Proceedings of CONF-MLA 2026 Symposium: Intelligent Systems and Automation: AI Models, IoT, and Robotic Algorithms
DOI: 10.54254/2755-2721/2026.LD39094

Warehouse Mobile Manipulation with Large Language Models and Foundation Models

Zhaoqi Chen

College of Mechanical Engineering, Zhejiang University of Technology, Hangzhou, China
qa19817755167@gmail.com

Abstract. Mobile manipulators can move between storage locations and handle goods, but transferring the semantic capabilities of foundation models to dependable warehouse operations remains constrained by execution interfaces, data coverage, and operating conditions. This thematic critical review examines perception, reasoning, planning and execution, and data and generalization. It compares hierarchical approaches with vision-language-action models and assesses their relevance to warehouse tasks using representative studies. Its distinctive value lies in relating model interfaces and transfer limits to warehouse order fulfillment, where stock-keeping unit (SKU) verification, shelf access, and onboard resource limits must be considered together. The evidence indicates that foundation models reorganize selected interfaces between observations, task descriptions, and actions; they do not remove the need for geometric feasibility checks, low-level control, or failure recovery. Open-vocabulary recognition must be connected to item-level verification, while semantic generalization should be distinguished from the acquisition of new motor skills. Reported transfer gains also depend on the task, embodiment, and adaptation budget. The review identifies order completion, coordinated base-arm motion, and measurable adaptation costs as priorities for research and evaluation. These conclusions support method selection and experimental design rather than a claim of validated warehouse deployment.

Keywords: mobile manipulation, large language models, foundation models, vision-language-action models, warehouse logistics, embodied intelligence

1. Introduction

A mobile manipulator is a mixture of an autonomous mobile platform and a robotic arm, which broadens object manipulation over storage areas and brings about coupled localization, manipulation and task coordination issues [1]. It is only a part of tasks to achieve the desired area with changing inventory and layouts. The robot should be able to determine the right item, pick a workable posture of the robot, and confirm that the given order was done.

There are two general approaches that can be used to meet these requirements. The modular systems combine localization and mapping, path planning and grasp control to enhance the execution [1, 2]. Foundation-model-based methods rely on pretrained representations to associate semantic descriptions with actions. Pretraining could lessen the amount of work needed to sustain

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.LD39094

these relationships as tasks evolve though this saving is dependent upon the interfaces and functionalities kept in the operational system.

CLIP matches images and text [3]; SayCan integrates language-related relevance with skill affordances [4]. PaLM-E produces textual responses or plans based on multimodal inputs [5], whereas RT-2 simultaneously acquires vision-language task and robot action outputs [6]. The methods are implemented by means of various interfaces and are not intended to be considered as one architecture.

The given review questions which information interface a pretrained knowledge alteration occurs and in which case these alterations could be advantageous to the warehouse operation. The hierarchical planning and vision-language-action (VLA) models are considered with the interactions between the data requirements, generalization, and the reliability of the operations.

The current reviews cover popular technical issues in embodied mobile manipulation [2], the progress in industrial systems [1] and theoretical frameworks of mobile service robots [7]. In this case, the model interfaces and transfer restrictions are tested on how well they meet the warehouse specifications of verifying items at the item level, synchronising motions of the base and arm, and working in a continuous way with limited onboard resources. This is a synthesis that focuses on warehouses, not a new generalized taxonomy of robotic manipulation.

2. System architecture and warehouse constraints

2.1. Conventional functional layers

Mobile manipulation can be organized into autonomous navigation, object manipulation, and task coordination [1, 2]. These functions address localization and route planning, object perception and manipulation control, and high-level goal decomposition, respectively. In practice, navigation and manipulation often share state estimates, and execution outcomes feed back into planning. The resulting system is therefore not simply a one-way pipeline.

An interface becomes limiting when its representation omits information needed by the next decision. The object category and position itself might not be enough to represent the package distortion or access limitation in case of a later grasp. Semantic representations in foundation models could add richness to semantic representations, but object relations, geometry, and control interfaces are still subject to explicit consideration [2, 5]. There is a transfer of information and tasks in the system; the execution constraints remain the same.

2.2. Constraints specific to warehouse tasks

The warehouse operation entails perception, motion and safety restrictions that have to be considered in the same order of tasks.

