Correct, Steer, Predict: Language-Mediated Deployment-Time Adaptation for Embodied Robots

Anonymous ACL submission

Abstract

In embodied robotics, language mainly specifies task goals before execution. Yet real-world deployment is rarely static: user intent may evolve, environments may shift, and failures may occur during execution. This motivates a broader role for language as a deployment-time adaptation interface, where it can redirect robot behavior without policy finetuning. This survey consolidates recent dispersed developments around three questions: what language adapts, where it intervenes, and how adaptation is implemented. We identify three paradigms: reactive behavior correction revises instantiated behavior, inference-time policy steering biases policy generation, and world-model-based adaptation uses predicted rollouts to guide behavior. We further analyze embodiment-specific constraints and gaps in deployment-time adaptation benchmarks and protocols. This survey offers an early map of the emerging area, clarifying task-space coverage, policy-dependent steering handles, and open challenges for deployment-time robot adaptation.

1 Introduction

Language is commonly used as a pre-execution interface for specifying what embodied agents should do, especially in modern language-conditioned robot policies (Kim et al., 2024; Octo Model Team et al., 2024; Black et al., 2025). Success in this setting depends on grounding an initial instruction into policy execution. Yet real-world deployment is rarely static: user intent may evolve, environments may shift, and failures may occur after action begins (Sui et al., 2025). Once execution unfolds, the original instruction may no longer be sufficient. This motivates a shift from language as task specification to language as a deployment-time adaptation interface, through which corrections, preferences, constraints, or new task-relevant information can redirect embodied behavior.

We study *language-mediated deployment-time adaptation*: the use of language during deployment to redirect robot behavior without task-specific policy finetuning for each new situation. Recent work has explored this broader role through language-conditioned planning, human-in-the-loop correction, inference-time steering, diffusion or flow-policy guidance, and world-model-based prediction. Yet these efforts remain scattered across policy architectures, adaptation targets, control mechanisms, and evaluation protocols, making it difficult to form a unified view of the area.

This survey organizes the emerging space by asking what language-mediated adaptation changes and where it intervenes in the embodied control stack. We identify three paradigms: *reactive behavior correction*, where language **corrects** instantiated behavior through goal refinement, plan revision, or recovery; *inference-time policy steering*, where language-derived preferences, constraints, rewards, or verifiers **steer** policy inference or action generation; and *world-model-based adaptation*, where language supports **prediction** by specifying what rollouts should be imagined, preferred, or avoided before behavior is selected (Figure 1; Secs. 2.1–2.3).

We next analyze how each paradigm is implemented across control architectures. Adaptation is both *policy-dependent* and *embodiment-dependent*: available steering handles vary across vision-language-action models, diffusion or flow policies, trajectory-revision modules, and world-model-based systems, while the same language feedback may require different physical responses across robot morphologies and task settings (Sec. 3). We also examine evaluation gaps, including limited task-space coverage, narrow adaptation bandwidth, and the scarcity of standardized online adaptation benchmarks and protocols (Sec. 4). Finally, we discuss future directions for deployment-time adaptation systems (Sec. 5).

Contributions. This survey makes four contributions: (i) **Scope**: it maps the emerging area of deployment-time language intervention for embodied robots; (ii) **Taxonomy**: it introduces a *correct–steer–predict* view that organizes recent work by adaptation targets, intervention loci, and control architectures; (iii) **Gaps**: it identifies key limitations in task-space coverage, adaptation bandwidth, and evaluation protocols; and (iv) **Outlook**: it outlines future opportunities, including compositional steering for deployment-time adaptation systems.

2 Language-Mediated Deployment-Time Adaptation

We first situate deployment-time adaptation within a taxonomy of language in embodied control (Appendix A), then examine three intervention roles: correction, steering, and prediction.

2.1 Reactive Plan and Behavior Correction

Reactive correction occupies a transitional space between instruction following and full deployment-time adaptation. Ideally, a deployed robot should revise its ongoing task strategy through its own policy or world model when new language feedback arrives. Existing methods approximate this goal through more constrained external interfaces, including online task re-conditioning, planner-objective modification, command-level repair, and failure-aware recovery.

Interactive Language (Lynch et al., 2023) represents a direct form of online task re-conditioning. It allows users to refine ongoing behavior by providing natural-language feedback during execution, which conditions a low-level visuomotor policy in real time. However, the correction mainly updates the task specification exposed to the policy. Correction is therefore bounded by the behavioral support of the trained imitation policy.

A more explicit form of test-time correction modifies the planner objective. Correcting Robot Plans (Sharma et al., 2022) translates language feedback into updates to a cost landscape, enabling users to correct robot plans without changing policy parameters. This provides an interpretable and deployable correction interface, but constrains adaptation to planner-compatible costs, such as spatial preferences or obstacle avoidance, in relatively structured tabletop settings.

Yell At Your Robot (Shi et al., 2024) moves correction into the task-command space. Rather than editing trajectories or costs, it uses verbal corrections as supervision for learning when to issue high-level repair commands, such as *"open the gripper wider"*. We include it as a post-training boundary case: language feedback is not merely an instruction update, but a signal for selecting corrective task-level actions during execution.

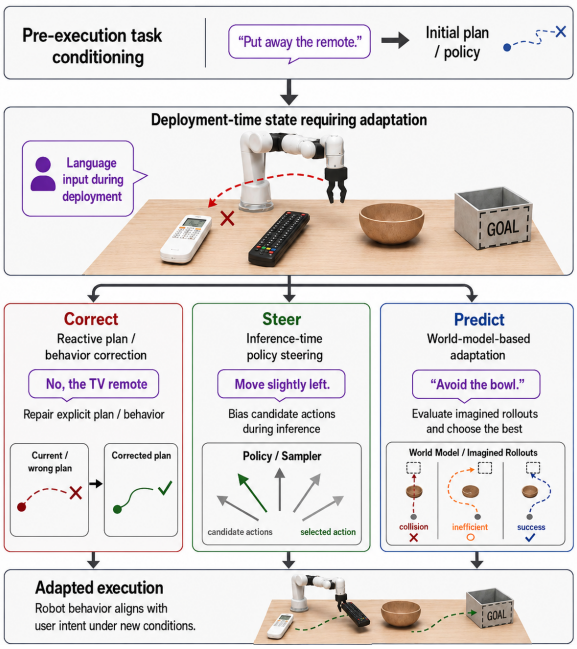

Figure 1: Language-mediated deployment-time adaptation by intervention locus. Language can correct instantiated behavior (Sec. 2.1), steer policy generation during inference (Sec. 2.2), or guide predictive rollouts for adapted decision-making (Sec. 2.3).

Beyond direct language correction, recent failure-aware systems extend this reactive view through foundation-model reasoning: they detect failures, infer causes or execution status, and trigger corrective behaviors (Duan et al., 2025; Zhou et al., 2025a; Lin et al., 2025; Li et al., 2026a). Together with direct correction methods, they make semantic feedback and reasoning actionable around robot behavior. However, their shared limitation remains *correction bandwidth*: adaptation is local, short-horizon, and constrained by predefined policy or command interfaces, limiting fine-grained changes in contact, force, dexterity, coordination, and open-ended semantic repair.

