# PRAXISWORLD: OBJECT-CENTRIC WORLD-ACTION MODEL FOR COMPOSABLE ROBOT MANIPULATION

Yu-song HUANG[1,†], Yu-yan LI[1,†], Hong-liang LU[2], Hao-tian WANG[1], Xiao-yun QIU[1], Wen-peng XU[1], Yang-gang SHENG[1], Zhao-nian KUANG[1], Yi-jie CHEN[1], Jin-tao HE[1], Dong-ling LIU[1], and Xin-hu ZHENG[1,*]

[1]Intelligent Transportation Thrust, Systems Hub, and Center of Seamless Connectivity & Connected Intelligence, Hong Kong University of Science and Technology (Guangzhou), Guangzhou 511453, China

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

[†]These authors contributed equally to this work.

[*]Corresponding author: xinhuzheng@hkust-gz.edu.cn

**ABSTRACT**

Robot manipulation remains difficult not because robots lack action labels such as *pick*, *place*, and *insert*, but because these labels under-specify the world state—object configuration, contact mode, metric tolerance, termination condition, and expected effect—on which a physical interaction succeeds. The bottleneck is therefore not the *quantity* of action labels in training data, but the *world-state understanding* a system brings to physical interaction. This survey develops **PraxisWorld**, an object-centric **world-action model** that recasts manipulation from a direct *language-to-action* mapping into a unified *world state – action interface – state transition* paradigm. PraxisWorld rests on four commitments: objects as anchors, world state as the substrate, atomic skills as executable interfaces, and state transition as the verifiable basis that links perception, prediction, planning, control, and recovery. In this view a world model does not merely predict future pixels or latent states; it explicitly represents object-centric world-state transitions through a unified atomic-skill framework, revealing whether a method exposes verifiable physical interfaces or hides them inside an end-to-end monolithic policy. We formulate an object-centric skill descriptor $K = (R, S, X, C, I, \pi, \beta, \Delta, E)$ as the executable interface of this model, and review more than 180 works through four stages: definition and discovery, learning and anchoring, composition and orchestration, and execution and recovery. Composition then becomes verifiable state-transition matching rather than language-level concatenation: a chain is physically executable only when the object state produced by one skill satisfies the initiation condition of the next ($\Delta_i \models I_j$). We analyze JEPA-style and action-conditioned world models as concrete instances of this transition interface, include a small Meta-World diagnostic probe of object-state interface gates, and identify open problems in learning compositional interfaces, unifying semantic/metric/tactile state, modeling contact-rich transitions, and benchmarking the full object-centric skill chain. The aim is to move manipulation from behavior imitation toward composable, verifiable, and recoverable physical interaction.

***K*eywords** Robot manipulation · World-action model · Object-centric perception · Atomic skills · State-transition interfaces · Skill composition · Vision-language-action models · World models · Survey

## 1 Introduction

Embodied AI becomes consequential when an agent can act in the physical world, and robot manipulation is the setting where this requirement is most unforgiving. A household, service, or industrial robot must not only parse instructions and recognize objects, but also establish contact, regulate force, maintain object identity through occlusion, and verify whether an intended state change has occurred. The everyday tasks that motivate embodied intelligence, such as preparing food, packing groceries, assembling furniture, sorting laundry, and performing laboratory procedures, are

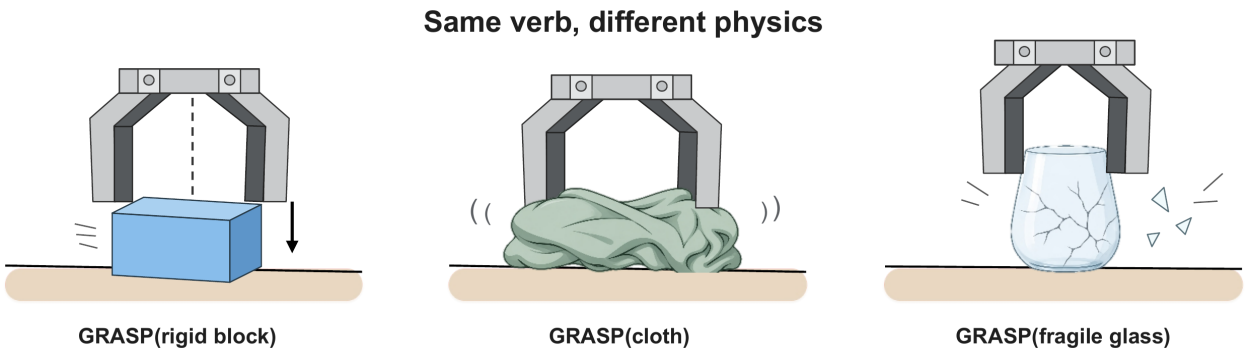


Fig. 1: **Same verb, different physics.** The semantic label *grasp* hides different contact topologies, force regimes, and failure modes across a rigid block, deformable cloth, and fragile glass. PraxisWorld treats the object-state transition behind the verb, rather than the verb alone, as the unit of definition, learning, composition, and verification: each skill is read as a world interface that consumes and produces object state.

therefore not single behaviors. They are organized sequences of smaller physical interactions whose validity depends on the objects being acted upon.

Modern robot learning has made striking progress on short-horizon and language-conditioned behavior. Vision-language-action (VLA) models transfer web-scale semantic knowledge to robot control (Brohan et al., 2023a), flow-matching action heads support broad visuomotor policies (Black et al., 2024a), and recent generalist policies show impressive cross-task and cross-embodiment behavior (Physical Intelligence et al., 2025, 2026). As scaling laws reshaped language and vision, it is tempting to expect that scaling action-labeled demonstrations will, by itself, deliver reliable manipulation. Yet a persistent gap remains between producing a plausible action and manipulating reliably across object instances, contact regimes, perturbations, and long horizons. The issue is not only model capacity or data volume. A semantic label such as *grasp*, *place*, or *insert* does not specify the world state in which the action is valid, the contact mode it must establish, the tolerance it must satisfy, or the object-state effect on which the next action depends. Fig. 1 illustrates this under-specification for a single verb: the label stays fixed while the required contact regime, force profile, and failure mode change with the object. The bottleneck, in other words, is less the *quantity* of action labels than the *world-state understanding* a system can bring to each interaction.

Human behavior offers a useful way to frame this gap. People usually do not solve complex physical tasks as flat streams of motor commands. They decompose goals into subgoals, reuse familiar interaction units, monitor intermediate states, and repair local failures before continuing. Cognitive models of resource-rational planning make this observation precise: people choose decompositions that trade off task utility against the cost of planning (Callaway et al., 2022; Correa et al., 2023). At the perceptual-motor level, human interaction is also object-centered: objects are tracked as persistent entities (Spelke, 1990; Kahneman et al., 1992; Pylyshyn, 2001), possible actions are constrained by affordances (Gibson, 1979), and motor skill improves as perception, contact, and correction become tightly coupled (Fitts and Posner, 1967). This is not a claim that robots should copy human cognition directly. It motivates a structural principle: reusable manipulation units should be grounded in persistent objects and their physical transitions, not only in verbs.

This survey develops that principle into a world model for action. We call the resulting framework *PraxisWorld*: an object-centric *world-action model* that recasts manipulation from a direct *language-to-action* mapping into a unified paradigm of *world state – action interface – state transition*. PraxisWorld rests on four commitments. Objects are the *anchors* to which interaction is bound; world state, decomposed into semantic, metric, and tactile/contact attributes, is the *substrate* on which transitions are defined; atomic skills are the *executable interfaces* through which an agent reads and writes that state; and state transition is the *verifiable basis* that connects perception, prediction, planning, control, and recovery. The name is deliberate: *praxis* is informed, purposeful action, and a manipulation system becomes reliable only when knowing, embodied in the world model, and acting, embodied in the skill interface, are the same object rather than two loosely coupled modules.

The executable interfaces in PraxisWorld are what we call *atomic skills*. An atomic skill is not simply the shortest possible motion segment or the smallest language token. It is a reusable object-conditioned transition that a planner can call, a controller can execute, and a verifier can check. The same verb *grasp* denotes different physical problems for a rigid block, a deformable cloth, a plastic bag, and a fragile beaker; *insert* changes character across loose pegs, tight connectors, and flexible cables; *place* differs radically between a broad tabletop region and a high-precision fixture.

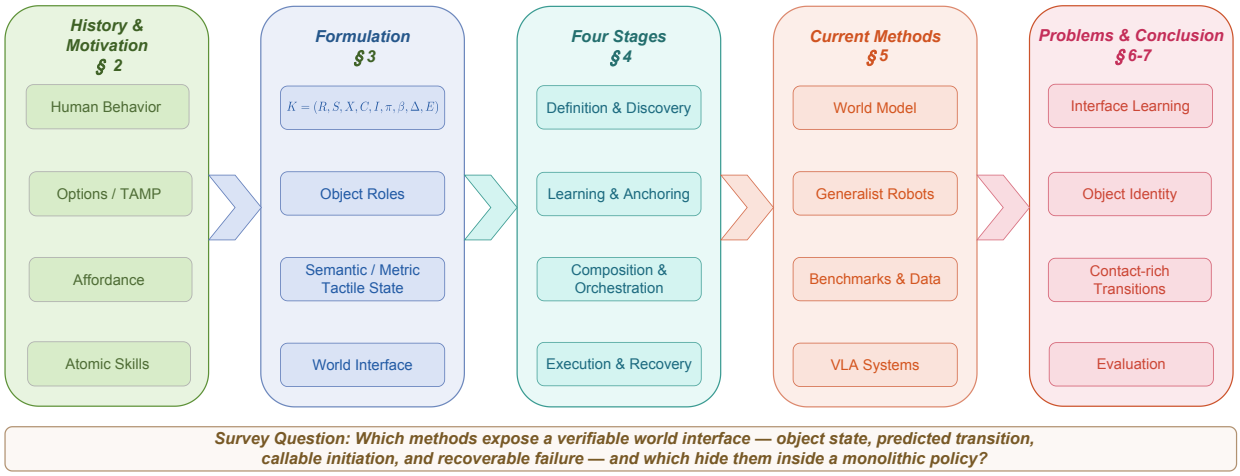


Fig. 2: **PraxisWorld at a glance.** The survey is organized as a world-action model rather than as a catalog alone: human behavioral organization motivates the object-centric paradigm; the descriptor makes the world interface explicit; the four-stage lifecycle of definition and discovery, learning and anchoring, composition and orchestration, and execution and recovery turns world state into callable, verifiable transitions; and recent world-model methods are analyzed as estimators of the missing transition model $\Delta$.

These differences live in object state, geometry, material, contact topology, force, tolerance, and embodiment. They cannot be recovered from the verb alone.

The central claim of this survey is that atomic skills should be represented as object-centric world interfaces with explicit composition semantics. By *object-centric*, we mean that a skill is bound to persistent object instances and conditioned on semantic, metric, and tactile/contact state. By *world interface*, we mean that the skill exposes when it can start, how it terminates, and what object-state effect it is expected to produce, so that the world model can predict that effect and a verifier can check it. Composition then stops being the concatenation of action names and becomes *verifiable state-transition matching*: the predicted effect of one skill must satisfy the initiation condition of the next. This effect-to-initiation test, written $\Delta_i \models I_j$ throughout the survey, is the difference between a language-level sequence such as "pick then insert" and a physically valid manipulation chain. In short, PraxisWorld does not rename actions; it gives actions a world interface.