Products are different in packaging, size, and materials and visually similar products could be different stock-keeping units (SKUs). Clutter on shelves may obscure the intended items. Reflective or deformable packaging makes it even more difficult to sense and grasp [21]. Open vocabulary representations permit flexible descriptions of targets [3] but the semantic correspondence needs to be related to the object required by the order. Reading barcodes, text recognition and instance matching are still pertinent complementary mechanisms.

The shelves and the narrow aisles limit both the base and the arm. The stopping pose should permit reaching the target without causing any collisions [2]. Simplifying the problem by navigating first and then grasping after stopping may help. It is not applicable to the cases when the task

DOI: 10.54254/2755-2721/2026.LD39094

requires moving of the base in the process of manipulation, e.g. coordinated door opening or movement in confined space.

The load operation, human activity and unpredictable interruptions increase runtime uncertainty. The response time should be based on the hazard and not on the comfort of a model inference cycle. Both uncertainty and real-time calculation are known issues with foundation-model-based robots [7]. These limits are also augmented by limited on-board memory and power [7], and in warehousing, inferences need to be compatible to the timing of picking and base arm coordination. Control systems that are not based completely on generative predictions are needed to implement emergency stopping and speed limitations.

The transfer claims should detail the situation of the task. It is not enough to state that an environment is either a home, or a warehouse in order to determine its level of difficulty.

2.3. Review scope and analytical approach

The paper is a thematic critical essay on sample Chinese and English language research whose full texts could be accessed. The evidence has been structured into inputs and outputs, execution interfaces, evaluation setting, data requirements, and failures that have been reported. The methodological background includes fixed-base manipulation and general vision studies, but does not include direct warehouse validation. The analytical questions in Table 1 are defined in four technical dimensions. The numerical outcomes are mainly interpreted in their original researches. The exhaustive searching, independent dual coding, and meta-analysis were not implemented, and the article does not state the coverage or reproducibility of a systematic review.

Table 1. Technical dimensions and analytical questions

Dimension	Method description	Warehouse transfer	Critical question
Perception	Input representations and target localization	Item instances and spatial changes	Does semantic matching support grasping?
Reasoning	Skill selection and program generation	Ambiguous instructions and skill coverage	How are feasibility and risk checked?
Planning and execution	Action outputs and base-arm coordination	Confined space and sustained operation	Are feedback, recovery, and time reported?
Data and generalization	Data sources and adaptation	Changes in objects, layouts, and embodiments	Are adaptation budgets and unseen conditions specified?

Note. The entries define questions for synthesis, not performance scores. Evaluation settings are considered separately.

3. Perception from closed categories to open vocabulary

3.1. Visual foundation models

Conventional supervised recognition typically limits predictions to categories represented by training labels. CLIP instead uses contrastive learning on 400 million image-text pairs to align visual and textual representations [3]. At inference, an image can be compared with candidate text descriptions without training on labeled examples from the target dataset. Its zero-shot ImageNet performance matched that of a supervised ResNet-50, but this result does not establish reliable identification of arbitrary unseen products.

The Vision Transformer (ViT) divides an image into patches, projects them into embeddings, and supplies positional information to a Transformer [8]. The architecture provides a scalable visual

DOI: 10.54254/2755-2721/2026.LD39094

backbone; open-vocabulary recognition depends on the training objective and associated language alignment rather than on patch tokenization alone.

For a warehouse, changing candidate descriptions may reduce the need to retrain a classifier when inventory changes. It does not remove the need to validate recognition on the actual goods. Similar packaging, incomplete views, and distribution shifts can still require additional labels, adaptation, or nonsemantic identification signals. Flexible category naming and dependable SKU verification are different capabilities [3].

3.2. Integrating perception and reasoning

PaLM-E projects visual or state encodings into the language embedding space and interleaves them with textual instructions [5]. IThe system produces textual answers or high-level plans that low-level policies execute, with new observations supporting replanning. Multimodal input reduces reliance on first describing observations in words, but retains the skill interface.

Spatial information can also remain explicit. OK-Robot integrates semantic memory, navigation, grasping, and placement [10]. These systems show that pretrained representations can support modular architectures and complement geometric maps rather than necessarily replace them.