2.2 Inference-Time Policy Steering

Inference-time policy steering adapts deployment behavior by intervening on policy outputs. We group methods by intervention handle: *external verification* of generated proposals, *sampler-level*

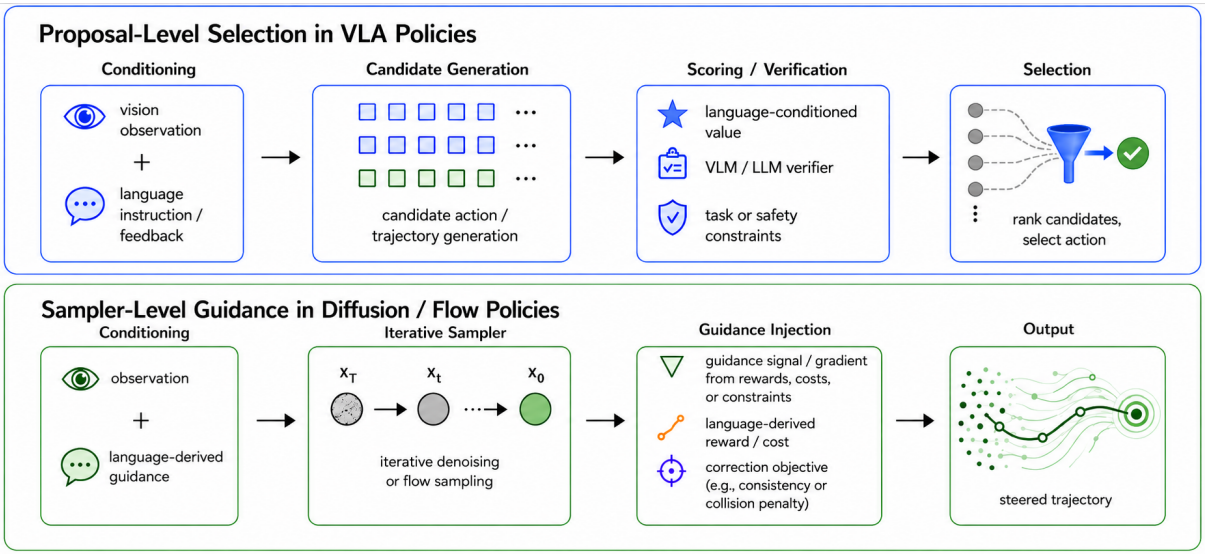

Figure 2: Two architecture-dependent handles for inference-time policy steering. In VLA policies, proposal-level selection ranks candidates using language-conditioned scores, verifiers, or constraints. In diffusion or flow policies, sampler-level guidance injects language-derived objectives into sampling to steer trajectory generation.

guidance within generative policies (Figure 2), and *trajectory revision* of existing behavior traces.

2.2.1 Steering via External Verification

Value-Guided Policy Steering (V-GPS) (Nakamoto et al., 2024) treats a frozen vision-language-action (VLA) policy as an action proposal generator and uses a language-conditioned value function to rerank candidate actions. At deployment time, it steers behavior by selecting higher-scoring proposals before execution. However, scalar value ranking explains little beyond preference: it does not reveal violated semantic constraints or failures that may emerge after execution. For semantically implicit or open-vocabulary preferences, action selection may therefore require outcome-level verification rather than scalar ranking alone.

FOREWARN (Wu et al., 2025) addresses this limitation by rolling out action plans with a latent world model and aligning latent predictions with language, allowing predicted future states to be evaluated by a vision-language model (VLM) verifier against the task description. This shifts verification from action scoring to semantic evaluation over predicted consequences. SEAL (Wu et al., 2026) provides a simulation-based counterpart: it samples candidate action sequences, predicts their outcomes through simulation, and uses a VLM verifier to select the outcome most consistent with the textual plan. In both cases, action diversity becomes a runtime selection resource.

UPS (Yuan et al., 2026b) shifts attention from candidate quality to verifier uncertainty. By calibrating the VLM verifier with conformal prediction, it maps uncertainty to resolution strategies: execute confident actions, ask users to clarify ambiguous instructions, or request low-level intervention when the policy is incapable. Proposal-level steering therefore becomes decision-level regulation, deciding not only *which* candidate to execute but also *whether* execution is appropriate.

This proposal-level logic also appears in recent test-time scaling methods for VLA policies, which generate multiple candidate actions or action chunks and select among them using VLM-based verifiers, task-progress verifiers, internal confidence signals, or relative action critics (Kwok et al., 2025; Zhao et al., 2026; Jang et al., 2026; Li et al., 2026b). These methods broaden proposal-level steering from explicit value guidance and semantic rollout verification to more general candidate generation and selection mechanisms.

Overall, proposal-level steering assumes pre-execution evaluation through value scores, predicted outcomes, verbalized rollouts, or verifier confidence. It is most tractable in simulation or structured tabletop domains, while real-world deployment remains bounded by latent-to-language omissions, real–sim gaps, verifier reliability, and candidate diversity.

2.2.2 Steering via Sampler-Level Guidance

For diffusion- and flow-based policies (Chi et al., 2023; Zhang et al., 2025), iterative generation ex-

poses a natural intervention handle. Instead of selecting among completed proposals, sampler-level guidance steers trajectories during generation by injecting language-derived rewards, costs, or constraints into the sampling dynamics. This makes sampler-level steering suitable for continuous motion preferences, trajectory shaping, and local behavioral adaptation (Wagenmaker et al., 2025).

Inference-Time Policy Steering (ITPS) (Wang et al., 2025) formulates this idea as test-time optimization over a frozen diffusion policy. Given a steering objective or cost, ITPS modifies denoising so that sampled trajectories are biased toward satisfying the guidance signal. This enables deployment-time redirection when the desired adaptation can be expressed as a computable trajectory-level objective, without updating policy parameters.

Vision-Language Steering (VLS) (Liu et al., 2026) automates objective specification by grounding language into task-relevant keypoints and predefined geometric reward templates. This supports spatial adaptation objectives such as *moving an object to a target*. However, template-based rewards remain limited for contact dynamics, force regulation, deformable objects, or dexterous in-hand manipulation, where simple distance-, projection-, or direction-based penalties are insufficient.

Adapt as You Say (BiSAIL) (Li et al., 2026c) broadens the language interface with a reason-then-steer design for bimanual adaptation. Given visual observations and verbal feedback, it infers a generalized adaptation objective through embodied skill-adaptation reasoning, then uses this objective to guide diffusion-based bimanual motion generation. Thus, although language enters through high-level deliberation, the effective intervention remains sampler-level: the inferred objective modulates trajectory generation at inference time.

Taken together, sampler-level guidance is most useful when the pretrained policy already contains the relevant motor skill and the desired adaptation remains near its behavior manifold. It also assumes that user intent can be compiled into a smooth, computable sampling objective. Language can specify desired outcomes, forbidden states, spatial preferences, or other constraints, making sampler-level steering expressive for continuous motion adaptation but still limited for contact dynamics, force regulation, dexterity, or objectives that are difficult to encode as differentiable guidance signals.

2.2.3 Steering via Trajectory-Level Revision

Trajectory-level revision treats an already instantiated motion trace as the adaptation substrate, typically a low-dimensional Cartesian trajectory represented by waypoints with simple kinematic attributes. Language reshapes this trace toward user preferences, shifting the intervention from policy inference to *post-policy behavior revision*.

LATTE (Bucker et al., 2023) casts language-conditioned trajectory revision as *supervised trajectory translation*, learning to map an existing waypoint trace to a language-aligned revision. Its supervision is generated by deforming base trajectories with handcrafted language-induced vector fields, which preserves coarse geometry and feasibility but limits revision to predefined spatial or speed-profile edits rather than open-ended corrective feedback.

OVITA (Maurya et al., 2025) uses LLMs to translate user instructions into executable waypoint-editing code, moving trajectory revision toward open-vocabulary and interpretable adaptation. This enables precise numerical changes, compositional commands, local waypoint-specific edits, and insertion of new waypoint patterns. For example, *"move forward while gradually slowing down to a stop"* combines spatial displacement, velocity adjustment, and terminal stopping behavior, beyond LATTE-style template-conditioned revision.