This positioning also clarifies the type of survey we aim to write. The paper is not merely a catalog of works that already use the term *atomic skill*, nor is it a method paper claiming a new implemented system. It is a framework survey: PraxisWorld is a unifying lens for comparing several lines of manipulation research that are often discussed separately, including options, task and motion planning, affordances, movement primitives, VLAs, tactile manipulation, and world models. For each method family we ask which parts of the world-action interface it exposes, hides, or leaves unlearned, and whether those parts are sufficient for reliable skill composition.

The survey makes four contributions. First, it reframes the literature around the evolution from monolithic policies, symbolic operators, and affordance models toward an object-centric world-action model, in which skills are interfaces to predicted world-state transitions. Second, it formulates a compact descriptor $K = (R, S, X, C, I, \pi, \beta, \Delta, E)$ that serves as the executable interface of this model, making object role bindings, skill type, semantic/metric/tactile state, contact topology, initiation, policy, termination, effect, and embodiment explicit. Third, it reviews more than 180 works through the four stages of the PraxisWorld lifecycle: how skills are *defined and discovered*; how they are *learned and anchored* to object state; how they are *composed and orchestrated* into task-oriented chains; and how *execution and recovery* close the loop. Fourth, it analyzes recent JEPA-style and action-conditioned world models, including LeWorldModel, V-JEPA 2-AC, WorldVLA, VLA-JEPA, and AtomVLA, as concrete instances of the world-action model that supply the predicted effect $\Delta$, and includes a small Meta-World diagnostic probe that tests whether an explicit object-state gate can detect and recover from perturbed skill-boundary transitions (Maes et al., 2026; Assran et al., 2025; Cen et al., 2025; Sun et al., 2026a,b). Fig. 2 summarizes how these contributions map onto the rest of the survey and highlights the world-action question that connects the sections.

The rest of the survey is organized as follows. Section 2 explains why robot manipulation has developed toward an object-centric world-action model. Section 3 defines the model and its atomic-skill interface, and compares the

formulation to options, TAMP, affordances, and VLAs. Section 4 reviews the four-stage skill lifecycle. Section 5 interprets current methods, systems, benchmarks, and world models as transition-interface estimators through this framework, and reports a small simulation probe of object-state interface gates. Section 6 identifies open problems and testable hypotheses, and Section 7 concludes.

## 2 History and Motivation: From Behavioral Organization to a World-Action Model

The idea of an object-centric world-action model did not appear fully formed. It is the result of several traditions trying to solve the same manipulation problem at different levels of abstraction: how to turn continuous interaction with objects into reusable, verifiable units that can support planning, learning, and recovery. This section traces that evolution, not as a chronology of names, but as a sequence of pressures that pushed the field from action labels toward skill interfaces defined over world state.

### 2.1 Behavior as Decomposition and Reuse

Human task performance is flexible because behavior can be decomposed and recomposed. In everyday physical tasks, a person does not solve every motion from scratch; they reuse units such as reaching, grasping, aligning, pouring, wiping, fastening, and checking. The relevant unit is not always a fixed motor program. It depends on task complexity, object properties, and planning cost. For simple interactions, behavior may appear nearly direct and reactive. For complex manipulation, decomposition becomes explicit: subgoals are selected, intermediate states are monitored, and local failures trigger repair.

Recent cognitive models make this intuition precise. Resource-constrained planning work argues that exhaustive lookahead is infeasible for humans, so people adopt planning strategies that balance performance with cognitive cost (Callaway et al., 2022). Task-decomposition work further shows that people choose subgoals that reduce planning cost while preserving task utility (Correa et al., 2023). For robotics, the lesson is not that a robot should imitate a specific human algorithm. It is that long-horizon behavior needs a structured intermediate level between raw actions and task goals. Atomic skills, understood as the executable interfaces of a world-action model, occupy that level.

### 2.2 From Temporal Abstraction to Physical Interfaces

Hierarchical reinforcement learning introduced temporal abstraction as a formal answer to this need. The options framework defines an option by an initiation set, an intra-option policy, and a termination condition (Sutton et al., 1999). Option-Critic showed that such components can be learned end-to-end (Bacon et al., 2017). This line gives atomic skills a crucial formal property: a skill is not merely a segment of action but a callable unit with boundaries.

However, generic options do not by themselves solve manipulation. An initiation set represented in abstract state space does not necessarily encode whether a cup is graspable, whether a connector is aligned, whether contact is stable, or whether the available gripper can realize the required force. The move from options to object-centric skills is therefore a move from temporal abstraction to physical interfaces defined over world state. The skill must specify the object state and contact regime under which the option is meaningful.

### 2.3 From Symbolic Operators to Perceptual Grounding

Task and motion planning (TAMP) supplies the complementary view. Classical operators specify preconditions, effects, and geometric constraints, and a plan is valid when each effect satisfies the next precondition (Garrett et al., 2021; Zhao et al., 2024). This is exactly the discipline that long-horizon manipulation needs, and it is the discipline that a world-action model must recover at the level of perceived state. A symbolic *pick* operator can create the condition *holding(object)*, which enables a later *place* or *insert* operator.

The limitation is that hand-coded symbolic predicates rarely match the perceptual and contact uncertainty of learned manipulation. The precondition *aligned(peg,hole)* is not a single Boolean fact; it depends on pose uncertainty, clearance, compliance, chamfer geometry, force limits, and tactile evidence. Object-centric atomic skills inherit the precondition–effect logic of TAMP but require these interfaces to be grounded in multimodal world state rather than only in symbolic labels.

### 2.4 From Affordances to Object-State Transitions

Affordance and Object–Action Complex (OAC) research already moved toward this view by tying actions to object properties and predicted effects. Affordances describe possibilities for action relative to the agent and environment

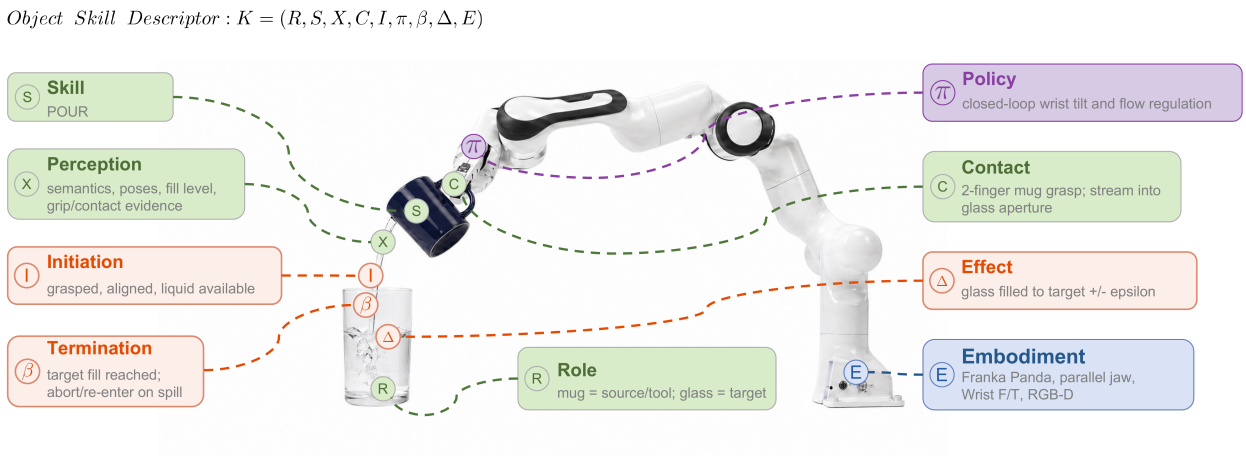


Fig. 3: **The atomic-skill interface of the world-action model.** The descriptor turns a skill from a semantic action name into a physically grounded world interface. Perception estimates semantic, metric, and tactile/contact world state; the policy acts under embodiment constraints; termination monitors success, failure, abort, or re-entry; and the predicted effect $\Delta$ is the world-state transition that the next skill's initiation interface $I$ must accept.

(Gibson, 1979), and robotic affordance learning models object–action–effect triples from interaction (Montesano et al., 2008). OACs bind objects, actions, effects, and verification into reusable sensorimotor units (Krüger et al., 2011). These ideas are close conceptual ancestors of the object-centric world-action model: they already treat the predicted object-state effect, rather than the action label, as the thing worth representing.

What has changed is the scale and architecture of modern robot learning. Large VLAs, diffusion policies, tactile foundation models, and world models can learn powerful policies and representations from broad data (Brohan et al., 2023a; Chi et al., 2023; Huang et al., 2025a; Cen et al., 2025). But unless their learned units expose object-state transitions, they remain difficult to compose, verify, debug, or transfer. The modern problem is therefore not to abandon affordances or OACs, but to reconnect their object-action-effect discipline with current neural policies inside an explicit world-action model.

### 2.5 Why a World-Action Model Now?

An object-centric world-action model has become timely because three trends now meet. First, generalist policies provide broad semantic and visuomotor competence, making reusable skills plausible at scale—from early robotics transformers and open generalist policies (Brohan et al., 2023b; Octo Model Team et al., 2024; Kim et al., 2024) to flow-based VLAs (Brohan et al., 2023a; Black et al., 2024a; Physical Intelligence et al., 2025). Second, multimodal grounding methods increasingly expose the metric and tactile variables that semantic labels omit, turning "world state" from a slogan into measurable quantities (Qu et al., 2025; Huang et al., 2025a; Yu et al., 2025b). Third, world models, including JEPA-style predictive architectures, provide a natural mechanism for estimating the effect of candidate actions or skills before execution (Cen et al., 2025; Assran et al., 2025; Maes et al., 2026; Sun et al., 2026a). These three trends are now crystallizing into an explicit *World Action Model* literature that unifies state prediction with action generation, and that has already produced its first surveys (Wang et al., 2026; Zhang et al., 2025d).

The convergence suggests a revised survey question. Rather than asking only which architecture performs best on a benchmark, we ask which parts of the object-centric world interface each method makes explicit: the world state it conditions on, the policy it executes, the initiation and termination interfaces it exposes, the transition it predicts, and the recovery path it enables when the observed effect does not match the intended one.

## 3 Formulation: The Object-Centric World-Action Model

This section defines the object-centric world-action model and the atomic-skill descriptor that serves as its executable interface. The model has two coupled parts: a world state defined over persistent object instances, and a skill interface that reads that state, acts on it, and predicts the resulting transition. It is a survey formulation rather than a claim that all systems must implement the same data structure. Its purpose is diagnostic: it reveals which parts of the world interface

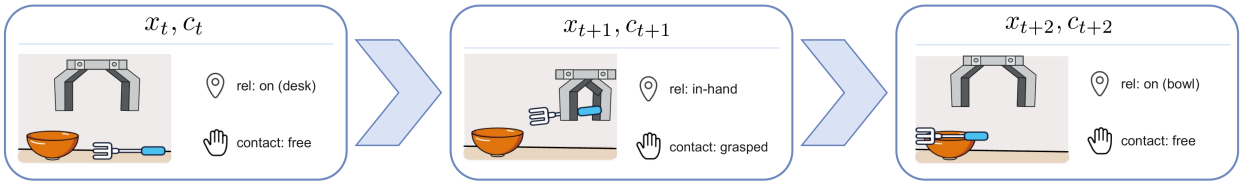


Fig. 4: **World-state transitions under skill execution.** A skill's effect $\Delta$ is read out as a transition over semantic-metric object state $x$ and contact topology $c$. For pick-and-place, the object may move from a free state on a support surface, to an in-hand grasped state, and then to a released state on a target object. The post-state of one skill is the world state against which the next skill's initiation interface $I$ is tested.

are explicit, learned, latent, or absent in each method family. Fig. 3 is the visual anchor for this section: it shows how perception, policy, embodiment, termination, and predicted effect are tied together by the descriptor.