Joint perception and reasoning alter where information is encoded and which module interprets it. They do not eliminate encoding or projection: PaLM-E still transforms sensor inputs before the language model processes them [5]. Whether this arrangement preserves more task-relevant information than a conventional interface must be assessed on the target task. A shared embedding space alone is not evidence of lower information loss.

3.3. Limits of transfer to warehouses

The recognition of warehouses needs to distinguish between product specifications and individual items even when they look alike. The fine-grained classification assessments made by CLIP do not provide the correct level of SKU accuracy in a specific warehouse [3]. A non-certain choice must be distinguished by a false acceptance; using visual similarities with texts, measurements or bar codes may help to verify before the wrong products are included in the delivery process.

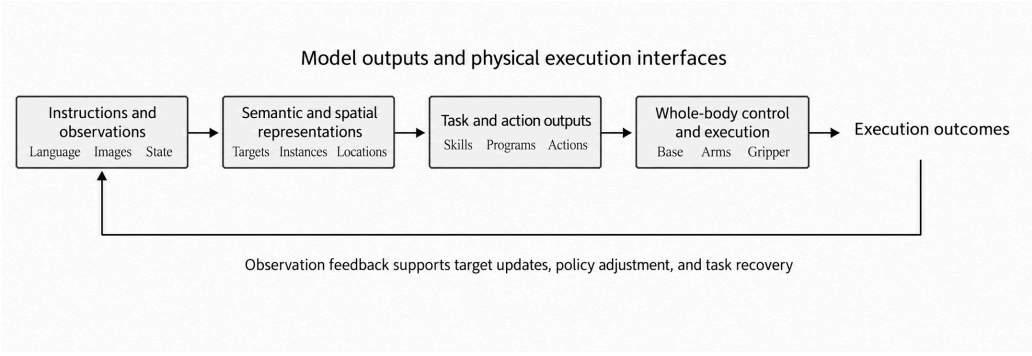


Figure 1. Information interfaces and execution feedback in warehouse mobile manipulation

Note. Conceptual synthesis of the methods discussed in the text; the figure does not depict an implemented system or validated safety architecture.

The full-task success of OK-Robot was found at 58.5% in 161 pick-and-place tasks in 10 households and failures were reported by the robot in terms of retrieval, grasping, and hardware [10]. Clutter removal can alter visual ambiguity as well as physical reachability. Post-scene

DOI: 10.54254/2755-2721/2026.LD39094

preparation improvement is not solely perceptual. The semantic queries should be linked to the latest state estimates and execution feedback. These relationships are summarized in Figure 1 on warehouse mobile manipulation.

4. Reasoning from symbolic planning to language grounding

4.1. Symbolic planning and its limitations

Task execution requires an instruction to be connected to actions that the robot can perform. Symbolic approaches, including Planning Domain Definition Language (PDDL) and hierarchical task network (HTN) planning, represent relevant entities, predicates, and action constraints explicitly. Such representations support inspectable planning in structured domains, although ambiguous requests and unmodeled events expose gaps in the domain specification. Maintaining the model can become costly as tasks change. These limitations motivate language-assisted and hybrid approaches, but do not justify dismissing symbolic planning as unusable in warehouses [2, 4].

4.2. Language grounding in SayCan

SayCan separates linguistic relevance from the robot's ability to execute a skill [4]. The language model estimates whether a candidate skill is useful for the instruction, while an affordance model evaluates the skill in the current state. A linguistically plausible next step is therefore not automatically selected as executable.

The system multiplies the language score and the skill value score, selects a skill, and proceeds sequentially [4]. Skill policies are obtained through behavioral cloning or reinforcement learning, while affordance value functions are learned through reinforcement learning. In a real-robot evaluation using 101 instructions in a mock kitchen, planning and execution success were 84% and 74%, respectively. These are distinct outcomes, not scores for two competing algorithms. Learned affordances are also bounded by their training experience and do not constitute safety guarantees.

4.3. Policy generation with Code as Policies

The Code as Policies project translates the given natural-language instructions into executable Python policy code [11]. With the provided examples of matching comment instructions with policy functions, a code language model produces programs that call perception APIs, make use of numerical or spatial libraries like NumPy and Shapely, and call parameterized control primitives. Unspecified functions may be created in a recursive fashion. The resultant program communicates the instruction in terms of software and control facilities that are already accessible to the robot.