These methods are bounded by the quality and editability of the initial trace. They support local semantic edits to otherwise feasible behavior, such as waypoint shifts, region avoidance, speed changes, or object-configuration adaptation, but struggle with substantial replanning, infeasible trajectories, and contact- or deformation-rich tasks where success depends on pose, force, and coordination rather than path geometry alone.

Towards Compositional Steering. The three steering handles intervene at different points in the deployment-time pipeline and can be composed. Policies or planners propose candidate trajectories, sampler-level guidance biases generation, trajectory revision repairs instantiated traces, and proposal selection evaluates candidates. This suggests a *proposal–adapt–select* loop: generate trajectories, revise them according to language-specified preferences, and select among them for execution. OVITA-style revision can thus serve as a post-policy refinement layer for preference alignment without retraining the base policy.

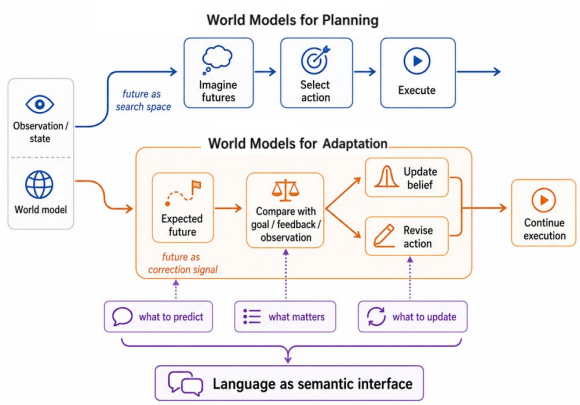

Figure 3: World-model-based adaptation. In contrast to planning-oriented use, where imagined futures support action selection, adaptation-oriented world models treat expected futures as correction signals to update beliefs or revise actions. Language provides a semantic interface for specifying what to predict, what matters, and what to update during deployment.

2.3 World-Model-Based Adaptation

Beyond correcting instantiated behavior or steering policy generation, adaptation can also operate through foresight. World models shift deployment-time adaptation from reacting to current behavior toward reasoning over possible futures. In this paradigm, language specifies what to predict, how imagined futures should be evaluated, and which assumptions should be updated during deployment.

In embodied decision-making, world models commonly support planning and policy learning by predicting visual or latent futures from current observations and candidate actions (Hafner et al., 2019, 2020; Du et al., 2023). These futures provide a search or optimization substrate for selecting actions, learning behaviors, or generating video plans. For deployment-time adaptation, however, prediction must expose consequences that can be compared with goals, feedback, or observations and used to update beliefs or revise actions. Such consequences may include imagined observations, semantic relations, physical states, contact outcomes, or latent dynamics. Language couples to this predictive loop by specifying which future states to query, what outcomes matter, which physical assumptions are relevant, or which prediction errors should trigger adaptation (Figure 3).

Semantic World Models (SWM) (Berg et al., 2026) instantiate this idea by recasting world modeling as action-conditioned visual question answering. Rather than representing futures only as pixels or latent states, SWM evaluates candidate action sequences through task-relevant questions about expected outcomes. Task goals are translated into desired question–answer predicates, whose likelihood provides a semantic planning score. This lets language specify both what to query in the imagined future and how to evaluate the predicted outcome. However, adaptation remains limited to properties that can be expressed and scored semantically, making contact-rich dynamics and fine geometric tolerances harder to address.

Phys2Real (Wang et al., 2026) targets physical adaptation under latent object properties, such as mass distribution, contact geometry, and friction, which determine feasible grasps, support points, and motion strategies. It uses a vision-language model (VLM) to infer priors about an object's center of mass from visual observations and language-grounded commonsense, then refines these priors through real-world grasping interactions and uses them to condition control. Here, language links visual appearance and object semantics to plausible physical properties, bootstrapping object-specific physical reasoning. However, the current formulation is best suited to compact object-level mismatches; richer contact dynamics, deformable objects, and evolving contact states remain harder to model and update online.

AdaWorldPolicy (Yuan et al., 2026a) represents prediction-error-driven adaptation through a policy-internal world model. It couples action generation with future visual and force prediction in a diffusion policy conditioned on text embeddings and robot state. During deployment, the policy executes generated actions and compares predicted visual–force outcomes with observations. The resulting prediction errors provide self-supervised losses for online updates, enabling adaptation to visual and physical domain shifts without external labels. Here, language specifies the task context, while deployment-time correction is driven by world- and force-prediction errors.

OSVI-WM (Goswami et al., 2025) is a boundary case: it uses world-model rollouts to generate waypoints for unseen tasks, supporting test-time task transfer rather than mid-execution language-mediated adaptation.

Overall, language contributes to world-model-based adaptation by structuring what is predicted, how imagined futures are evaluated, and which latent physical assumptions are initialized for online refinement. Yet reliability remains the central challenge: when imagined semantics, inferred physical

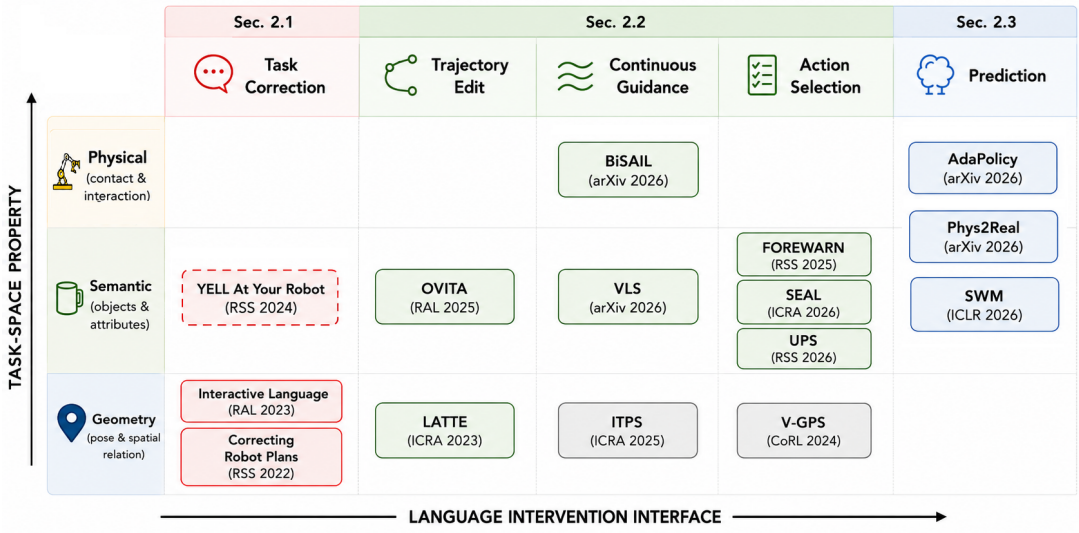

Figure 4: Task-space view of deployment-time language intervention. Representative methods are organized by the task-space property they primarily ground and by the intervention interface. Gray boxes indicate non-language-framed methods included for comparison, while dashed boxes denote related post-training adaptation methods included for completeness. See Table C.2 for a fuller method summary.

parameters, or predicted sensory outcomes are inaccurate, adaptation may optimize against a misleading future. Fine-grained multimodal grounding is therefore central to reliable world-model-based adaptation in robot manipulation.

Intervention-space synthesis. Deployment-time language interventions act on different targets, including instantiated behaviors, trajectories, policy generation, candidate proposals, and imagined futures (Sec. 2.2.1–2.2.3). They also differ in primary task-space coverage, from geometric and semantic properties to physical interaction properties. Figure 4 positions the methods reviewed above along these two axes, highlighting existing concentrations and underexplored regions (Appendix Table C.2).