### 3.1 The Object Skill Descriptor

**Definition 1** (Object-centric atomic skill)**.** An object-centric atomic skill is a reusable object-conditioned transition module

$$K = (R, S, X, C, I, \pi, \beta, \Delta, E),$$
$$X = (X_{\text{sem}}, X_{\text{met}}, X_{\text{tac}}). \tag{1}$$

Here $R$ binds object instances to roles; $S$ is the skill type; $X$ is the object state decomposed into semantic, metric, and tactile/contact attributes; $C$ is contact topology; $I$ is the initiation interface; $\pi$ is the closed-loop policy; $\beta$ is the termination and re-entry function; $\Delta$ is the expected object-state effect; and $E$ specifies embodiment.

**Object and state grounding.** $R : \mathcal{O} \to \mathcal{L}_{\text{role}}$ maps concrete object instances to roles such as *source*, *target*, *tool*, or *container*. $S$ is the skill type, such as *grasp*, *insert*, *pour*, or *wipe*. $X_{\text{sem}}$ captures category, function, and language-level affordance; $X_{\text{met}}$ captures pose, geometry, spatial relation, clearance, and tolerance; and $X_{\text{tac}}$ captures force, friction, compliance, slip, and material/contact evidence. For articulated objects, $X_{\text{met}}$ must also encode part-wise pose, joint state, and kinematic constraints; category-level articulated pose perception methods such as CAPER++ make these variables explicit through a root-part/constrained-part hierarchy and SE(3) manifold tracking (Zhang et al., 2026b). $C$ describes contact topology, including contact patch, normals, constraints, and mode changes (Cutkosky, 1989).

**Interfaces.** $I$ is the initiation interface: the object, contact, and embodiment conditions under which the skill can be called reliably. $\beta$ is the termination and re-entry function: it determines success, failure, abort, retry, or re-entry into another skill. $\Delta$ is the expected effect model, predicting the distribution of post-states over object state and contact topology. The triple $(I, \beta, \Delta)$ is the core of composability. Without it, a learned policy may act well in isolation but cannot be safely chained or repaired. Fig. 4 shows the same idea on a concrete state stream: each skill is read not only as a command, but as a transition from one object/contact state to the next.

**Policy and embodiment.** $\pi$ is the closed-loop policy conditioned on object state, contact state, history, and parameters. $E$ specifies the manipulator, gripper, sensor suite, controller class, and embodiment constraints. The same object-state transition may transfer across embodiments, but the policy and initiation set often change when the body, gripper, or sensors change.

### 3.2 Atomicity as a Physical Criterion

Atomicity is not unique. A segment can be atomic under one criterion and composite under another. A *pick* can be atomic as a language token, but it may include approach, contact, grasp closure, lift, slip correction, and verification. An insertion can be atomic as a planner operator, but it includes alignment, compliant search, contact, seating, and force confirmation. A diffusion action chunk can be atomic as a trajectory sample, but it may hide which object-state predicate has changed (Chi et al., 2023; Zhao et al., 2023).

The object-centric criterion resolves this ambiguity operationally:

> An atomic skill is a single reusable object-conditioned state transition whose initiation, termination, and effect are separable, identifiable, and reusable across tasks.

Table 1: Axes for characterizing atomicity in manipulation skills, plus the object-centric view adopted in this survey.

| Axis | Definition of atomicity | Sem. | Tact. | Met. | Typical use |
|---|---|---|---|---|---|
| Contact mode | A skill is atomic within a single contact configuration or contact-mode segment. | Partial | Yes | Partial | Hybrid dynamics and contact-rich manipulation (Cutkosky, 1989; Garrett et al., 2021) |
| Gripper state | A skill spans an open–close or close–open gripper-state interval. | No | Partial | No | Demonstration segmentation and boundary detection (Gutierrez and Beksi, 2026) |
| Object relation | A skill changes one relation such as *on*, *inside*, or *attached*. | Yes | No | Partial | Scene-graph planning and symbolic effects (Garrett et al., 2021) |
| Planner operator | A skill is one symbolic operator with preconditions and effects. | Yes | No | Partial | TAMP and classical planning (Garrett et al., 2021; Zhao et al., 2024) |
| Learned token | A skill is a discrete code or latent token from trajectories. | Implicit | No | Implicit | VLA tokenization and learned skill libraries (Brohan et al., 2023a; Ma et al., 2024) |
| Semantic label | A skill is a verb or verb-object phrase such as *pick mug*. | Yes | No | No | Language planning and instruction following (Brohan et al., 2023a; Haresh et al., 2024) |
| Process spec. | A skill is a process with temporal, force, or state-change constraints. | Partial | Yes | Yes | Assembly, insertion, wiping, and pouring (Krüger et al., 2011; Garrett et al., 2021) |
| **Object-centric** | **A skill is an object-conditioned transition module whose identity depends on object attributes, contact, interfaces, effects, and embodiment.** | **Yes** | **Yes** | **Yes** | **Object-state transformation, interface matching, verification, and recovery** |

This definition does not force all skills to be short. It forces them to expose a transition that can be tested. If a candidate unit lacks an observable initiation condition, a termination condition, or an effect model, it may still be useful for imitation or compression, but it is not yet a composable object-centric skill.

Table 1 restores an important distinction: the literature calls many different units "atomic", but under different criteria. This survey adopts the object-centric criterion because it is the one that makes downstream composition and verification testable.

### 3.3 Relation to Existing Formalisms

Table 2 previews the comparison by asking which components of $K$ each family usually makes explicit; the paragraphs below explain the trade-offs behind the entries. We assign the marks by a consistent rule: • means the component is named and used as an explicit, inspectable variable in the method's interface, loss, or planning state; ◦ means it appears only implicitly, as a learned or latent representation, or only for some tasks; and – means it is absent or delegated to an external module. The marks describe the *typical* form of each family rather than any single system, and are intended as a diagnostic reading aid rather than a quantitative score.

**Options** provide $(I, \pi, \beta)$ and thus formalize temporal abstraction, but they typically omit persistent object variables, contact topology, effect models, and embodiment constraints (Sutton et al., 1999; Bacon et al., 2017). Object-centric skills can be read as physically enriched options whose initiation and termination are stated over object state and contact.

**TAMP operators** provide preconditions, effects, and geometric constraints (Garrett et al., 2021; Zhao et al., 2024). They offer the clearest composition discipline, but their predicates are often hand-designed and symbolic. Object-centric skills preserve the precondition–effect idea while asking how such interfaces can be learned, perceived, and verified. PDDLStream and related approaches make the same point operationally: symbolic skeletons become useful only when streams or samplers can certify the continuous geometric conditions that make an operator callable (Garrett et al., 2020).

**OACs and affordances** bind objects, actions, and effects (Krüger et al., 2011; Montesano et al., 2008). They are the closest conceptual ancestor. The present formulation extends this tradition with modern multimodal perception,

Table 2: Representative method families compared by object-centric skill components. •: usually explicit; ○: partial, implicit, or task-dependent; –: usually absent or external.

| **Family** | $X$ | $\pi$ | $I$ | $\beta$ | $\Delta$ | $E$ | **Main gap** |
|---|---|---|---|---|---|---|---|
| Options / HRL | ○ | • | • | • | ○ | ○ | Object state and effects are weak or latent |
| TAMP | • | ○ | • | • | • | ○ | Learned perception and policy robustness are limited |
| OAC / affordances | • | ○ | ○ | ○ | • | ○ | Less connected to modern VLA and diffusion policies |
| Semantic VLAs | ○ | • | ○ | ○ | ○ | ○ | Geometry, contact, termination, and effects are hidden |
| Spatial VLAs | • | • | ○ | ○ | ○ | ○ | Metric grounding improves, but interfaces remain implicit |
| Tactile VLAs | • | • | ○ | ○ | ○ | ○ | Contact grounding improves, but coverage is hardware-limited |
| Diffusion policies | ○ | • | ○ | ○ | ○ | ○ | Skill boundaries and effects are usually latent |
| Movement primitives | ○ | • | ○ | ○ | ○ | ○ | Object semantics and composability are often external |
| World models | ○ | ○ | ○ | ○ | • | ○ | Predictive effects exist, but often as pixels or latent states rather than object predicates |

learned policies, tactile grounding, cross-embodiment transfer, and VLA/world-model integration. Beyond a change of application domain, the descriptor adds three structural commitments to the classical OAC: it factors world state into separable semantic, metric, and tactile/contact components with an explicit contact topology $C$, rather than a single attribute set; it treats the effect-to-initiation link as a *probabilistic, learnable* compatibility test $\Delta_i \models I_j$ (Eq. 4) instead of a symbolic precondition match; and it represents recovery as re-entry into the same interface graph (Eq. 7) rather than as separate error handling. These commitments are also what keep the model falsifiable in practice: a method either exposes a testable $\Delta \to I$ relation or it does not, and the diagnostic probe in Section 5 shows that this distinction is measurable rather than merely descriptive.

**VLAs and diffusion policies** provide powerful $\pi$ but usually hide $I$, $\beta$, and $\Delta$ inside learned representations (Brohan et al., 2023a; Chi et al., 2023; Black et al., 2024a). The formulation is useful precisely because it distinguishes behavioral competence from exposed compositional structure.

**Movement primitives and learned action chunks** provide compact executable policies—from dynamical and probabilistic movement primitives (Ijspeert et al., 2013; Paraschos et al., 2013) to learned action chunks (Zhao et al., 2023)—but often treat object semantics, contact verification, and chain-level effects as external annotations rather than as part of the primitive itself (Gutierrez and Beksi, 2026).

## 4 The Four-Stage Lifecycle of the World-Action Model

The descriptor in Section 3 becomes useful when connected to the lifecycle of a manipulation system. We organize that lifecycle into four stages: defining and discovering the skill, learning and anchoring it to object state, composing and orchestrating it with other skills, and closing the loop through execution and recovery. The four stages are not separate topics; they are one pass through the same object-centric world-action model, traced from world state to verified transition. Fig. 5 follows the lifecycle from input state and embodiment through skill discovery, anchoring, composition, and closed-loop execution and recovery.

### 4.1 Stage I: Definition and Discovery

Skill definition asks what counts as an atomic skill. Skill discovery asks how such units are found from language, demonstrations, interaction, or cross-embodiment data. The two questions are coupled but distinct. A segmentation algorithm can cut a trajectory into neat pieces without discovering reusable skills; a language model can propose plausible step names without grounding them in physical transitions.

In the object-centric view, a candidate unit qualifies as a skill only if four criteria hold. First, it must have **transition separability**: executing the unit changes object state $X$ or contact topology $C$ in a measurable way. Second, it must have **interface testability**: initiation and termination can be checked at runtime. Third, it must have **parameter identifiability**: variation across instances can be explained by roles, geometry, force, tolerance, or embodiment. Fourth, it must have **reuse potential**: the transition recurs across objects, scenes, tasks, or embodiments.