Compared to skill scoring, generated code puts some aspects of the decision logic and its parameters under direct inspection. Object binding, preconditions, and runtime behavior is yet to be verified [4, 11]. The quantitative comparisons covered by Code as Policies are limited to simulated environments (robotics) whereas the physical demonstrations demonstrate specific functionalities. Such assessments do not determine the strength of warehouse operation in the real world [11].

4.4. Execution feedback and failure recovery

CENTAURO system integrates a language model, a behaviour library, and behaviour tree with failure detection and recovery [12]. When recovering, the reported rate of success in the physical fetch trials was raised by 88% as compared to that when no recovery was made at 80% (25 trials per

DOI: 10.54254/2755-2721/2026.LD39094

condition). The execution times also grew to 121.3-167.1 s. This small comparison demonstrates the recovery-time trade-off, but the overall or statistically significant improvement has not yet been determined. The system is not capable of real-time replanning, hence recovery should be analyzed together with latency, retry limits, and human intervention..

5. Planning and execution from modular interfaces to VLA policies

5.1. Vision-language-action modeling in RT-2

RT-2 exemplifies the vision-language-action (VLA) approach [6]. Whereas RT-1 learns a policy from robot demonstrations [14], RT-2 represents robot actions as tokens and co-fine-tunes a pretrained model on robotic and vision-language data.Actions can be updated as new observations arrive, while low-level execution and safety mechanisms remain part of the surrounding system.

RT-2 demonstrates semantic generalization under changes in objects, backgrounds, and environments, with performance on familiar tasks broadly comparable to RT-1 [6]. Its chain-of-thought variant adds a Plan field to the training examples and applies several hundred gradient updates to the PaLM-E-based configuration. The reported behavior therefore involves additional training rather than prompting alone.

The joint training may recycle one of the existing motor repertoires in a changed semantic environment. Pretraining on the web does not in itself provide new physical skills that are not already known to the robot [6]. The two approaches are differentiated by their output interfaces as presented in Figure 2 without implying a universal performance ranking.

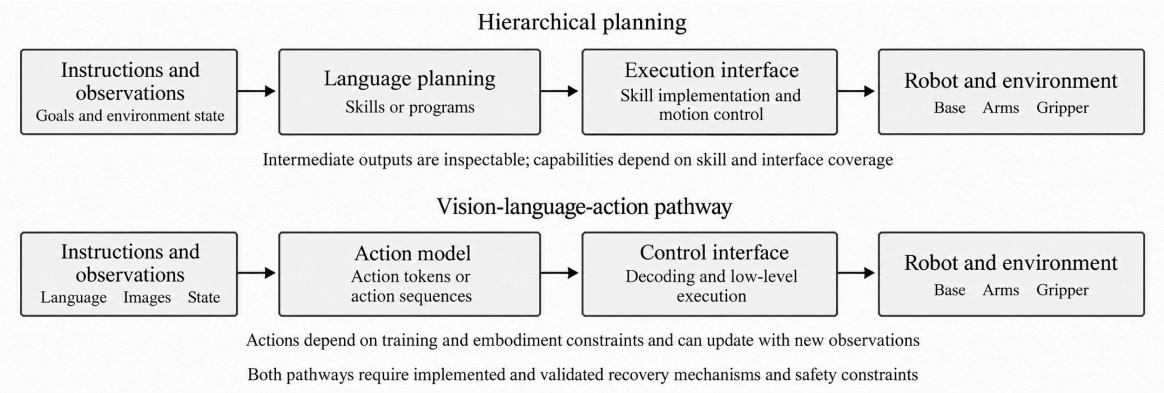


Figure 2. Interface-level comparison of hierarchical planning and vision-language-action policies

Note. The diagram distinguishes skill or program outputs from action outputs. Both approaches require execution feedback and appropriate deployment safeguards; it is not a benchmark result.