3 Embodiment Constraints on Deployment-Time Adaptation

Test-time adaptation in robotics remains an embodied control problem: adaptations must be physically feasible, contact-consistent, and temporally stable. Language can specify *what* should change; embodiment determines *whether* and *how* it can be changed.

Language-mediated adaptation is therefore *embodiment-dependent*. For example, the feedback *"hold the object more steadily"* may require different responses across embodiments. A single-arm gripper may adjust grasp pose or gripping force, a bimanual robot may redistribute load between arms, and a dexterous hand may change finger-level contact configurations.

Existing language-mediated adaptation studies have primarily been evaluated in single-arm tabletop manipulation, where corrections can often be mapped to object goals, end-effector poses, or short-horizon trajectories (Liu et al., 2026; Wang et al., 2026). Richer embodiments, such as bimanual and dexterous systems, are not merely larger action spaces; they change the structure of adaptation because feedback must be grounded into coordinated contacts, load sharing, and temporally coupled control.

Adjacent work on bimanual inference-time control highlights this shift. Tactile-guided policy steering, safety-constrained test-time optimization, and update-free verifier-based steering show how bimanual diffusion policies can be adapted or regulated during deployment, even when language is not the primary interface (Zhang et al., 2026; Deng et al., 2025; Attarian et al., 2026). These studies suggest that richer embodiments require adaptation mechanisms that monitor contact, safety, and coordination rather than only geometric task progress.

Among language-mediated methods, BiSAIL (Li et al., 2026c) studies online interactive adaptation of offline-learned bimanual skills, where verbal feedback is grounded into coordinated dual-arm motion. However, bimanual language feedback often extends beyond spatial coordination:

Table 1: Language feedback types and embodied adaptation targets. Feedback is grouped by grounding level, from motion/contact adjustment and semantic constraints to task-state adaptation.

Type	Language Feedback Example	Adaptation Target
	Motion-level grounding	
Spatial	*"Move more to the left."* *"Place it closer to the bowl."*	End-effector pose, object pose, placement region, trajectory geometry
Temporal	*"Wait until both hands hold it."* *"Slow down before contact."*	Action timing, synchronization, velocity profile, execution order
Force/contact	*"Do not squeeze it."* *"Slide instead of lifting."*	Grasp force, contact mode, compliance, frictional interaction
	Semantic constraint grounding	
Semantic	*"Use the safer path."* *"Avoid the fragile object."*	Object affordances, implicit constraints, safety or user preferences
	Task-state adaptation	
Task revision	*"Actually put it in the other drawer."* *"Use the blue container instead."*	Goal state, target object, receptacle, or task specification
Failure recovery	*"It slipped; regrasp before continuing."*	Partial failure state, recovery action, continuation strategy

i) Force–contact coupling: *"put more weight on the left"* requires force adaptation, including grip regulation, load sharing, and torque balancing across the two arms. *"slide it along the table instead of lifting it"* further requires deciding which contacts should be maintained or released.

ii) Temporal coordination: *"do not twist until both hands are firmly holding it"* introduces temporal and dynamic constraints. The system must coordinate timing, speed, and execution phases across both arms, beyond modifying the geometric path.

iii) Functional role assignment: *"use the left hand to stabilize the base and the right hand to insert the peg"* requires assigning asymmetric roles, where one arm acts as a stabilizer and the other as an active manipulator.

Takeaway. Deployment-time adaptation is *policy-dependent*, *embodiment-dependent*, and *task-/feedback-dependent*: policy architectures determine the steering handle, embodiments determine the feasible response, and task context determines what the feedback should change.

4 Evaluation Benchmarks and Gaps

Existing evaluation resources only weakly instantiate language-mediated deployment-time adaptation. Language-conditioned benchmarks provide linguistic inputs, but mainly evaluate pre-execution task specification rather than feedback-driven adaptation. Robot learning suites support perturbations, scene variations, and changed goals, but typically lack language-feedback protocols. As a result, adaptation studies often rely on custom evaluation settings. This section reviews benchmark gaps and summarizes language feedback types and adaptation targets in Table 1; Appendix C further distinguishes task generalization from deployment-time adaptation and surveys representative language-conditioned manipulation resources.

Language-conditioned manipulation testbeds. CALVIN (Mees et al., 2022), LIBERO (Liu et al., 2023), Language Table (Lynch et al., 2023), and Language-World (Zentner et al., 2024) primarily evaluate language-conditioned task execution. In these benchmarks, language is usually provided before execution as a task specification, rather than during execution as feedback for revising goals, preferences, constraints, or recovery strategies. A broader summary of such testbeds and training resources is provided in Appendix C.2, Table 3 and Table 4.

Robot learning suites. ManiSkill2 (Gu et al., 2023), Robosuite/robomimic (Zhu et al., 2020; Mandlekar et al., 2021), and Meta-World (Yu et al., 2020) evaluate robustness, multi-task learning, and generalization under object, scene, dynamics, or task variations, but are largely non-linguistic. Recent large-scale platforms such as RoboVerse (Geng et al., 2025) and RoboTwin (Mu et al., 2025) add language-specified tasks or VLA-oriented evaluation, but still lack deployment-time language-feedback protocols.

Study-specific evaluation protocols. Recent works often rely on ad hoc task variants to test deployment-time redirection through language feedback. ITPS (Wang et al., 2025), for example, uses a 4-block stacking domain in Isaac Sim to test whether a frozen pick-and-place policy can be steered toward a user-specified stacking sequence among multiple feasible modes. SEAL (Wu et al., 2026) and VLS (Liu et al., 2026) introduce study-specific perturbation protocols on CALVIN (Mees et al., 2022) and LIBERO-PRO (Zhou et al., 2025b) to evaluate inference-time steering under spatial or semantic shifts. Such evaluations remain narrow and fragmented, with limited scenario coverage and cross-method comparability.

Grounding levels and benchmark coverage. Table 1 groups language feedback by where it must be grounded: motion-level control, semantic constraints, and task-state adaptation. Existing benchmarks cover spatial and object-related semantic feedback relatively well, as in LIBERO / LIBERO-PRO (Liu et al., 2023; Zhou et al., 2025b). Coverage is weaker for temporal and force/contact feedback, which require execution traces, contact states, force profiles, or dynamics-aware metrics. The largest gap is task-state adaptation, where task revision and failure recovery require evaluating state recognition, feedback interpretation, and corrected continuation. Overall, current benchmarks provide limited evidence for error-aware correction under changing deployment states.

5 Future Directions

Embodied robots are expected to adapt to evolving human intent, changing physical states, and execution errors. This calls for language-mediated systems that translate feedback into policy-compatible, physically grounded, and verifiable adaptation signals. We highlight several directions toward reliable deployment-time adaptation.

Intermediate adaptation signals. As embodied agents move beyond instruction following, directly mapping language feedback to new actions becomes a brittle adaptation mechanism. A key challenge is to compile feedback into structured intermediate signals, such as task constraints, reward functions, cost landscapes, verifier scores, or latent steering directions. These signals can bridge ambiguous human intent and policy-specific control representations, while making adaptation more modular, inspectable, and physically grounded.

Embodied grounding. Language should be grounded in embodiment-specific control rather than treated as a body-agnostic objective. Future methods should map linguistic intent to each robot's feasible adaptation variables, including geometry, force, contact, timing, and inter-effector coordination. This requires modeling affordances, limits, and dynamics to keep adaptation physically grounded across embodiments.

