Siméon, T. (2007). A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, *23*(5), 874–883. https://doi.org/10.1109/tro.2007.904911

Skreta, M., Zhou, Z., Yuan, J. L., Darvish, K., Aspuru-Guzik, A., & Garg, A. (2024). *RePLan: Robotic replanning with perception and language models* [Preprint]. arXiv. https://arxiv.org/abs/2401.04157

Song, C. H., Sadler, B. M., Wu, J., Chao, W. L., Washington, C., & Su, Y. (2023). LLM-Planner: Few-shot grounded planning for embodied agents with large language models. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2986–2997. https://doi.org/10.1109/iccv51070.2023.00280

Srivastava, S., Fang, E., Riano, L., Chitnis, R., Russell, S., & Abbeel, P. (2014). Combined task and motion planning through an extensible planner-independent interface layer. *Proceedings of the IEEE International Conference on Robotics and Automation*, 639–646. https://doi.org/10.1109/icra.2014.6906922

Srivastava, S., Li, C., Lingelbach, M., Martín-Martín, R., Xia, F., Vainio, K. E., Lian, Z., Gokmen, C., Buch, S., Liu, K., Savarese, S., Gweon, H., Wu, J., & Fei-Fei, L. (2022). BEHAVIOR: Benchmark for everyday household activities in virtual, interactive, and ecological environments. *Proceedings of Machine Learning Research*, *164*, 477–490. https://proceedings.mlr.press/v164/srivastava22a.html

Stone, A., Xiao, T., Lu, Y., Gopalakrishnan, K., Lee, K. H., Vuong, Q., Wohlhart, P., Kirmani, S., Zitkovich, B., Xia, F., Finn, C., & Hausman, K. (2023). Open-world object manipulation using pre-trained vision-language models. *Proceedings of Machine Learning Research*, *229*. https://arxiv.org/abs/2303.00905

Sun, N., Zhang, Y., Yang, Y., Zhao, W., Li, P., Guo, J., Song, W., Ding, P., Suo, R., Su, Y., Xiao, X., Li, X., & Liu, H. (2026). *Revisiting embodied chain-of-thought for generalizable robot manipulation* [Preprint]. arXiv. https://arxiv.org/abs/2606.03784

Tellex, S., Gopalan, N., Kress-Gazit, H., & Matuszek, C. (2020). Robots that use language. *Annual Review of Control, Robotics, and Autonomous Systems*, *3*, 25–55. https://doi.org/10.1146/annurev-control-101119-071628

Tellex, S., Kollar, T., Dickerson, S., Walter, M. R., Banerjee, A. G., Teller, S., & Roy, N. (2011). Understanding natural language commands for robotic navigation and mobile manipulation. *Proceedings of the AAAI Conference on Artificial Intelligence*, *25*(1), 1507–1514. https://doi.org/10.1609/aaai.v25i1.7979

Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 23–30. https://doi.org/10.1109/iros.2017.8202133

Valmeekam, K., Marquez, M., Sreedharan, S., & Kambhampati, S. (2023). On the planning abilities of large language models: A critical investigation. *Advances in Neural Information Processing Systems*, *36*. https://doi.org/10.52202/075280-3320

Valmeekam, K., Olmo, A., Marquez, M., Sreedharan, S., & Kambhampati, S. (2023). PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, *36*, 38975–38987. https://doi.org/10.52202/075280-1693

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Villani, V., Pini, F., Leali, F., & Secchi, C. (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics*, *55*, 248–266. https://doi.org/10.1016/j.mechatronics.2018.02.009

Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic learning in a random world*. Springer.

Wang, F., Wang, Z., Pei, G., Zhang, M., Liang, C., Hu, J., Li, Z., Wu, J., Han, N., Zhang, Z., Qi, J., Wu, H., Zhang, S., Zheng, P., Pan, J., Navarro-Alarcon, D., Liu, S., & Zhou, P. (2026). *World models for robotic manipulation: A survey* [Preprint]. arXiv. https://arxiv.org/abs/2606.00113

Wang, J., Xu, X., & Huang, X. (2026). *Probing collision grounding in vision-language models for safe human-robot collaboration* [Preprint]. arXiv. https://arxiv.org/abs/2605.31196

Wang, Y., Zheng, S., Luo, H., Zhang, W., Yuan, H., Xu, C., Xu, H., Feng, Y., Yu, M., Kang, Z., Lu, Z., & Jin, Q. (2026). *Rethinking visual-language-action model scaling: Alignment, mixture, and regularization* [Preprint]. arXiv. https://arxiv.org/abs/2602.09722