The 55B RT-2 configuration ran through cloud inference at approximately 1–3 Hz, compared with about 5 Hz for the 5B configuration [6]. These are policy inference rates, not servo-control frequencies. Communication, execution, and recovery time must also be included in an operational latency budget. For warehouse deployment, model selection should account for onboard memory, action-update timing, and communication delays, rather than parameter count alone.

5.2. Mobile ALOHA as a complementary imitation-learning approach

Mobile ALOHA combines whole-body teleoperation with imitation learning on a dual-arm platform and differential-drive base costing approximately US$32,000 in hardware [16]. Its actions comprise

DOI: 10.54254/2755-2721/2026.LD39094

14 target joint positions for the arms and grippers and two base velocities. Co-training incorporates 825 static ALOHA demonstrations by padding their base actions with zeros and mixing them with mobile demonstrations. The approach links accessible data collection to coordinated mobile manipulation without requiring language-model-based task decomposition.

For the Action Chunking with Transformers (ACT) policy, co-training increased wine-wiping success from 50% to 95%, whereas full cabinet-use success remained at 85% in both conditions. Each task used 50 mobile demonstrations and 20 evaluation trials per condition [16]. Cabinet opening requires both arms to hold the handles while the base moves backward, illustrating coupled motion. The different gains across tasks show that the benefit of additional data is task dependent. Hardware cost alone also understates the total cost of collecting demonstrations and adapting a system.

Table 2 retains task definitions, denominators, and within-study comparisons. Success rates from different papers are not directly ranked because their tasks and evaluation conditions differ.

Table 2. Representative mobile manipulation experiments and evaluation conditions

Study	Task and setting	Evaluation size	Success	Interpretation boundary
SayCan [4]	Mock kitchenPlanning / execution	101 instructions	84% / 74%	Different outcomes, not an algorithm comparison
OK-Robot [10]	10 homesFull pick-and-place	171 trials	58.5%	Non-warehouse setting; not grasp-only success
CENTAURO [12]	Fetch taskWithout / with recovery	25 trials per condition	80% / 88%	Execution time: 121.3 / 167.1 s
Mobile ALOHA [16]	Wipe wineWithout / with co-training	20 trials per condition	50% / 95%	50 mobile demonstrations per task
Mobile ALOHA [16]	Use cabinetWithout / with co-training	20 trials per condition	85% / 85%	50 mobile demonstrations per task

Note. The Mobile ALOHA entries compare ACT with and without static-data co-training. Sources: SayCan, Table 2 [4]; CENTAURO, Table IV [12]; Mobile ALOHA, Table 1 and Section 6.1 [16]; and the OK-Robot evaluation and project record [10].

6. Data and generalization across robot embodiments

6.1. Data constraints in robot learning

Robot learning is constrained by limited samples and narrow experience distributions. Physical interaction requires hardware, supervision, and environment preparation; demonstrations from one platform do not cover all products, arrangements, or failures [17, 18]. Dataset value depends not only on trajectory count but also on task coverage and whether collection effort can be shared across applications.

6.2. Positive transfer in open x-embodiment

The Open X-Embodiment collects over a million real world trajectories of 22 robot embodiment types [17]. Heterogeneous modalities are supported via a single storage format, whereas RT-X training chooses the observations and converts them into action and normalizes the values. Cross-platform learning is facilitated by the shared form; the original observations, coordinate systems and control definition do not have to be the same.

Lessons learned with other robots may help to make the tested platform more effective in tasks that are unsatisfactory covered by its own data [17]. There are some assessments where RT-2 and

DOI: 10.54254/2755-2721/2026.LD39094

RT-2-X have comparable performance with unseen objects and environments. The research will not give any straight generalization to unseen robot embodiments. Cross-embodiment co-training may help in reducing adaptation needs but this cannot be assured of zero-shot deployment on a new robot.

6.3. Action adaptation and transfer costs

Action arrays can share a format without sharing control semantics. Across datasets, an action may denote a position, displacement, or velocity [17], so the same numerical value need not produce the same motion. Fine manipulation further depends on timing, workspace limits, force-related information, and the low-level control interface. These differences must be addressed during transfer rather than hidden by a common data schema.