Language-guided methods provide strong proposal mechanisms. Systems such as AAS, Manual2Skill, DeCo, and Gentle Manipulation use language or VLMs to identify candidate steps and skill hierarchies (Tabakov et al., 2026;

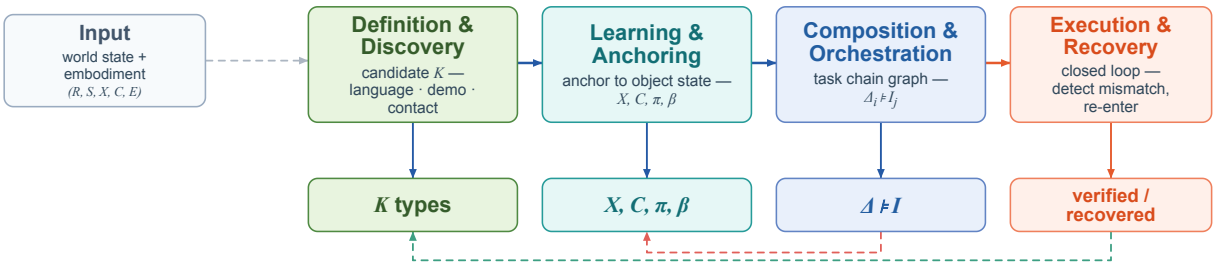


Fig. 5: **Four stages of the object-centric world-action model.** The framework merges skill definition and discovery, learning and anchoring, composition and orchestration, and execution and recovery into a single lifecycle. The descriptor $K = (R, S, X, C, I, \pi, \beta, \Delta, E)$ connects the stages, while the interface test $\Delta_i \models I_j$ explains both forward chaining and recovery re-entry.

Tie et al., 2025; Chen et al., 2026; Zhou et al., 2026). Their weakness is that a plausible label does not determine the physical transition. The phrase *align connector* might refer to reducing pose error, establishing compliant contact, changing cable deformation, or orienting a keyed part. Object-centric discovery requires deciding which of these transitions is actually present.

Unsupervised and demonstration-based methods search for recurring structure in trajectories. BUDS, LOTUS, LOKI, AtomSkill, KISA, SymSkill, and related methods discover latent segments, symbols, or skill boundaries from behavior (Zhu et al., 2022; Wan et al., 2024; Zhu et al., 2026, 2025; Kou et al., 2024; Shao et al., 2025). This line builds on foundational skill-discovery and decomposition work: skill chaining and skill trees construct reusable skills from demonstration and experience (Konidaris and Barto, 2009; Konidaris et al., 2012), CompILE segments trajectories into composable latent skills (Kipf et al., 2019), and play- and relay-style methods learn reusable behaviors from unstructured interaction (Lynch et al., 2020; Gupta et al., 2019). They become object-centric only when clusters align with recurring object effects rather than merely with velocity, gripper aperture, visual saliency, or demonstrator style. Contact- and object-state-based methods are more physically aligned because they use force, tactile, or object-relation changes to define boundaries (Mao et al., 2024b; Johannsmeier et al., 2025; Chen et al., 2025b). Cross-embodiment discovery adds another test: if a transition is preserved across human hands, parallel-jaw grippers, and dexterous hands, it is more likely object-defined than embodiment-specific (Xu et al., 2023; Kim et al., 2025; Wan et al., 2025). SBD-style boundary discovery and GSL-style object-centric hierarchies illustrate the difference between useful segmentation and object-centric discovery: a boundary caused by prediction error is useful only when it corresponds to a change in $X$, $C$, $I$, or $\beta$ (Deng et al., 2025; Zhao et al., 2025a).

This criterion can be written as an evaluation objective rather than as a single prescribed algorithm:

$$\min_{\{z_t\},\{\tau_j\}} \sum_j \sum_{t:z_t=j} \|\Delta_t - \Delta_j\|^2_{\mathcal{X},\mathcal{C}} + \lambda\, \mathcal{L}_{\text{interface}}(I_j, \beta_j). \tag{2}$$

The first term asks whether segments assigned to the same type produce consistent object/contact effects; the second asks whether the proposed unit has testable boundaries. Language supplies candidate names, unsupervised learning supplies recurring latent structure, contact events supply physically meaningful boundaries, and cross-embodiment recurrence tests whether the transition is defined by the object rather than by one actuator.

### 4.2 Stage II: Learning and Anchoring

Once a candidate skill type $S$ exists, it must be parameterized as a policy $\pi$ grounded in object state. The key point is that perception is not merely upstream of the skill. It is part of the skill definition. $X_{\text{sem}}$ determines which object and action family are relevant; $X_{\text{met}}$ determines pose, geometry, clearance, and tolerance; $X_{\text{tac}}$ determines whether contact has occurred, whether slip or jam is present, and whether force is acceptable.

Semantic grounding is now highly developed. RT-2 casts robot actions as token sequences inside a VLM (Brohan et al., 2023a); FAST, VQ-VLA, ActionCodec, and UniAct improve action tokenization and cross-embodiment action representation (Pertsch et al., 2025; Wang et al., 2025; Dong et al., 2026b; Zheng et al., 2025). These methods narrow the skill hypothesis space, but they do not by themselves specify physical validity. A token for *insert* does not encode peg diameter, hole pose, chamfer geometry, allowable force, or jam recovery.

Metric and spatial grounding make this physical content more explicit. SpatialVLA, PointVLA, DepthVLA, SpatialBot, SG-VLA, SP-VLA, AlphaSpace, and MetricAnything each expose part of $X_{\text{met}}$: egocentric 3D structure, point-cloud

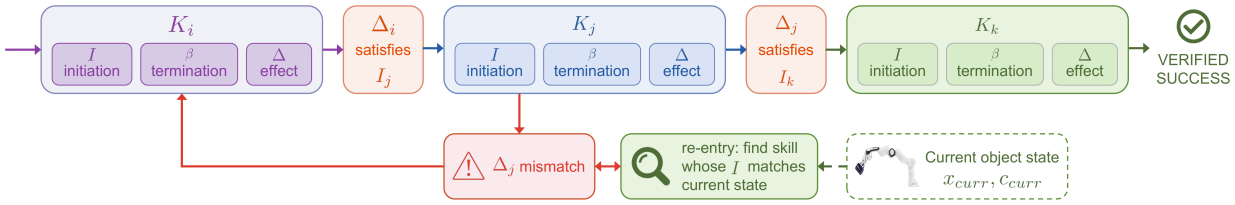


Fig. 6: **Composition as effect-to-initiation matching.** A task goal is decomposed into candidate skills, but an edge is valid only when the predicted effect $\Delta_i$ of one skill satisfies the initiation interface $I_j$ of the next. On effect mismatch, recovery is re-entry into the same interface graph from the observed abnormal state rather than a separate engineering patch.

geometry, depth, relative pose, spatial priors, or metric scale (Qu et al., 2025; Li et al., 2026a; Yuan et al., 2025; Cai et al., 2024a; Tu et al., 2026; Ye et al., 2026; Dao et al., 2025; Ma et al., 2026). For articulated objects, CAPER++ shows a complementary perception-side route: joint-centric pose estimation and tracking recover root-part pose, constrained-part joint states, and temporally coherent SE(3) increments, turning object geometry into a structured state that a skill interface can inspect (Zhang et al., 2026b). Tactile and force-aware methods expose $X_{\text{tac}}$ and $C$. Tactile-VLA, TacVLA, ForceVLA, OmniVTLA, VTLA, Sparsh, and UniTouch show how tactile images, force–torque signals, and cross-sensor tactile embeddings can be integrated into policy learning (Huang et al., 2025a; Zhang et al., 2026a; Yu et al., 2025b; Cheng et al., 2025; Zhang et al., 2025b; Higuera et al., 2024; Yang et al., 2024).

Policy parameterization determines how these signals are used. Action chunks provide temporal stability but require effect checks at chunk boundaries (Zhao et al., 2023). Diffusion policies represent multimodal trajectories but should organize modes by equivalent object effects rather than by action-space similarity alone (Chi et al., 2023; Huang et al., 2026a; Liang et al., 2024). Discrete skill codes and atomic VLAs compress the action or skill space, but their codes must preserve the state variables that define the transition (Zhang et al., 2026c; Sun et al., 2026b; Hao et al., 2026). Expert routing naturally resembles skill libraries when each expert specializes in a transition type over $X$ and $C$ (Reuss et al., 2025; Tang et al., 2026). The common diagnostic question is whether the learned representation preserves the information needed for $I$, $\beta$, and $\Delta$. Action chunks and diffusion policies can be excellent controllers while still hiding the effect they produce. Discrete codes and atomic VLAs can become genuine skill vocabularies only when a code corresponds to a reusable transition rather than to a statistically convenient action fragment. Expert routing is most compatible with the object-centric view when experts specialize by transition type, such as support establishment, containment, insertion, sliding, or deformable alignment, instead of by scene identity alone.

### 4.3 Stage III: Composition and Orchestration

Composition is the central bottleneck. It is not enough for each skill to succeed alone. The post-state of one skill must land inside the initiation interface of the next. For skills $K_i$ and $K_j$, define the interface triple

$$\mathcal{J}_i = (I_i, \beta_i, \Delta_i). \tag{3}$$

The compatibility condition is

$$K_i \rightsquigarrow K_j \iff \Pr_{(x',c')\sim\Delta_i}\left[(x', c', e) \in I_j\right] \geq \tau_j. \tag{4}$$

This equation expresses the core idea of the survey: a chain edge is not just temporal adjacency between names; it is a verified object-state relation. Fig. 6 expands this compatibility check into a full decomposition, execution, mismatch, and recovery loop.

Language-level planners, such as SayCan, Code as Policies, VoxPoser, SayPlan, Inner Monologue, and ELLMER, are effective at proposing candidate sequences (Ahn et al., 2022; Liang et al., 2023; Huang et al., 2023; Rana et al., 2023; Huang et al., 2022; Mon-Williams et al., 2025). Planner-integrated and hierarchical systems, including RoboMatrix, Hi Robot, PSL, SPIRE, TASP, and MOSAIC, connect high-level proposals to lower-level skills (Mao et al., 2024a; Shi et al., 2025; Dalal et al., 2024; Zhou et al., 2024; Hedegaard et al., 2025; Mishani et al., 2025). Affordance-grounded planners make initiation conditions more explicit through object affordances or scene-graph relations (Birr et al., 2024; Jiang et al., 2024; Yu et al., 2025a). Across these systems, the object-centric question is the same: does the selected next skill's $I$ actually hold in the current object state?