Wang, Z., Wang, B., Zhang, H., Du, T., Chen, T., Sun, G., He, Y., Shen, Z., Ye, W., & Li, A. (2026). *Vision-language-action in robotics: A survey of datasets, benchmarks, and data engines* [Preprint]. arXiv. https://arxiv.org/abs/2604.23001

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, *35*. https://doi.org/10.52202/068431-1800

Wickens, C. D., & Dixon, S. R. (2007). The benefits of imperfect diagnostic automation: A synthesis of the literature. *Theoretical Issues in Ergonomics Science*, *8*(3), 201–212. https://doi.org/10.1080/14639220500370105

Wu, F., Zheng, X., Qu, Y., Wang, Z., Feng, Z., & Li, H. (2026). *Grounding generative planners in verifiable logic: A hybrid architecture for trustworthy embodied AI* [Preprint]. arXiv. https://arxiv.org/abs/2602.08373

Wu, J., Antonova, R., Kan, A., Lepert, M., Zeng, A., Song, S., Bohg, J., Rusinkiewicz, S., & Funkhouser, T. (2023). TidyBot: Personalized robot assistance with large language models. *Autonomous Robots*, *47*(8), 1087–1102. https://doi.org/10.1007/s10514-023-10139-z

Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., & Hooi, B. (2024). Can LLMs express their uncertainty? An empirical evaluation of confidence elicitation in LLMs. *International Conference on Learning Representations (ICLR 2024)*, 23650–23678.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). ReAct: Synergizing reasoning and acting in language models. *International Conference on Learning Representations (ICLR 2023)*. https://arxiv.org/abs/2210.03629

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., & Levine, S. (2020). Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. *Proceedings of Machine Learning Research*, *100*, 1094–1100. https://proceedings.mlr.press/v100/yu20a.html

Yu, W., Gileadi, N., Fu, C., Kirmani, S., Lee, K. H., Arenas, M. G., Chiang, H. T. L., Erez, T., Hasenclever, L., Humplik, J., Ichter, B., Xiao, T., Xu, P., Zeng, A., Zhang, T., Heess, N., Sadigh, D., Tan, J., Tassa, Y., & Xia, F. (2023). Language to rewards for robotic skill synthesis. *Proceedings of Machine Learning Research*, *229*. https://arxiv.org/abs/2306.08647

Yu, Y., & Qiu, X. (2026). *Benchmarking vision-language-action models on SO-101: Failure and recovery analysis*. arXiv:2606.08881 [Preprint]. arXiv. https://arxiv.org/abs/2606.08881

Zacharaki, A., Kostavelis, I., Gasteratos, A., & Dokas, I. (2020). Safety bounds in human robot interaction: A survey. *Safety Science*, *127*, 104667. https://doi.org/10.1016/j.ssci.2020.104667

Zeng, A., Attarian, M., Ichter, B., Choromanski, K., Wong, A., Welker, S., Tombari, F., Purohit, A., Ryoo, M. S., Sindhwani, V., Lee, J., Vanhoucke, V., & Florence, P. (2022). Socratic models: Composing zero-shot multimodal reasoning with language. *International Conference on Learning Representations (ICLR 2023)*. https://arxiv.org/abs/2204.00598

Zeng, A., Florence, P., Tompson, J., Welker, S., Chien, J., Attarian, M., Armstrong, T., Krasin, I., Duong, D., Sindhwani, V., & Lee, J. (2021). Transporter networks: Rearranging the visual world for robotic manipulation. *Proceedings of Machine Learning Research*, *155*, 726–747. https://proceedings.mlr.press/v155/zeng21a.html

Zhang, H., Du, W., Shan, J., Zhou, Q., Du, Y., Tenenbaum, J. B., Shu, T., & Gan, C. (2024). Building cooperative embodied agents modularly with large language models. *International Conference on Learning Representations (ICLR 2024)*, 19373–19401.

Zhang, K., Zhang, H., Xu, Z., Zhang, Z., Prince, M. R. I., Li, X., Han, X., Zhou, Y., Ajoudani, A., & She, Y. (2026). *TacVLA: Contact-aware tactile fusion for robust vision-language-action manipulation* [Preprint]. arXiv. https://arxiv.org/abs/2603.12665

Zhao, W., Queralta, J. P., & Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: A survey. *Proceedings of the IEEE Symposium Series on Computational Intelligence*, 737–744. https://doi.org/10.1109/ssci47803.2020.9308468

Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., & Misra, I. (2022). Detecting twenty-thousand classes using image-level supervision. *Lecture Notes in Computer Science*, *13669*, 350–368. https://doi.org/10.1007/978-3-031-20077-9_21