Compositional deployment-time control. Reactive correction, inference-time steering, and world-model prediction are often studied separately, but deployment-time adaptation will likely require them to work together. Future systems should predict likely failures, steer policy generation toward preferred behaviors, and correct execution when behavior deviates from intent. This composition could support more reliable closed-loop adaptation beyond one-shot language intervention.

Towards evaluating closed-loop adaptation. Evaluating physical intelligence requires more than measuring whether a robot completes an instruction. Future benchmarks should capture adaptation loops, where robot states, actions, language feedback, and corrected behaviors are jointly observed under changing conditions. Evaluation should include adaptation efficiency, human effort, semantic fidelity, physical feasibility, failure recovery, and robustness across embodiments and environments, so that language-mediated systems are assessed by their ability to support error-aware correction rather than narrow task completion (Appendix Sec. C.2).

6 Conclusion

Language in embodied robotics is shifting from pre-execution task specification to deployment-time adaptation. This survey synthesized recent work through three intervention roles: correcting instantiated behavior, steering policy inference, and supporting predictive adaptation through world models. This functional view clarifies where language enters the embodied control stack and reshapes behavior after policy training. We further identified embodiment-specific constraints and evaluation gaps that must be addressed for reliable deployment. Progress depends on physically grounded, policy-compatible, and verifiable mechanisms that can translate changing goals, feedback, and deployment states into executable behavioral revisions.

Limitations

This survey focuses on language-mediated deployment-time adaptation, where language serves as the primary interface for correcting, steering, or predicting embodied behavior. We therefore do not cover studies that steer policies through *internal representations* rather than external semantic feedback. For example, (Häon et al., 2025) identifies semantic directions in VLA activations through vocabulary projection and intervenes on them at inference time to bias action tendencies. Although such representation-level steering may better integrate policy inference and execution, its embedded control signal makes behavioral changes less interpretable, harder to express through human feedback, and more difficult to constrain or verify than explicit language-mediated adaptation signals.

Ethics Statement

AI assistants were used for language polishing and clarity improvement. The authors remain fully responsible for the manuscript.

References

AGIBOT WORLD. 2026. Agibot world 2026 dataset. Hugging Face dataset. Accessed 2026-05-25.

Allen Institute for AI. 2026. Molmoact2: Action reasoning models for real-world deployment. *Preprint*, arXiv:2605.02881.

Maria Attarian, Ian Vyse, Claas Voelcker, Jasper Gerigk, Evgenii Opryshko, Anas Almasri, Sumeet Singh, Yilun Du, and Igor Gilitschenski. 2026. Update-free on-policy steering via verifiers. *Preprint*, arXiv:2603.10282.

Jacob Berg, Chuning Zhu, Yanda Bao, Ishan Durugkar, and Abhishek Gupta. 2026. Semantic world models. In *Proceedings of the International Conference on Learning Representations*.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, and 5 others. 2025. π_0: A vision-language-action flow model for general robot control. *Preprint*, arXiv:2410.24164.

Arthur Bucker, Luis Figueredo, Sami Haddadin, Ashish Kapoor, Shuang Ma, Sai Vemprala, and Rogerio Bonatti. 2023. LATTE: LAnguage Trajectory TransformEr. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 7287–7294.

Jiahang Cao, Yize Huang, Hanzhong Guo, Rui Zhang, Mu Nan, Weijian Mai, Jiaxu Wang, Hao Cheng, Jingkai Sun, Gang Han, Wen Zhao, Qiang Zhang, Yijie Guo, Qihao Zheng, Chunfeng Song, Xiao Li, Ping Luo, and Andrew F. Luo. 2026. Compose your policies! improving diffusion-based or flow-based robot policies via test-time distribution-level composition. In *International Conference on Learning Representations*.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, Weiliang Deng, Yubin Guo, Tian Nian, Xuanbing Xie, Qiangyu Chen, Kailun Su, Tianling Xu, Guodong Liu, Mengkang Hu, and 7 others. 2025. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *Preprint*, arXiv:2506.18088.

Zeren Chen, Zhelun Shi, Xiaoya Lu, Lehan He, Sucheng Qian, Zhenfei Yin, Wanli Ouyang, Jing Shao, Yu Qiao, Cewu Lu, and Lu Sheng. 2024. RH20T-P: A primitive-level robotic dataset towards composable generalization agents. *Preprint*, arXiv:2403.19622.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. 2023. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea.

Haoyuan Deng, Wenkai Guo, Qianzhun Wang, Zhenyu Wu, and Ziwei Wang. 2025. SafeBimanual: Diffusion-based trajectory optimization for safe bimanual manipulation. *Preprint*, arXiv:2508.18268.

Yilun Du, Mengjiao Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B. Tenenbaum, Dale Schuurmans, and Pieter Abbeel. 2023. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36.

Jiafei Duan, Wilbert Pumacay, Nishanth Kumar, Yi Ru Wang, Shulin Tian, Wentao Yuan, Ranjay Krishna, Dieter Fox, Ajay Mandlekar, and Yijie Guo. 2025. AHA: A vision-language-model for detecting and reasoning over failures in robotic manipulation. In *International Conference on Learning Representations*.

Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. 2023. RH20T: A comprehensive robotic dataset for learning diverse skills in one-shot. *Preprint*, arXiv:2307.00595. RSS 2023 LTAMP Workshop poster.

Haoran Geng, Feishi Wang, Songlin Wei, Yuyang Li, Bangjun Wang, Boshi An, Charlie Tianyue Cheng, Haozhe Lou, Peihao Li, Yen-Jen Wang, Yutong Liang, Dylan Goetting, Chaoyi Xu, Haozhe Chen,

Yuxi Qian, Yiran Geng, Jiageng Mao, Weikang Wan, Mingtong Zhang, and 18 others. 2025. Roboverse: A unified platform, benchmark and dataset for scalable and generalizable robot learning. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA.

Raktim Gautam Goswami, Prashanth Krishnamurthy, Yann LeCun, and Farshad Khorrami. 2025. OSVI-WM: One-shot visual imitation for unseen tasks using world-model-guided trajectory generation. *Preprint*, arXiv:2505.20425.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. 2023. Maniskill2: A unified benchmark for generalizable manipulation skills. In *Proceedings of the International Conference on Learning Representations*.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. 2020. Dream to control: Learning behaviors by latent imagination. In *Proceedings of the International Conference on Learning Representations*.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2555–2565. PMLR.

Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, Fei Liao, Guangrun Li, Jingyang He, Qiuxuan Feng, Zhao Jin, Chenyang Gu, Zhuoyang Liu, Nuowei Han, Xiangju Mi, Yaoxu Lv, Yankai Fu, Gaole Dai, Langzhe Gu, Tao Li, and 14 others. 2025. Robomind 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence. *Preprint*, arXiv:2512.24653.

Bear Häon, Kaylene Stocking, Ian Chuang, and Claire Tomlin. 2025. Mechanistic interpretability for steering vision-language-action models. *Preprint*, arXiv:2509.00328.

Suhyeok Jang, Dongyoung Kim, Changyeon Kim, Youngsuk Kim, and Jinwoo Shin. 2026. Verifier-free test-time sampling for vision-language-action models. In *International Conference on Learning Representations*.

Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. 2023. Vima: General robot manipulation with multimodal prompts. *Preprint*, arXiv:2210.03094.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, and 82 others. 2024. DROID: A large-scale in-the-wild robot manipulation dataset. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2024. Openvla: An open-source vision-language-action model. *Preprint*, arXiv:2406.09246.

Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. 2025. Robomonkey: Scaling test-time sampling and verification for vision-language-action models. In *Proceedings of Machine Learning Research*.