Composition structures differ in how naturally they expose interfaces. Sequential chains are easy to generate but brittle. DAGs expose partial order but often lack feedback. Behavior trees encode success, failure, fallback, and retry, making

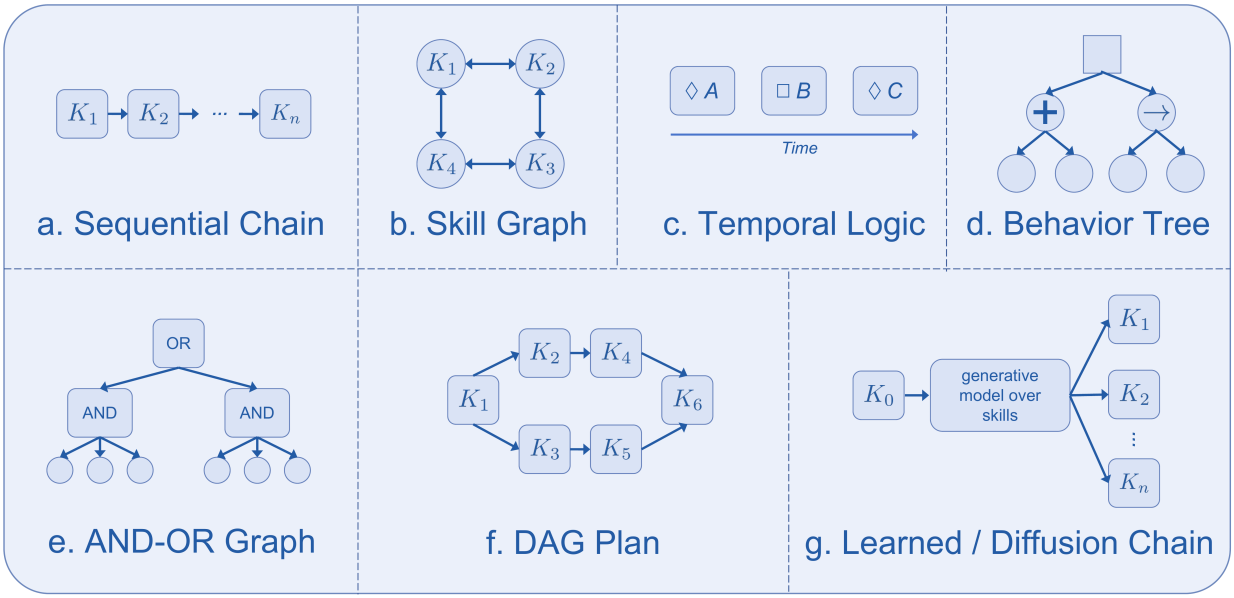


Fig. 7: **Seven composition families for atomic skills.** Sequential chains, DAG plans, behavior trees, skill graphs, temporal logic, AND–OR graphs, and learned chains differ less in expressive power than in how explicitly they expose the interface triple $(I, \beta, \Delta)$.

them well aligned with $\beta$. Skill graphs can store reusable $\Delta \to I$ edges. Temporal logic provides formal composition semantics if propositions are grounded in object state. AND–OR graphs capture alternative decompositions, as in TMP-IDAN (Karami et al., 2025). Diffusion and learned chains generate flexible long-horizon rollouts but usually keep boundaries latent. Hierarchical VLAs can show impressive emergent composition but often do not expose reusable interfaces. Fig. 7 and Table 3 organize these alternatives by the kind of graph they use and by how explicitly they expose object state, recovery, and $\Delta \to I$ edges.

Sequential chains remain the most common form because they match language plans and script-like task descriptions. SayCan-style iterative selection is representative: an LLM proposes the next skill and a learned value function scores whether the current state affords it (Ahn et al., 2022). In object-centric terms, the edge should be a $\Delta \to I$ compatibility check rather than only temporal precedence. DAG plans, including DAG-Plan, LiP-LLM, and RoboPARA, expose partial order and parallel dependencies, but usually provide weak feedback when execution changes the state unexpectedly (Gao et al., 2024; Obata et al., 2024; Duan et al., 2026).

Behavior trees are especially aligned with $\beta$ because control flow is defined by success, failure, fallback, and retry. BTGenBot, HBTP, and Code-BT show how natural-language task descriptions can be converted into executable tree structures, but the predicates in the tree must still be grounded in object state for the tree to serve as a verifiable skill graph (Izzo et al., 2024; Cai et al., 2024b; Zhang et al., 2025e). Skill graphs and knowledge-graph planners, such as HP-KG and GSC, are natural carriers for reusable libraries because an edge can store an empirical $\Delta \to I$ relation (Zhou et al., 2025b; Tian et al., 2024). Temporal-logic methods such as Skill Machines, SELP, and LTLDoG offer strong formal semantics, but their guarantees depend on whether each proposition can be perceived and verified in the metric/tactile world (Tasse et al., 2022; Wu et al., 2024; Feng et al., 2024a).

Learned composition methods attack the same interface problem from the policy side. ChainedDiffuser and DOPPLER generate long-horizon plans through diffusion-style rollouts, while GPC and FDP compose policy distributions through score combination (Xian et al., 2023; Feng et al., 2024b; Cao et al., 2026; Liu et al., 2025a). These methods are flexible, but they become object-centric composition only when generated segments expose checkable initiation and effect variables. Hierarchical VLAs, including Long-VLA, LiLo-VLA, SkillVLA, and $\pi_{0.7}$, provide the strongest current evidence that composition can emerge implicitly under scale (Fan et al., 2025; Yang et al., 2026a; Zhai et al., 2026; Physical Intelligence et al., 2026). The limitation is not behavior but inspectability: the model may compose useful subtasks without exposing the reusable interface that made the next subtask physically callable.

Several algorithms target the boundary between composed skills directly. T-STAR regularizes terminal states so that the next skill can start; SCaR adds intra-skill and inter-skill regularization; SeqVLA uses completion-aware transition signals; and MuST estimates continuous progress for long-horizon switching (Lee et al., 2021; Chen et al., 2024; Yang

Table 3: Comparison of task-oriented skill composition forms. ●: usually explicit; ○: partial or implicit; –: usually absent.

| Form | Nodes | Edges | Obj. state? | Recovery? | $I/\Delta$? | Key papers | Limitation |
|---|---|---|---|---|---|---|---|
| Sequential chains | Ordered skills | Next-step order | ○ | – | ○ | SayCan; Code as Policies; Inner Monologue (Ahn et al., 2022; Liang et al., 2023; Huang et al., 2022) | Simple but brittle; weak branching and recovery |
| DAGs | Partial-order skills | Dependency | ○ | ○ | ○ | DAG-Plan; LiP-LLM; RoboPARA (Gao et al., 2024; Obata et al., 2024; Duan et al., 2026) | Captures ordering; weak feedback |
| Behavior trees | Condition-action nodes | Tick/control flow | ○–● | ● | ○–● | BTGenBot; HBTP; Code-BT (Izzo et al., 2024; Cai et al., 2024b; Zhang et al., 2025e) | Good recovery; engineering-heavy |
| Skill graphs | Reusable skill nodes | Reachability | ○ | ● | ○ | HP-KG; GSC (Zhou et al., 2025b; Tian et al., 2024) | Natural for libraries; edge semantics often vague |
| Temporal logic | Propositions | Formal constraints | ●* | ● | ●* | Skill Machines; SELP; LTLDoG (Tasse et al., 2022; Wu et al., 2024; Feng et al., 2024a) | Verifiable; hard grounding (*if propositions are object-centric) |
| AND–OR graphs | Decompositions | Refinement | ○ | ● | ○ | TMP-IDAN (Karami et al., 2025) | Captures alternatives; exponential branching |
| Diffusion / learned chains | Latent segments | Learned rollout | ○ | ○ | ○/– | ChainedDiffuser; DOPPLER (Xian et al., 2023; Feng et al., 2024b) | Flexible generation; weak explicit interfaces |
| Hierarchical VLA | Language subgoals | Planner routing | ○ | ○ | ○/– | Long-VLA; LiLo-VLA; SkillVLA (Fan et al., 2025; Yang et al., 2026a; Zhai et al., 2026) | Powerful behavior; composition often implicit |

et al., 2025; Gao et al., 2025). These works can be read as attempts to reduce the $\Delta \to I$ gap: they make the terminal state of one skill more predictable, or make the next initiation easier to satisfy. Reasoning and subgoal-generation methods such as ECoT, CoT-VLA, CoA-VLA, FlowVLA, CoTDiffusion, SuSIE, and WorldVLA are similarly useful when they infer candidate interfaces rather than only producing explanations (Zawalski et al., 2024; Zhao et al., 2025b; Li et al., 2025a; Zhong et al., 2025; Ni et al., 2024; Black et al., 2024b; Cen et al., 2025).

### 4.4 Stage IV: Execution and Recovery

Execution closes the loop. A composed plan matters only if the robot can preserve, verify, and repair the intended object-state transitions in the real world. Closed-loop execution should track the predicted trajectory of $(X, C)$, not only end-effector motion. For contact-rich manipulation, this includes force signatures, slip, deformation, support, containment, insertion depth, and latch engagement. Formally, the policy is a closed-loop controller conditioned on the intended transition:

$$a_t \sim \pi_i(a_t \mid X_t, C_t, h_t, \theta_i), \quad g_i \equiv \Delta_i(x_t, c_t). \tag{5}$$

The execution metric is therefore not action similarity alone. A robot may follow a demonstrated trajectory while failing to seat a connector, maintain a stable grasp, or establish the correct support relation; conversely, a different trajectory may succeed if it realizes the same object-state transition.

Verification is effect-mismatch detection:

$$d\big(\widehat{x}_{t+1}, x_{t+1}^{\text{obs}}\big) + d_C\big(\widehat{c}_{t+1}, c_{t+1}^{\text{obs}}\big) \le \epsilon. \tag{6}$$

If the observed post-state differs from the predicted effect, the system should not blindly advance to the next skill. It should update the current object state, classify the boundary through $\beta$, and re-enter the interface graph from the abnormal state. Recovery is therefore not an add-on; it is composition from a failed or unexpected state. This can be written as ordinary skill selection from an abnormal state:

$$K_{\text{next}} = \arg\max_{K_j \in \mathcal{L}} \Pr\big[(\hat{x}_t, \hat{c}_t, e) \in I_j\big]\, U(\Delta_j, g), \tag{7}$$

where $\mathcal{L}$ is the skill library and $U$ estimates expected progress toward the task goal.

Recent systems show partial versions of this loop. Failure-detection and recovery methods such as FailSafe, RoboFAC, I-FailSense, ARMOR, RAMA, RationalVLA, RACER, ConditionNET, and recovery chaining identify, classify, or repair execution errors (Lin et al., 2025; Ye et al., 2025; Grislain et al., 2025; Qi et al., 2026; Song et al., 2025; Dai et al., 2024; Sliwowski and Lee, 2025; Vats et al., 2025). Human-in-the-loop systems such as HIL-SERL and RoboCopilot provide practical recovery channels in real robots (Luo et al., 2025; Wu et al., 2025). Hi-WM moves this loop inside a learned world model, caching intermediate states and supporting rollback and branching so that a single failure state can spawn multiple human-guided corrective trajectories (Li et al., 2026d). Online adaptation and RL refinement methods such as DPPO, SimpleVLA-RL, STARE-VLA, SC-VLA, TT-VLA, and VLA-RL improve execution after deployment, but their reward or correction signal is most useful when it tracks the intended object-state effect rather than generic task success alone (Ren et al., 2025; Li et al., 2026b; Xu et al., 2025; Liu et al., 2026b,a; Lu et al., 2025a). Perturbation benchmarks and robustness studies, including LIBERO-Plus, RobustVLA, and Eva-VLA, show why this matters: viewpoint shifts, pose perturbations, slippery surfaces, and unexpected contacts perturb different variables in $K$ and should induce different re-entry decisions (Fei et al., 2025; Zhang et al., 2025c; Liu et al., 2025b). Generalist policies such as $\pi_{0.7}$ use subtask language, metadata, and generated subgoal images as steering signals, functioning as an implicit prompt-level substitute for explicit $I$, $\beta$, and $\Delta$ (Physical Intelligence et al., 2026). The remaining gap is to expose these variables as reusable, verifiable interfaces rather than leaving them internal to a monolithic policy.