Octo is a generalist policy trained on 800,000 trajectories and designed for adaptation to different observation and action spaces [19]. DROID emphasizes diverse real-world data collection [18], while the Universal Manipulation Interface (UMI) connects demonstrations collected with handheld grippers to robot execution through its policy interface [20]. Table 3 separates these roles to avoid comparing a dataset, a collection interface, and a policy as equivalent algorithms.

Table 3. Roles and transfer boundaries of data and policy approaches

Work	Role	Relevant capability	Transfer boundary
Open X-Embodiment [17]	Cross-platform data and RT-X models	Standardized storage and shared robot experience	Gains on tested platforms do not establish zero-shot transfer to arbitrary robots
Octo [19]	Generalist policy	800,000 trajectories; adaptation of observation and action spaces	Initialization and fine-tuning are not warehouse validation
DROID [18]	Real-world dataset	Distributed collection and diverse scenes	Diversity does not replace target-task evaluation
UMI [20]	Collection and policy interface	Handheld grippers, relative trajectories, and latency matching	Deployment still depends on compatible motion and observation conditions

Note. These works serve different functions and are not ranked as competing algorithms.

Warehouse transfer costs include calibration, interface conversion, additional demonstrations, and debugging. Unseen objects, unseen layouts, and unseen embodiments define different test conditions. Reporting adaptation effort alongside task completion is necessary to determine whether a transfer benefit is practically useful.

7. Warehouse challenges and research directions

7.1. Open problems

Fine-grained perception must progress from category recognition to verification of the ordered item. Evaluation should distinguish false acceptance from uncertainty, record the cost of reobservation, and trace whether identification errors propagate into grasping or delivery. The system-level failure analysis in OK-Robot offers a useful precedent [10], but equivalent tests are needed on warehouse goods.

Reasoning reliability includes skill selection, feasibility checking, and recovery. SayCan, Code as Policies, and behavior-tree-based execution provide different interfaces for these functions [4, 11, 12], yet each remains vulnerable to perception errors and unmodeled conditions. Uncertain tasks

DOI: 10.54254/2755-2721/2026.LD39094

may be escalated to a human, provided that takeover triggers and missed detections are evaluated. An understandable plan is not necessarily a safe plan.

Architecture comparisons should return to matched task conditions. Explicit interfaces can help localize failures, while VLA policies may reduce some manually designed mappings from semantics to actions. A useful comparison holds product sets, layouts, data budgets, and intervention rules constant and retains a conventional hierarchical baseline. It should examine both maintenance effort and runtime cost rather than assume that an end-to-end formulation is inherently preferable.

7.2. Near-term priorities and evaluation conditions

Bounded recovery and economical adaptation are suitable near-term research priorities. Recovery can be tested by comparing order completion with and without recovery under a fixed total-time budget. Adaptation can be tested with a fixed number of additional demonstrations while varying objects, layouts, and embodiments separately. The task set should include continuous base-arm coordination rather than only grasping after the base has stopped.

An order-level protocol should specify item identity, quantity, delivery location, and deadline before execution. Complete success requires all criteria to be met, while partial completion, elapsed time, retries, and human takeovers are recorded separately. Figure 3 illustrates this proposed evaluation workflow, derived from the differences among published evaluations rather than from new experiments in this review. Safety monitoring must operate throughout execution and may interrupt the sequence before delivery verification.