Wenhao Li, Xiu Su, Yichao Cao, Hongyan Xu, Xiaobo Xia, Shan You, Yi Chen, and Chang Xu. 2026a. Sentinel-VLA: A metacognitive VLA model with active status monitoring for dynamic reasoning and error recovery. *Preprint*, arXiv:2605.01191.

Wenhao Li, Xiu Su, Dan Niu, Yichao Cao, Hongyan Xu, Zhe Qu, Lei Fan, Shan You, and Chang Xu. 2026b. VLA-ATTC: Adaptive test-time compute for VLA models with relative action critic model. *Preprint*, arXiv:2605.01194.

Zhuo Li, Dianxi Li, Tao Teng, Quentin Rouxel, Zhipeng Dong, Dennis Hong, Darwin Caldwell, and Fei Chen. 2026c. Adapt as You Say: Online Interactive Bimanual Skill Adaptation via Human Language Feedback. https://arxiv.org/abs/2603.26466v1.

Zijun Lin, Jiafei Duan, Haoquan Fang, Dieter Fox, Ranjay Krishna, Cheston Tan, and Bihan Wen. 2025. Failsafe: Reasoning and recovery from failures in vision-language-action models. *Preprint*, arXiv:2510.01642.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. 2023. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Preprint*, arXiv:2306.03310.

Shuo Liu, Ishneet Sukhvinder Singh, Yiqing Xu, Jiafei Duan, and Ranjay Krishna. 2026. Vls: Steering pretrained robot policies via vision-language models. *Preprint*, arXiv:2602.03973.

Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. 2023. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 8(9):5217–5224.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. 2021. What matters in learning from offline

human demonstrations for robot manipulation. In *Conference on Robot Learning*.

Anurag Maurya, Tashmoy Ghosh, Anh Nguyen, and Ravi Prakash. 2025. Ovita: Open-vocabulary interpretable trajectory adaptations. *IEEE Robotics and Automation Letters*. Accepted.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. 2022. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *Preprint*, arXiv:2112.03227.

Yao Mu, Tianxing Chen, Zanxin Chen, Shijia Peng, Zhiqian Lan, Zeyu Gao, Zhixuan Liang, Qiaojun Yu, Yude Zou, Mingkun Xu, Lunkai Lin, Zhiqiang Xie, Mingyu Ding, and Ping Luo. 2025. Robotwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Mitsuhiko Nakamoto, Oier Mees, Aviral Kumar, and Sergey Levine. 2024. Steering your generalists: Improving robotic foundation models via value guidance. In *Proceedings of the Conference on Robot Learning*.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. 2024. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands.

Pratyusha Sharma, Balakumar Sundaralingam, Valts Blukis, Chris Paxton, Tucker Hermans, Antonio Torralba, Jacob Andreas, and Dieter Fox. 2022. Correcting robot plans with natural language feedback. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA.

Lucy Xiaoyang Shi, Zheyuan Hu, Tony Z. Zhao, Archit Sharma, Karl Pertsch, Jianlan Luo, Sergey Levine, and Chelsea Finn. 2024. Yell At Your Robot: Improving On-the-Fly from Language Corrections. *Preprint*, arXiv:2403.12910.

Xiuchao Sui, Daiying Tian, Qi Sun, Ruirui Chen, Dongkyu Choi, Kenneth Kwok, and Soujanya Poria. 2025. From grounding to manipulation: Case studies of foundation model integration in embodied robotic systems. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1324–1340, Suzhou, China. Association for Computational Linguistics.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. 2025. Steering your diffusion policy with latent space reinforcement learning. *Conference on Robot Learning*.

Homer Rich Walke, Kevin Black, Tony Z. Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, Abraham Lee, Kuan Fang, Chelsea Finn, and Sergey Levine. 2023. BridgeData V2: A dataset for robot learning at scale. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1723–1736. PMLR.

Maggie Wang, Stephen Tian, Aiden Swann, Ola Shorinwa, Jiajun Wu, and Mac Schwager. 2026. Phys2real: Fusing vlm priors with interactive online adaptation for uncertainty-aware sim-to-real manipulation. *Preprint*, arXiv:2510.11689.

Yanwei Wang, Lirui Wang, Yilun Du, Balakumar Sundaralingam, Xuning Yang, Yu-Wei Chao, Claudia Perez-D'Arpino, Dieter Fox, and Julie Shah. 2025. Inference-time policy steering through human interactions. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Yilin Wu, Anqi Li, Tucker Hermans, Fabio Ramos, Andrea Bajcsy, and Claudia Pérez-D'Arpino. 2026. Do what you say: Steering vision-language-action models via runtime reasoning-action alignment verification. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Yilin Wu, Thomas Tian, Gokul Swamy, and Andrea Bajcsy. 2025. From foresight to forethought: Vlm-in-the-loop policy steering via latent alignment. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. 2020. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR.

Ge Yuan, Qiyuan Qiao, Jing Zhang, and Dong Xu. 2026a. Adaworldpolicy: World-model-driven diffusion policy with online adaptive learning for robotic manipulation. *Preprint*, arXiv:2602.20057.

Jessie Yuan, Yilin Wu, and Andrea Bajcsy. 2026b. When to act, ask, or learn: Uncertainty-aware policy steering. In *Proceedings of Robotics: Science and Systems*.

K. R. Zentner, Ryan Julian, Brian Ichter, and Gaurav S. Sukhatme. 2024. Conditionally combining robot skills using large language models. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. 2025. Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. 2024. Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. *Preprint*, arXiv:2412.18194.

Zhemeng Zhang, Jiahua Ma, Xincheng Yang, Xin Wen, Yuzhi Zhang, Boyan Li, Yiran Qin, Jin Liu, Can Zhao, Li Kang, Haoqin Hong, Zhenfei Yin, Philip Torr, Hao Su, Ruimao Zhang, and Daolin Ma. 2026. TouchGuide: Inference-time steering of visuomotor policies via touch guidance. *Preprint*, arXiv:2601.20239.

Sizhe Zhao, Shengping Zhang, Shuo Yang, Weiyu Zhao, Shuigen Wang, and Xiangyang Ji. 2026. Tapsampling: Inference-time sampling with a task-progress-understanding verifier for robotic manipulation. In *International Conference on Machine Learning*.

Enshen Zhou, Qi Su, Cheng Chi, Zhizheng Zhang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and He Wang. 2025a. Code-as-monitor: Constraint-aware visual programming for reactive and proactive robotic failure detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. 2025b. Libero-pro: Towards robust and fair evaluation of vision-language-action models beyond memorization. *Preprint*, arXiv:2510.03827.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Soroush Nasiriany, and Yifeng Zhu. 2020. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*.

A A Functional Taxonomy of Language in Embodied Robotic Systems

Positioning deployment-time adaptation within the broader landscape of language-mediated embodied control requires a functional view of language in robotic systems: when language is introduced, where it enters the system, and what role it plays in shaping behavior. From this perspective, language can operate as a task specification signal, an optimization objective, a planning interface, an online adaptation channel, or a source of semantic priors. This formulation unifies several previously fragmented research threads in language-related policy learning and control, and highlights deployment-time adaptation as a distinct regime in which language is used to revise ongoing behavior rather than to specify a task once at the outset.

Task Conditioning. Language is introduced before execution and serves as a specification of the task or goal. The policy is conditioned directly on both perception and language, typically in the form

$$\pi(a_t \mid o_t, l) \tag{1}$$

where o_t denotes the current observation, l the language instruction, and a_t the action at time t. Under this formulation, language primarily acts as a task-level condition that disambiguates behavior across multiple skills or goals. This setting covers language-conditioned imitation learning, instruction-conditioned control policy, and many Vision-Language-Action (VLA) models.