## 5 Understanding Current Methods Through the World-Action Model

The object-centric world-action model changes how current methods should be read. Instead of sorting work only by architecture, we ask where each method sits in the world-action loop: what world state it sees, what transition it predicts or executes, what interface it exposes, and what failure it can detect. This section applies that lens to real-world systems, benchmarks, data, and recent world-model methods.

### 5.1 Generalist Robot Systems

Large generalist systems internalize increasingly large parts of the skill chain. $\pi_0$ and $\pi_{0.5}$ show that flow-based VLA policies can scale across tasks and embodiments (Black et al., 2024a; Physical Intelligence et al., 2025). $\pi_{0.7}$ extends this line with a large vision-language backbone, a flow-matching action expert, video history, metadata-conditioned prompting, and generated subgoal images from a lightweight world model (Physical Intelligence et al., 2026). The reported behaviors are important for this survey because the model can compose skills seen in separate contexts and transfer to an unseen UR5e bimanual platform. In our terms, this is evidence that $\Delta \to I$ compatibility can emerge implicitly under scale. The limitation is that the interface remains hidden: the system does not expose the postcondition that made the next subtask callable. The example is concrete. $\pi_{0.7}$ is steered by subtask language, episode metadata, and generated subgoal images, and reports behaviors such as loading a sweet potato into an air fryer by composing experience from related but separately observed skills. For this survey, such behavior is important not because it proves that explicit skill libraries are unnecessary, but because it reveals what a successful implicit interface looks like: each subtask and generated subgoal image acts as a low-resolution proxy for an expected object-state transition. The missing step is to expose that proxy as a reusable $\Delta$ that another planner, verifier, or recovery module can inspect.

Task-specific and industrial systems expose different parts of the chain. MT3 decomposes manipulation into alignment and interaction phases and obtains broad task coverage with limited human demonstration time (Dreczkowski et al., 2025). HIL-SERL uses human-in-the-loop reinforcement learning for reliable real-world improvement (Luo et al., 2025). IndustReal and AutoMate make contact and assembly structure more explicit (Tang et al., 2023, 2024). These systems are less open-ended than generalist VLAs, but they often provide stronger object-state, contact, and verification structure. This contrast is central: broad policies tend to hide interfaces, while narrower assembly systems often expose them. HAMSTER and related mobile manipulation systems sit between these extremes: they broaden task coverage and embodiment diversity, but still rely heavily on learned internal representations rather than declared skill interfaces

Table 4: Representative real-world systems read through the object-centric skill descriptor. ●: explicit; ○: partial or implicit; –: usually absent.

| System | $X_{\text{sem}}$ | $X_{\text{met}}$ | $X_{\text{tac}}$ | $C$ | $I/\Delta$ | $\pi$ | $\beta$ | Recov. | Multi-$E$ |
|---|---|---|---|---|---|---|---|---|---|
| $\pi_0/\pi_{0.5}$ | ● | ○ | – | ○ | ○ | ● | ○ | ○ | ● |
| $\pi_{0.7}$ | ● | ○ | – | ○ | ○ | ● | ○ | ○ | ● |
| MT3 | ● | ● | – | ○ | ○ | ● | ○ | – | – |
| HIL-SERL | ○ | ● | ○ | ● | ○ | ● | ● | ● | – |
| IndustReal | ○ | ● | ● | ● | ● | ● | ● | ○ | – |
| AutoMate | ● | ● | – | ● | ● | ● | ○ | – | – |
| HAMSTER | ● | ○ | – | ○ | ○ | ● | ○ | ○ | ○ |

(Li et al., 2025c). The comparison in Table 4 should therefore be read as a diagnostic map, not a leaderboard. Different systems are strong at different links of the object-centric chain.

### 5.2 Benchmarks and Data

Benchmarks reveal what the field is actually optimizing. LIBERO and CALVIN stress long-horizon language-conditioned manipulation and compositionality (Liu et al., 2023; Mees et al., 2022). RLBench and ManiSkill3 provide broad simulated task families (James et al., 2020; Tao et al., 2025). RoboCasa targets household manipulation (Nasiriany et al., 2024). ClevrSkills isolates compositional reasoning over well-specified skills (Haresh et al., 2024). FurnitureBench stresses real-world assembly and long horizons (Heo et al., 2025). RAMA and related failure benchmarks stress rejection and recovery (Song et al., 2025). LIBERO-Plus is especially useful for this survey because it shows how quickly VLA performance can degrade under realistic perturbation dimensions, including object layout, viewpoint, and sensor noise (Fei et al., 2025). Such drops are not only robustness failures; they are evidence that semantic skill labels are insufficient when the interface variables in $K$ move out of distribution.

No current benchmark fully spans the object-centric chain. Semantic compositional benchmarks under-stress force, slip, and tight tolerance. Assembly benchmarks stress contact and precision but have limited cross-object and cross-embodiment diversity. Failure benchmarks stress verification and recovery but rarely combine this with rich tactile contact and long-horizon composition. Recent efforts such as VLA-Arena, ManipArena, and ManiSkill-ViTac reduce the gap by adding safety, reasoning-oriented evaluation, and vision–tactile coverage (Zhang et al., 2025a; Sun et al., 2026c; Li et al., 2024). The key missing benchmark unit is not a single task but a chain with observable interfaces: the dataset should record whether each intermediate object-state transition occurred, which initiation condition failed, and what recovery route was available. Without this information, a benchmark can report success rate but cannot tell whether the system failed at discovery, grounding, composition, execution, verification, or recovery.

Data have the same issue. Large robot and egocentric datasets provide scale, but object-centric skills require more than state-action pairs. The data should preserve object identity, contact evidence, skill boundaries, and observed state transitions. Egocentric human data are valuable because they capture object-centered perception–action coupling at close range (Kareer et al., 2024; Hoque et al., 2026; Generalist AI Team, 2026). Large-scale cross-embodiment corpora such as Open X-Embodiment aggregate manipulation data across many robots and laboratories (Open X-Embodiment Collaboration, 2024), and cross-embodiment data such as XSkill and UniSkill test whether a transition persists when the body changes (Xu et al., 2023; Kim et al., 2025). The key question is whether the dataset supports reconstructing $R$, $X$, $C$, $I$, $\beta$, and $\Delta$, even when they are not explicitly annotated. GEN-0 and GEN-1 style egocentric pretraining are relevant here because they preserve the actor's closed perception-action loop around persistent objects rather than separating semantic labels from action traces (Generalist AI Team, 2025, 2026). Other recent data and representation efforts point to complementary missing variables: EgoDex and EgoMimic emphasize dexterous human-object interaction; XSkill and UniSkill stress cross-embodiment recurrence; tactile and visuotactile datasets stress contact evidence; and world-model datasets stress observed transitions (Kareer et al., 2024; Hoque et al., 2026; Xu et al., 2023; Kim et al., 2025; Higuera et al., 2026). The object-centric survey question is whether these sources can be aligned into a shared state space rather than remaining separate semantic, metric, tactile, and transition corpora.

### 5.3 JEPA-Style World Models as Transition Interfaces

World models are central to PraxisWorld: they supply the predicted transition $\Delta$ on which every interface check depends, and the effect model is precisely the weakest part of most current skill systems. In this survey, JEPA is not treated as a separate category from world models. It is a major world-modeling route in which prediction is performed in a learned

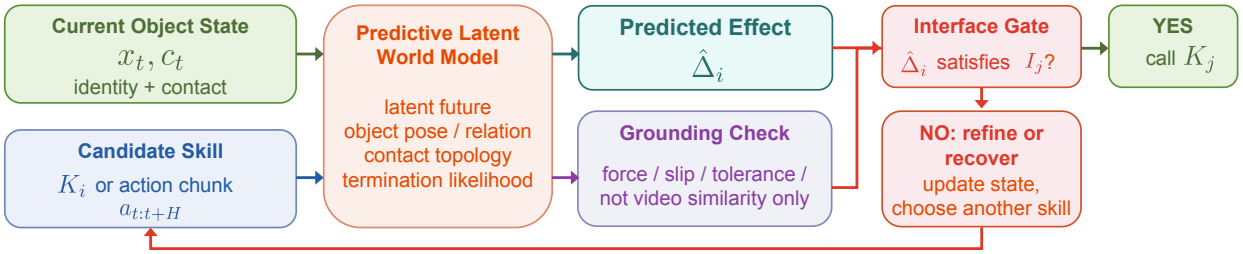


Fig. 8: **World models as transition-interface estimators.** JEPA-style latent predictors and action-conditioned world models are useful for this survey because they target the weakest variable in many skill frameworks: the predicted effect $\Delta$. The open challenge is to align latent or visual futures with explicit object-state, contact, termination, and initiation interfaces that a planner can verify.

representation space rather than only in pixel space. For manipulation, the important question is therefore not whether a model predicts pixels, latent embeddings, or action-conditioned rollouts, but whether the prediction can be converted into the object-state transition needed by a skill interface.

A useful manipulation world model should estimate how object state and contact topology will change under candidate actions or skills:

$$p_\theta(x_{t+1:t+H}, c_{t+1:t+H}, \Delta_{t:t+H} \mid x_t, c_t, X_t, K_i, a_{t:t+H}). \tag{8}$$

This model becomes a transition interface only after its prediction can be tested against $I_j$ and $\beta_j$. In the descriptor notation, the current state $(X_t, C_t)$ and candidate skill $K_i$ produce a predicted effect $\Delta_i$; the next edge is valid only if that effect satisfies the initiation interface of the next skill. Fig. 8 places recent JEPA-style and action-conditioned world models in this role. Their value for composable manipulation is not future generation by itself, but the ability to supply checkable evidence about object state, contact, tolerance, and termination.

**LeWorldModel.** LeWorldModel (LeWM) is a recent representative of the JEPA-style world-modeling direction (Maes et al., 2026). It trains a stable end-to-end joint-embedding predictive world model from raw pixels using a next-embedding prediction loss plus a Gaussian latent regularizer. The paper reports a small $\sim$15M-parameter model that can be trained on a single GPU in a few hours, plans up to 48$\times$ faster than foundation-model-based world models, and remains competitive across several 2D and 3D control tasks. It also reports that the learned latent space encodes physical quantities and responds strongly to physically implausible events. For our framework, the importance of LeWM is not that it solves robot manipulation by itself. It shows that stable, efficient latent transition prediction can be learned without relying on a frozen foundation encoder or pixel reconstruction, making it a plausible substrate for learning $\Delta$.