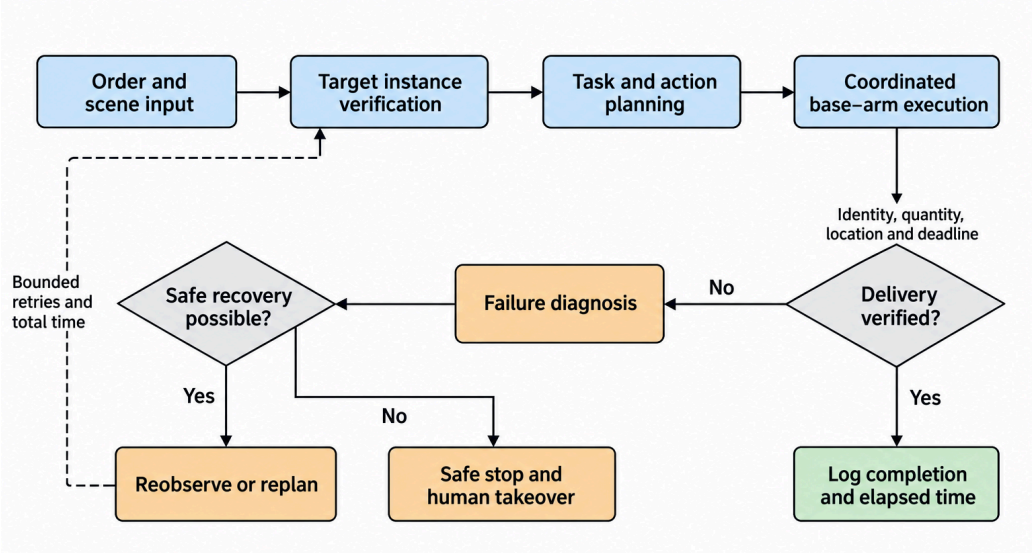


Figure 3. Proposed order execution and bounded failure-recovery workflow. Delivery checks concern task completion; continuous safety monitoring remains necessary throughout execution

7.3. Contribution and limitations

The review relates changes in information interfaces to warehouse constraints and identifies conditions under which particular methods may be useful. It does not introduce a new algorithm. Full-text availability and purposive selection limit coverage, and no formal risk-of-bias assessment was conducted. Much of the manipulation evidence comes from kitchens, homes, or laboratories,

DOI: 10.54254/2755-2721/2026.LD39094

with limited matched comparisons in warehouses. The synthesis therefore supports research design and critical method selection, not deployment certification.

8. Conclusion

Foundation models change how observations and instructions are connected to robot behavior. Open-vocabulary perception expands how targets can be described, while language grounding connects instructions to available skills. VLA policies learn a more direct mapping to action outputs. None of these developments removes the need to reconcile semantic decisions with geometry, control, and observed execution outcomes. Cross-embodiment data can help, but its value remains conditional on task coverage and adaptation requirements.

The other bottlenecks are materialized at the warehouse shelf. Occlusion may hinder checking of an item; alteration of packaging appearance, reflectivity or deformability influence both recognition and grasping. The onboard memory and power capacities also determine what models a mobile platform can support with the needed update frequency. Remote inference reduces certain local requirements, but it relies on the communication. These are not the only problems of warehouses. Together with SKU-level precision, limited shelf access and order deadlines it does however limit what can be inferred by successful household or laboratory demonstrations.

The key issue in warehouse mobile manipulation is whether the right products are being transported according to specified motion, time and intervention limits. The assessment must be based on validated order fulfillment and synchronized base arm movement as well as the expense of adjusting to evolving items and conditions. It can be used as an operational point of view to compare architectures without mixing up semantic flexibility with existing physical reliability.

References

[1] Ghodsian, N., Benfriha, K., Olabi, A., Gopinath, V., & Arnou, A. (2023). Mobile manipulators in Industry 4.0: A review of developments for industrial applications. *Sensors, 23*(19), Article 8026. https://doi.org/10.3390/s23198026

[2] Lan, F., Zhao, W., Zhu, K., & Zhang, T. (2024). Jiyu jushen zhineng de yidong caozuo jiqiren xitong fazhan yanjiu [Development of mobile manipulator robot system with embodied intelligence]. *Strategic Study of CAE, 26*(1), 139–148. https://doi.org/10.15302/j-sscae-2024.01.010

[3] 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. In M. Meila & T. Zhang (Eds.), *Proceedings of the 38th International Conference on Machine Learning (Vol. 139)* (pp. 8748–8763). PMLR. https://proceedings.mlr.press/v139/radford21a.html

[4] Ichter, B., Brohan, A., Chebotar, Y., Finn, C., Hausman, K., Herzog, A., Ho, D., Ibarz, J., Irpan, A., Jang, E., Julian, R., Kalashnikov, D., Levine, S., Lu, Y., Parada, C., Rao, K., Sermanet, P., Toshev, A. T., Vanhoucke, V., . . . Fu, C. K. (2023). Do as I can, not as I say: Grounding language in robotic affordances. In K. Liu, D. Kulic, & J. Ichnowski (Eds.), *Proceedings of the 6th Conference on Robot Learning (Vol. 205)* (pp. 287–318). PMLR. https://proceedings.mlr.press/v205/ichter23a.html