Optimization Signal. In optimization-oriented approaches, language does not necessarily enter the policy as a direct input, but instead provides an auxiliary signal for learning, such as reward shaping, preference supervision, or behavioral critique. A generic formulation is

$$r = f(o_t, a_t, l) \tag{2}$$

where language helps define the objective optimized by the agent. In this view, language functions as an evaluative or supervisory channel, influencing policy improvement indirectly through the learning signal rather than through direct action generation. This category includes language-based reward modeling, and preference-guided policy optimization.

Planning Interface. In planning-oriented approaches, language does not directly parameterize the low-level control policy, but instead specifies an intermediate decision variable such as a plan, subgoal sequence, or skill graph. A common formulation is

$$p = g(o_t, l, h_t), a_t \sim \pi(o_t, p) \tag{3}$$

where h_t denotes the interaction history, and p denotes the high-level plan or subgoal representation. language primarily operates at the deliberative layer, enabling task decomposition, subgoal generation, and skill sequencing, while low-level execution is delegated to a separate policy. This formulation covers LLM-based planning, language-guided hierarchical control, and skill-library selection.

Deployment-Time Adaptation. Language is introduced during deployment or ongoing execution and serves to revise, constrain, or redirect ongoing behavior. Rather than specifying the task once at the beginning, language modifies the policy input, inference process, planner state, or future action distribution conditioned on the execution history. A generic formulation is

$$\pi^{'} = \text{Adapt}(\pi, o_{1:t}, a_{1:t}, l), \tag{4}$$

or equivalently,

$$a^{'}_{t:T} = \pi(\cdot \mid o_{1:t}, a_{1:t}, h_{1:t}, l), \tag{5}$$

where adaptation may be implemented either by updating the policy parameters or by steering the policy at inference time. The former corresponds to finetuning or online policy update, whereas the latter modifies behavior through context, planner revision, proposal selection, sampler guidance, or predictive evaluation without changing the base policy weights. Language functions as an intervention signal that updates the agent's behavior online, for example by correcting failures, imposing new constraints, refining user intent, or steering the policy under distribution shift. This perspective distinguishes deployment-time adaptation from conventional instruction following: the central problem is no longer how to ground a command before execution, but how to modify behavior once deployment conditions change.

Semantic Prior. In prior-based approaches, language contributes semantic knowledge that augments embodied reasoning without directly specifying actions or optimization targets. This knowledge may encode commonsense, affordances, object semantics, or world knowledge, and can be injected into either planning or control. A generic formulation is

$$z = \phi(l), \pi(a_t \mid o_t, z) \tag{6}$$

where z denotes a language-derived semantic prior. Under this view, language acts as an external source of structured knowledge that can bias policy inference, improve planning, or support generalization in open-world settings. This category includes knowledge-augmented control, affordance reasoning, and language-informed embodied decision making.

B Scope of Literature Coverage.

This survey provides a representative, not exhaustive, overview of methods, datasets, and testbeds related to language-mediated deployment-time adaptation. We emphasize recent 2025–2026 work from major robotics and machine learning venues, such as RSS, CoRL, ICLR, NeurIPS and ICRA, as well as relevant arXiv preprints, while retaining earlier foundational papers that define framing concepts or widely used benchmarks. Representative adaptation methods are summarized in Table 2; Appendix C further summarizes language-conditioned manipulation testbeds and training resources.

C Dataset Support for Generalization vs. Adaptation

C.1 Task Generalization vs. Deployment-Time Adaptation

Task generalization and deployment-time adaptation address different but related questions. Task generalization asks whether a robot can execute unseen task specifications, such as new object combinations, goals, scenes, or language instructions. Deployment-time adaptation asks how the policy or control stack adjusts when user intent, environmental conditions, task requirements, or execution states change during deployment. It can be viewed as an online mechanism for refining task generalization under deployment conditions.

The distinction matters because adaptation is both structurally and functionally policy-dependent. Structurally, the available adaptation mechanisms are constrained by the intervention handles exposed by the policy architecture (Sec. 2). Functionally, adaptation is constrained by the behavioral support of the pretrained policy, including the range of skills, task variations, and language-grounded behaviors it can already express.

Specifically, stronger task generalization can make adaptation easier by providing a broader behavioral space over which feedback can operate. For example, a policy that has learned diverse grasping behaviors may be more easily redirected by feedback such as *"grasp the handle instead"*. Task generalization does not by itself guarantee adaptability. A policy may follow many unseen initial instructions while lacking runtime mechanisms for online preference steering.

C.2 Representative Language-Conditioned Manipulation Resources

Task generalization provides an important substrate for adaptation: diverse language–action data can teach policies task structure, semantic grounding, and embodiment constraints that later support correction, revision, or recovery at test time. We therefore summarize both *evaluation testbeds* and *training datasets*: the former show what kinds of language-conditioned generalization are currently measured, while the latter show what data supports policy learning and VLA pretraining. Together, they expose the gap between pre-execution task generalization and runtime language-mediated adaptation.

Language-conditioned manipulation testbeds. Table 3 summarizes representative testbeds for language-conditioned manipulation. These benchmarks cover long-horizon instruction following, compositional generalization, multimodal prompting, robustness perturbations, and bimanual manipulation. However, language is usually provided before execution as a task specification. They therefore evaluate instruction-conditioned execution and generalization, rather than deployment-time adaptation where language revises goals, corrects behavior, expresses preferences, or guides recovery.

Language-conditioned manipulation training data. Table 4 summarizes representative training-oriented language-guided robot manipulation datasets. These resources provide real, simulated, or hybrid robot data for policy learning, VLA pretraining, and task generalization across objects, scenes, embodiments, and skills. Yet language is mostly used as task conditioning or task/subtask annotation, not runtime feedback. Their adaptation relevance is therefore indirect, motivating future benchmarks that couple deployment-time changes with language feedback and explicit adaptation metrics.

Formulating Benchmark Instances for Deployment-Time Adaptation. Current embodied robot benchmarks often define each instance as an $(l, \text{episode})$ pair, where language is provided *before* execution begins and evaluation centers on task completion. In contrast, a benchmark instance for language-mediated deployment-time adaptation can be formalized as

$$\mathcal{B} = (o_{1:t}, a_{1:t}, e_t, l_t, y^\star), \tag{7}$$

where $(o_{1:t}, a_{1:t})$ denotes the execution history before intervention, e_t is the adaptation trigger at time t, l_t is the mid-execution language feedback, and $y^\star$ is the desired adapted outcome after intervention. This formulation makes explicit that evaluation begins not only from task initialization, but also from the moment when ongoing behavior must be revised in response to a newly introduced event and linguistic signal. The target is therefore not merely eventual task success, but the quality, timeliness, and efficiency of the behavioral revision induced by deployment-time feedback.

Table 2: Representative methods for deployment-time language intervention.