**V-JEPA 2-AC.** V-JEPA 2 moves JEPA-style world modeling closer to robot manipulation (Assran et al., 2025). The model is pretrained on internet-scale video and then post-trained as an action-conditioned latent world model using less than 62 hours of DROID robot video (Khazatsky et al., 2024). The authors deploy V-JEPA 2-AC zero-shot on Franka arms in two labs for reaching, grasping, and pick-and-place with image goals, without environment-specific data collection or task-specific reward. In object-centric terms, V-JEPA 2-AC is closer to a manipulation-relevant $\Delta$ model: it scores candidate action rollouts by predicted progress toward a visual goal. However, its output is still primarily latent or image-goal based, not an explicit predicate over object state, contact, tolerance, and termination.

**WorldVLA, VLA-JEPA, and AtomVLA.** WorldVLA integrates action generation and world-model prediction in one autoregressive VLA, using future-image prediction to improve action generation (Cen et al., 2025). VLA-JEPA adopts a JEPA-style latent world model for VLA pretraining and reports gains on LIBERO, LIBERO-Plus, SimplerEnv, and real-world manipulation tasks (Sun et al., 2026a; Liu et al., 2023; Fei et al., 2025; Li et al., 2025b). AtomVLA decomposes demonstrations into atomic subtasks and uses a predictive latent world model to score candidate action chunks against subtask goals, reporting strong LIBERO and real-world results (Sun et al., 2026b). Together, these systems form a natural bridge from LeWM-style latent transition learning to VLA manipulation. They suggest a concrete way to combine world models with the object-centric descriptor: a VLA proposes a subgoal or candidate action chunk, the world model predicts the future state or latent effect, and an interface module checks whether the predicted state is compatible with the next intended transition. In this role, the world model supplies $\Delta$, while the skill descriptor supplies the variables that make $\Delta$ inspectable.

**The emerging World Action Model (WAM) literature.** A fast-growing 2026 line of work makes this world-action coupling explicit under the name *World Action Models*: models that target a joint distribution over future states and actions rather than actions alone. Two recent surveys map this territory, one on world models for manipulation

broadly (Zhang et al., 2025d) and one proposing the WAM taxonomy directly (Wang et al., 2026), and a wave of systems instantiate it. STARRY jointly denoises future spatio-temporal latents and actions (Tian et al., 2026); an action-regularized world model adds an inverse-dynamics objective so that latent transitions stay action-relevant (Han and Yilmaz, 2026); the World-Value-Action model couples prediction with a trajectory value function for implicit long-horizon planning (Li et al., 2026c); and event-centric models such as EA-WM and the open-source WALL-WM organize learning around object-state-changing events, such as reaching, contact, grasping, lifting, and placing, rather than fixed action chunks (Yang et al., 2026b; X Square Robot, 2026). A robustness study reports that such world-action models degrade less than reactive VLAs under perturbation (Zhang et al., 2026d), which is consistent with the central claim of this survey: exposing a predictive transition interface, rather than memorizing observation-to-action maps, is what makes manipulation robust and recoverable. The work closest to our formulation is OA-WAM, which decomposes the scene into per-object slots with a persistent address separated from time-varying content, so that an instruction can be bound to *which* object independently of *what* it currently looks like (Liu et al., 2026c). Object-centric atomic skills can be read as the interface-level counterpart of this representation: where OA-WAM resolves object identity inside the world model, our descriptor exposes the initiation, termination, and effect interfaces $(I, \beta, \Delta)$ that make the predicted transition checkable. The distinction worth drawing is therefore not whether to build a world-action model, since the field is converging on that, but whether its transitions are exposed as *verifiable object-state interfaces* ($\Delta_i \models I_j$) that support composition and recovery, or left implicit inside a monolithic joint model.

**Physics-grounded and object-centric world models.** A complementary route makes the predicted effect $\Delta$ explicit by grounding prediction in object state and physics rather than only in latent video features. Object-centric world models predict per-object futures: FOCUS and a slot-based language-guided world model organize prediction around object slots rather than whole frames (Ferraro et al., 2025; Jeong et al., 2025), while latent-feature predictors such as DINO-WM plan directly on pre-trained visual features (Zhou et al., 2025a). Particle- and graph-based neural dynamics learn material-adaptive object interactions, can be driven from dynamic 3D Gaussian tracking, and extend to multi-object, multi-material scenes (Zhang et al., 2024a,b; Huang et al., 2025b, 2026b). A further line ties generative 4D object dynamics and contact physics to the same representation: neural object kinematics turn a static shape into a simulatable 4D asset (Geng et al., 2026), physically-aware human–object interaction generation enforces contact-driven momentum transfer through material-point simulation (Benishu et al., 2026), and scalable Gaussian world models predict future object states under candidate actions (Lu et al., 2025b). For object-centric skills, the value of these models is that their predictions are already expressed over object state and contact—exactly the form an initiation interface $I_j$ can test.

The survey-level conclusion is cautious but useful. World models are not automatically object-centric. Pixel futures can be visually plausible while missing contact, force, or hidden state. Latent futures can improve planning while remaining uninterpretable. But JEPA-style world models are one of the most logical and influential directions for this survey because they attack the exact missing variable $\Delta$. The next step is to turn latent or visual predictions into explicit object-state effects that can be checked against initiation interfaces.

### 5.4 A Diagnostic Simulation Probe: Object-State Interface Gates

The framework above suggests a simple experimental question: when the object state at a skill boundary changes unexpectedly, is it enough to continue executing the planned action suffix, or should the system first check whether the next interface is still callable? To make this question concrete without introducing a new learned policy, we ran a small diagnostic probe in Meta-World V3, a MuJoCo-based manipulation benchmark with scripted expert policies (Yu et al., 2020; Todorov et al., 2012). This probe is not intended as a full benchmark or a new method. Its purpose is narrower: to test, in simulation, whether an explicit object-state gate can expose a transition mismatch that is invisible to an open-loop action suffix.

We evaluate three Meta-World tasks: basketball, bin-picking, and shelf-place. For each task, we first roll out the public scripted expert policy and record a nominal action trajectory and object-state stream. At a fixed transition point, we perturb the manipulated object in the simulator by a random planar displacement of 0–12 cm and use a 4 cm tolerance as the transition gate. We then compare six continuations over 30 seeds per task. *Nominal expert replay* replays the recorded trajectory without perturbation. *Perturbed open-loop suffix* applies the perturbation but continues the originally recorded action suffix, representing a chain that assumes the previous effect $\Delta_i$ still satisfies the next initiation interface $I_j$. *Semantic gate* checks only the semantic/task-level interface $X_{\text{sem}}$; because the perturbation preserves object identity and task label, it behaves like the open-loop suffix. *Object-state gate, verify only* checks the metric object-state interface $X_{\text{met}}$ and rejects the transition if the mismatch exceeds tolerance, but does not attempt recovery. *Object-state gate + recovery* uses the same gate and, when the transition is rejected, re-queries the scripted expert policy in closed loop. *Oracle feedback recovery* always re-queries the expert policy after the transition and serves only as an upper bound on

Table 5: Diagnostic Meta-World probe over 90 simulated rollouts. The ablation separates label-level continuation, semantic gating, object-state verification, state-gated recovery, and an oracle feedback upper bound.

| **Continuation** | **Interface used** | **Success (%)** | **State mismatch (%)** | **Alarm (%)** | **Reject (%)** | **Mismatch (cm)** |
|---|---|---|---|---|---|---|
| Nominal expert replay | – | 80.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| Perturbed open-loop suffix | – | 21.1 | 72.2 | 0.0 | 0.0 | 6.1 |
| Semantic gate | $X_{\text{sem}}$ | 21.1 | 72.2 | 0.0 | 0.0 | 6.1 |
| Object-state gate, verify only | $X_{\text{met}}$ | 18.9 | 72.2 | 72.2 | 72.2 | 6.1 |
| Object-state gate + recovery | $X_{\text{met}}, \Delta \to I$ | 54.4 | 72.2 | 72.2 | 0.0 | 6.1 |
| Oracle feedback recovery | always closed loop | 55.6 | 72.2 | 0.0 | 0.0 | 6.1 |

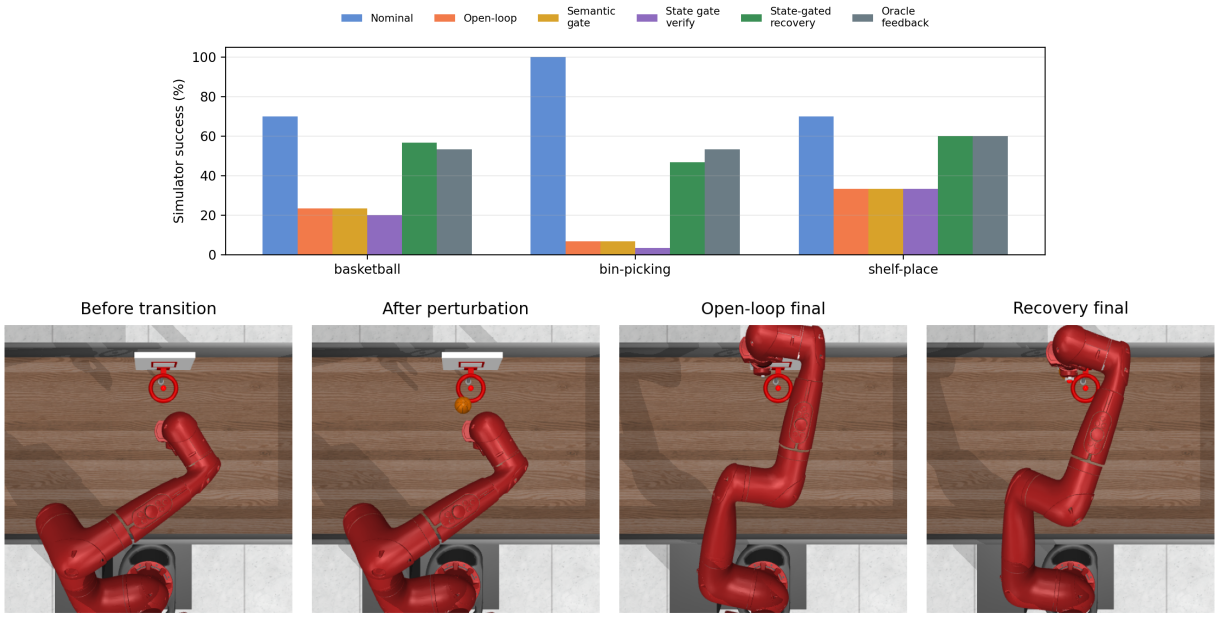


Fig. 9: **A small Meta-World object-state interface probe.** Top: per-task success rates for nominal replay, open-loop continuation, semantic gating, verify-only object-state gating, state-gated recovery, and oracle feedback. Bottom: an upright top-view example in which the object is displaced at the skill boundary; continuing the old action suffix fails, while the same expert policy can recover when the mismatch is detected and control is re-queried in closed loop.

what feedback recovery can achieve in this setup. Thus the comparison separates semantic gating, metric verification, recovery, and an oracle feedback reference.