[5] 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. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, & J. Scarlett (Eds.), *Proceedings of the 40th International Conference on Machine Learning (Vol. 202)* (pp. 8469–8488). PMLR. https://proceedings.mlr.press/v202/driess23a.html

[6] Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P. R., Salazar, G., . . . Han, K. (2023). RT-2: Vision-language-action models transfer web knowledge to robotic control. In J. Tan, M. Toussaint, & K. Darvish (Eds.), *Proceedings of the 7th Conference on Robot Learning (Vol. 229)* (pp. 2165–2183). PMLR. https://proceedings.mlr.press/v229/zitkovich23a.html

DOI: 10.54254/2755-2721/2026.LD39094

[7] Lisondra, M., Benhabib, B., & Nejat, G. (2026). Embodied AI with foundation models for mobile service robots: A systematic review. *Robotics, 15*(3), Article 55. https://doi.org/10.3390/robotics15030055

[8] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations*. https://openreview.net/forum?id=YicbFdNTTy

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

[10] Liu, P., Orru, Y., Vakil, J., Paxton, C., Shafiullah, N., & Pinto, L. (2024). Demonstrating OK-Robot: What really matters in integrating open-knowledge models for robotics. In *Proceedings of Robotics: Science and Systems XX*. https://doi.org/10.15607/rss.2024.xx.091

[11] 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. In *2023 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 9493–9500). IEEE. https://doi.org/10.1109/icra48891.2023.10160591

[12] Wang, J., Laurenzi, A., & Tsagarakis, N. (2024). Autonomous behavior planning for humanoid loco-manipulation through grounded language model. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 10856–10863). IEEE. https://doi.org/10.1109/iros58592.2024.10802344

[13] 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. In J. Tan, M. Toussaint, & K. Darvish (Eds.), *Proceedings of the 7th Conference on Robot Learning (Vol. 229)* (pp. 540–562). PMLR. https://proceedings.mlr.press/v229/huang23b.html

[14] 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. (2023). RT-1: Robotics Transformer for real-world control at scale. In *Proceedings of Robotics: Science and Systems XIX*. https://doi.org/10.15607/rss.2023.xix.025

[15] Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E. P., Sanketi, P. R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., & Finn, C. (2025). OpenVLA: An open-source vision-language-action model. In P. Agrawal, O. Kroemer, & W. Burgard (Eds.), *Proceedings of the 8th Conference on Robot Learning (Vol. 270)* (pp. 2679–2713). PMLR. https://proceedings.mlr.press/v270/kim25c.html

[16] Fu, Z., Zhao, T. Z., & Finn, C. (2025). Mobile ALOHA: Learning bimanual mobile manipulation using low-cost whole-body teleoperation. In P. Agrawal, O. Kroemer, & W. Burgard (Eds.), *Proceedings of the 8th Conference on Robot Learning (Vol. 270)* (pp. 4066–4083). PMLR. https://proceedings.mlr.press/v270/fu25b.html

[17] 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. In *2024 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 6892–6903). IEEE. https://doi.org/10.1109/icra57147.2024.10611477

[18] Khazatsky, A., Pertsch, K., Nair, S., Balakrishna, A., Dasari, S., Karamcheti, S., Nasiriany, S., Srirama, M., Chen, L., Ellis, K., Fagan, P., Hejna, J., Itkina, M., Lepert, M., Ma, Y., Miller, P., Wu, J., Belkhale, S., Dass, S., . . . Finn, C. (2024). DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems XX*. https://doi.org/10.15607/rss.2024.xx.120

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

[20] Chi, C., Xu, Z., Pan, C., Cousineau, E., Burchfiel, B., Feng, S., Tedrake, R., & Song, S. (2024). Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems XX*. https://doi.org/10.15607/rss.2024.xx.045

[21] Correll, N., Bekris, K. E., Berenson, D., Brock, O., Causo, A., Hauser, K., Okada, K., Rodriguez, A., Romano, J. M., & Wurman, P. R. (2018). Analysis and observations from the first Amazon Picking Challenge. IEEE Transactions on Automation Science and Engineering, 15(1), 172–188. https://doi.org/10.1109/TASE.2016.2600527