Representative Work	Base Policy / Model	Adaptation Mechanism
Reactive Plan and Behavior Correction (Sec. 2.1)		
Interactive Language (Lynch et al., 2023)	Language-conditioned BC	Conditions behavior on real-time language input during execution.
Correcting Robot Plans (Sharma et al., 2022)	Task and motion planner	Converts language feedback into residual costs for revising robot plans.
YELL At Your Robot (Shi et al., 2024)	Language-conditioned policy	Uses language feedback during failures for post-training command-space adaptation.
AHA (Duan et al., 2025)	VLM-based failure reasoner	Detects and reasons over manipulation failures, providing semantic feedback for failure-aware recovery.
Code-as-Monitor (Zhou et al., 2025a)	VLM-based monitor	Uses constraint-aware visual programs to detect execution failures and trigger reactive or proactive correction.
FailSafe (Lin et al., 2025)	VLA recovery model	Reasons about VLA failures and produces executable recovery actions during deployment.
Sentinel-VLA (Li et al., 2026a)	Metacognitive VLA	Monitors execution status and invokes dynamic reasoning for error recovery and self-correction.
Trajectory Editing (Sec. 2.2)		
LATTE (Bucker et al., 2023)	Trajectory representation	Translates language commands into trajectory deformations over waypoints or motion traces.
OVITA (Maurya et al., 2025)	Waypoint trajectory	Uses open-vocabulary language and code-like edits to adapt instantiated trajectories.
Continuous Guidance (Sec. 2.2)		
ITPS (Wang et al., 2025)	Diffusion policy	Injects interaction-derived objectives into sampling to steer trajectory generation.
VLS (Liu et al., 2026)	Pretrained diffusion policy	Uses VLM-derived rewards or costs as sampler-level guidance.
Compose Your Policies (Cao et al., 2026)	Diffusion / flow policy	Composes pretrained policy distributions at test time to steer trajectory generation without retraining.
Diffusion Policy Steering (Wagenmaker et al., 2025)	Diffusion policy	Optimizes latent noise variables to adapt generated trajectories while keeping the base policy fixed.
Action Selection (Sec. 2.2)		
V-GPS (Nakamoto et al., 2024)	Robotic foundation model	Reranks candidate actions using value guidance.
FOREWARN (Wu et al., 2025)	VLA / latent alignment model	Uses VLM-in-the-loop verification to select policy outputs aligned with future consequences.
SEAL (Wu et al., 2026)	VLA / simulation	Verifies reasoning–action alignment at runtime and selects aligned actions.
UPS (Yuan et al., 2026b)	Pretrained policy with verifier	Calibrates VLM–policy uncertainty to choose among execution, language clarification, and action intervention.
RoboMonkey (Kwok et al., 2025)	VLA with verifier	Scales test-time action sampling and selects candidates using VLM-based verification.
TapSampling (Zhao et al., 2026)	VLA with progress verifier	Samples candidate actions at inference time and selects them with a task-progress-understanding verifier.
Verifier-free TTS (Jang et al., 2026)	VLA	Selects among sampled candidate actions using internal confidence signals rather than an external verifier.
VLA-ATTC (Li et al., 2026b)	VLA with action critic	Allocates adaptive test-time compute and selects actions using a relative action critic.
World-Model-Based Adaptation (Sec. 2.3)		

Continued on next page

Representative Work	Base Policy / Model	Adaptation Mechanism
Semantic World Model (Berg et al., 2026)	VLM-based semantic world model	Searches or evaluates imagined action-conditioned futures for semantic adaptation.
Phys2Real (Wang et al., 2026)	VLM-guided digital twin	Uses language-inferred physical priors and imagined rollouts for online adaptation.
AdaPolicy (Yuan et al., 2026a)	World-model-based policy	Adapts policy behavior by evaluating predictive rollouts under deployment-time conditions.
OSVI-WM (Goswami et al., 2025)	World-model-guided trajectory generator	Predicts latent state–action sequences and decodes them into waypoints for test-time task transfer.
Embodiment Constraints and Adaptation Targets (Sec. 3)		
UF-OPS (Attarian et al., 2026)	Diffusion policy; real bi-arm ALOHA tasks	Uses update-free verifier-based steering for bi-arm manipulation; although not language-mediated, it provides a relevant example of deployment-time steering in richer embodiments.
BiSAIL (Li et al., 2026c)	Offline-learned bimanual skill policy	Studies online interactive adaptation of bimanual skills from verbal feedback, grounding language into spatial coordination of dual-arm motion.
SafeBimanual (Deng et al., 2025)	Bimanual diffusion policy	Applies safety-constrained test-time trajectory optimization for bimanual manipulation, highlighting embodiment-specific constraints such as inter-arm coordination and collision avoidance.
TouchGuide (Zhang et al., 2026)	Visuomotor policy; contact-rich manipulation	Performs inference-time steering with tactile-guided feasibility signals, showing that adaptation may involve contact and touch rather than only visual geometry.

Table 3: Representative language-conditioned manipulation testbeds, highlighting their evaluation focus. Source types are abbreviated as R = real robot and S = simulation.

Benchmark	Embodiment / Setting	Language Use and Evaluation Focus
CALVIN (Mees et al., 2022)	S; single-arm tabletop manipulation.	Long-horizon task instructions over reusable atomic skills; evaluates sequential task execution without online feedback.
LIBERO (Liu et al., 2023)	S; single-arm tabletop; object, scene, goal, and spatial variations.	Task instructions; evaluates language-conditioned generalization without deployment-time feedback protocols.
LIBERO-PRO (Zhou et al., 2025b)	S; LIBERO-based perturbations over objects, states, instructions, and environments.	Task instructions under distribution shifts; evaluates robustness while treating language as pre-execution input.
Language Table (Lynch et al., 2023)	R; planar tabletop manipulation and object rearrangement.	Spatial rearrangement instructions; evaluates spatial grounding, compositional generalization.
Language-World (Zentner et al., 2024)	S; controlled manipulation environment with task and scene variations.	Task instructions on goal, object, and relation; evaluates language-grounded task generalization.
VIMA-Bench (Jiang et al., 2023)	S; tabletop manipulation with multimodal prompt templates.	Evaluates prompt-conditioned generalization.
VLABench (Zhang et al., 2024)	S; large-scale language-conditioned manipulation with long-horizon tasks.	Language instructions involving world knowledge; evaluates VLA reasoning and generalization.
RoboTwin 2.0 (Chen et al., 2025)	S; bimanual; randomized scenes, objects, lighting, and tasks.	Foundation model generated tasks and language randomization; evaluates bimanual generalization.

Table 4: Representative training-oriented language-guided manipulation datasets. These resources primarily support policy training, pretraining, and task generalization, while their relevance to deployment-time adaptation is mostly indirect because they do not define online language-feedback protocols. Source types are abbreviated as R = real robot, M = mobile data, and H = human-centric data.

Dataset	Data Source	Language Annotation	Generalization vs. Adaptation
		Single-arm	
BridgeData V2 (Walke et al., 2023)	R; diverse environments and skills.	Task instructions.	Task generalization; no online correction protocol.
DROID (Khazatsky et al., 2024)	R; diverse scenes, viewpoints, objects, and tasks.	Task descriptions.	Scene and task generalization; no online correction protocol.
RH20T (Fang et al., 2023)	R+H; multimodal data with vision, force, audio, action, and paired human demos.	Task descriptions.	Force/contact-aware generalization; no online correction protocol.
RH20T-P (Chen et al., 2024)	R; robot clips with primitive-level annotations.	Primitive-level annotations.	Useful for sequence-level generalization and recovery analysis; no online feedback protocol.
		Bimanual	
MolmoAct2-BimanualYAM (Allen Institute for AI, 2026)	R; teleoperated bimanual manipulation tasks.	Task descriptions.	Task generalization; relevant to adaptation through bimanual coordination and task-correction potential.
RoboMIND 2.0 (Hou et al., 2025)	R+M; diverse tasks and scenarios.	Task instruction decomposition and planner/executor-oriented annotations.	Long-horizon task generalization; potentially useful for task-state or sequence-level adaptation.
AgiBot World (AGIBOT WORLD, 2026)	R+M; diverse tasks and scenarios.	Task/action-text annotations, action slices, and instructions.	Task generalization; adaptation relevance depends on explicit feedback or revision protocols.