Table 5 reports the aggregate result over the 90 rollouts (3 tasks $\times$ 30 seeds) and Fig. 9 shows the per-task breakdown and an example rollout; we report Wilson 95% confidence intervals on the pooled success rates. The nominal expert succeeds in 80.0% of rollouts (95% CI $[70.6, 87.0]$). After object-state perturbation, blindly continuing the action suffix drops success to 21.1% ($[14.0, 30.6]$). Because the perturbation is drawn and applied identically across continuations and *before* they diverge, the raw state-mismatch rate (72.2%) and its mean magnitude (6.1 cm) describe the disturbance itself; they are therefore shared across all perturbed rows, and only success, alarm, and reject vary with the continuation strategy. The semantic gate is identical to the open-loop suffix by construction: a label-level interface that inspects only object identity and task label cannot, in principle, observe a metric displacement that preserves both. The verify-only object-state gate detects and rejects 72.2% of transitions (18.9% success, $[12.1, 28.2]$), showing that verification alone

can expose unsafe continuation but, by aborting, cannot complete the task. Adding closed-loop recovery raises success to $54.4\%$ ($[44.2, 64.3]$); the improvement over the open-loop suffix is unambiguous ($+33.3$ points, with 30 of 90 rollouts recovered and none lost), and the residual gap to the $55.6\%$ oracle feedback reference ($[45.3, 65.4]$) is a single rollout (49 versus 50 of 90), i.e. within the noise of this sample. This does not prove that our descriptor is a complete algorithm, and the probe deliberately exercises only the metric interface $X_{\mathrm{met}}$ on non-contact-rich tasks. It does show that a minimal $\Delta_i \to I_j$ check can make the difference between treating a disturbed transition as an ordinary action suffix and treating it as a recoverable interface failure.

The probe also clarifies the role of future world models in this framework. Here, the expected transition state is recorded from a scripted expert trajectory, which makes the interface check deliberately simple. In a learned system, a JEPA-style or action-conditioned world model could supply the expected future object state instead. The same gate would then compare a predicted or observed $\Delta_i$ against the next initiation condition $I_j$ and decide whether to continue, replan, or recover. This is the practical meaning of converting visual or latent prediction into an object-state transition interface.

To support scrutiny and reuse, the probe code, scripted-policy rollouts, and per-seed results are released as supplementary material. The experiment is fully deterministic—it fixes the Meta-World V3 task seed and a per-episode random seed for the perturbation—and an independent re-run reproduces every reported number exactly.

### 5.5 What the World-Action Model Adds

The world-action model does not require every method to implement the full tuple $K$. It asks a sharper set of review questions. For a VLA, which parts of $X$ are visible to the policy, and where are $I$, $\beta$, and $\Delta$ hidden? For a diffusion policy, are multimodal trajectories organized by equivalent object effects? For a TAMP system, are preconditions and effects grounded in perception rather than hand-coded symbols? For a world model, does prediction correspond to object-state transition or only to image similarity? For a benchmark, which links of the chain are actually tested?

This lens also resolves the ambiguous status of the survey itself. The paper is not merely a catalog of existing atomic-skill papers, because the object-centric world-action model is partly a unifying proposal. Nor is it a method paper, because it does not claim a new implemented algorithm. It is a framework survey: it reviews existing work by asking how each method contributes to a physically grounded world-action loop and where the loop still breaks.

## 6 Open Problems and Testable Hypotheses

The object-centric world-action model clarifies which problems remain open. The main gaps are not isolated architectural weaknesses; they are missing links in the world-action loop.

### 6.1 Learning Interfaces from Data

The hardest open problem is learning $I$, $\beta$, and $\Delta$ directly from interaction data. Hand-coded preconditions do not scale, but monolithic policies hide the boundaries needed for composition. A useful data-driven interface learner must infer when a skill is callable, when it has succeeded or failed, and what object-state distribution it produces. This requires datasets where object identity, contact events, and transition effects can be reconstructed. This problem cuts across all four stages. For discovery, the system must tell whether a proposed segment has a stable effect rather than merely a stable appearance. For learning, the policy must preserve the variables needed to monitor progress. For composition, the effect of one skill must be represented in the same state space as the initiation condition of the next. For recovery, abnormal post-states must be represented as valid starting points for repair skills rather than as out-of-distribution failures.

**Hypothesis H1.** For matched policy capacity and single-skill success, systems with explicit learned $I/\beta/\Delta$ interfaces should fail less often in deep chains than systems with only language-level subgoals. LIBERO, CALVIN, and ClevrSkills provide partial testbeds, but stronger tests should include contact, occlusion, and recovery (Liu et al., 2023; Mees et al., 2022; Haresh et al., 2024).

### 6.2 Unified Semantic, Metric, and Tactile State

Most current systems treat semantic, metric, and tactile grounding as separate capabilities. Object-centric skills require a shared state space in which object category, geometry, pose, compliance, friction, contact mode, and force limits constrain the same callable transition. This is especially important under occlusion. A robot cannot verify that an object has been inserted, latched, folded, or poured correctly if it loses object identity or lacks tactile evidence for hidden contact. Recent perception methods suggest partial remedies: video amodal segmentation hallucinates the occluded

extent of objects across frames (Chen et al., 2025a), and occlusion-robust reconstruction recovers complete, physically simulatable object geometry from partial views (Dong et al., 2026a); integrating such object-completion signals into the skill interface, so that $X$ and $C$ remain defined under occlusion, is an open problem. The difficulty is not only sensor fusion. Semantic state must preserve persistent object identity, metric state must preserve pose and tolerance under camera changes, and tactile/contact state must describe hidden physical events that vision may not observe. The field has strong semantic VLAs, improving spatial VLAs, and emerging tactile VLAs, but object-centric skills require these streams to agree on the same object instance and transition boundary.

**Hypothesis H2.** On insertion, assembly, deformable manipulation, and pouring, systems that jointly use $X_{\mathrm{met}}$ and $X_{\mathrm{tac}}$ should outperform semantic-only systems under perturbations in geometry, friction, compliance, and visibility. FurnitureBench and ManiSkill-ViTac are natural starting points (Heo et al., 2025; Li et al., 2024).

### 6.3 World Models over Object-State Transitions

World models should be evaluated by whether they predict manipulation-relevant transitions, not only by video realism or latent rollout stability. LeWorldModel, V-JEPA 2-AC, WorldVLA, VLA-JEPA, and AtomVLA each point toward predictive transition modeling (Maes et al., 2026; Assran et al., 2025; Cen et al., 2025; Sun et al., 2026a,b). The open question is how to align latent predictions with object-state predicates that can be used for composition and verification. Several related directions make the same point from different angles: image-editing subgoal methods generate visual futures, JEPA-style world models predict representation futures, and action-conditioned world models score candidate rollouts. The missing object-centric step is to convert these futures into interface variables: which object changed, which contact mode changed, whether a tolerance has been met, and whether the next skill is now callable.

**Hypothesis H3.** World models that predict object-state $\Delta$ should improve long-horizon manipulation more than models trained only for generic pixel or latent prediction. A controlled test would compare no world model, pixel future prediction, latent future prediction, and object-state transition prediction under the same downstream VLA or planner.

### 6.4 Recovery as First-Class Composition

Recovery is often treated as an engineering patch. In object-centric skills, recovery is the same problem as composition: after an unexpected state, find a skill whose initiation interface accepts that state and whose effect moves the system back toward the goal. This requires failure classifiers, state estimators, and skill libraries that include repair transitions, not only nominal task steps. This reframing changes what must be learned. A retry policy is not enough if the object has slipped, jammed, deformed, moved out of view, or entered the wrong contact mode. The system must classify the post-failure state, decide whether the original skill remains callable, and choose a repair skill whose own initiation interface matches the abnormal state.

**Hypothesis H4.** Systems whose recovery policies are represented as re-entry into an explicit interface graph should recover from contact-rich failures more reliably than systems that use only retry or human intervention. The test should include slip, partial insertion, occluded placement, deformable-object mismatch, and wrong-object interaction.

### 6.5 Benchmarks for the Full Chain

The field still lacks a benchmark that simultaneously stresses long horizons, object identity, metric precision, tactile grounding, verification, recovery, and cross-embodiment transfer. Without such a benchmark, progress will remain fragmented: VLA benchmarks will reward semantic generalization, assembly benchmarks will reward precision, tactile benchmarks will reward contact sensing, and failure benchmarks will reward recovery in isolation. The next generation of benchmarks should evaluate the entire object-centric skill chain. Concretely, such a benchmark should annotate or reconstruct intermediate object-state transitions, not just final success. It should include perturbations at each interface: wrong object identity, shifted geometry, hidden contact, force mismatch, partial completion, and recovery from abnormal states. It should also report chain depth, interface exposure, recovery success, and cross-embodiment transfer separately, so that a method is not rewarded for solving only the easiest link of the chain.

## 7 Conclusion

This survey argues that reliable robot manipulation requires more than large policies and semantic action labels. It requires an object-centric world-action model, PraxisWorld, in which atomic skills are the executable interfaces over world state: reusable transition units whose object state, contact topology, initiation condition, policy, termination, effect, and embodiment constraints can be learned, composed, verified, and repaired. The descriptor

$K = (R, S, X, C, I, \pi, \beta, \Delta, E)$ is the interface of that model and a common language for comparing options, TAMP, affordances, VLAs, diffusion policies, tactile methods, world models, benchmarks, and real-world systems.

The main message is that manipulation should be modeled as world state, action interface, and state transition, and that composition is therefore a physical interface problem. A skill chain is valid only when the predicted effect of one skill lands in the initiation interface of the next ($\Delta_i \models I_j$), and recovery is re-entry into the same interface graph after mismatch. Recent JEPA-style and action-conditioned world models make this direction especially timely: they are concrete instances of the world-action model that predict future latent or visual states, but the field still needs to align those predictions with explicit object-state transitions (Maes et al., 2026; Assran et al., 2025; Cen et al., 2025; Sun et al., 2026a,b). Bridging that gap is a concrete path from behavior imitation toward manipulation systems that are composable, verifiable, and recoverable across long-horizon physical interaction.

## Compliance With Ethics Guidelines

Yu-song HUANG, Yu-yan LI, Hong-liang LU, Hao-tian WANG, Xiao-yun QIU, Wen-peng XU, Yang-gang SHENG, Zhao-nian KUANG, Yi-jie CHEN, Jin-tao HE, Dong-ling LIU, and Xin-hu ZHENG declare that they have no conflict of interest.

## References

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J. Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as I can, not as I say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025. doi: 10.48550/arXiv.2506.09985.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pages 1726–1734, 2017.

Omer Benishu, Gal Fiebelman, and Sagie Benaim. PhyGenHOI: Physically-aware 4D generation of dynamic human-object interactions. *arXiv preprint arXiv:2605.30268*, 2026.

Timo Birr, Christoph Pohl, Abdelrahman Younes, and Tamim Asfour. AutoGPT+P: Affordance-based task planning with large language models. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

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, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. pi-0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024a. doi: 10.15607/rss.2025.xxi.010.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. SuSIE: Zero-shot robotic manipulation with pretrained image-editing diffusion models. In *International Conference on Learning Representations*, 2024b. arXiv:2310.10639.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, 2023a. arXiv:2307.15818.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, et al. RT-1: Robotics transformer for real-world control at scale. In *Robotics: Science and Systems (RSS)*, 2023b.

Wenxiao Cai, Iaroslav Ponomarenko, Jianhao Yuan, Xiaoqi Li, Wankou Yang, Hao Dong, and Bo Zhao. SpatialBot: Precise spatial understanding with vision language models. *arXiv preprint arXiv:2406.13642*, 2024a. ICRA 2025.

Yishuai Cai, Xinglin Chen, Yunxin Mao, Minglong Li, Shaowu Yang, Wenjing Yang, and Ji Wang. HBTP: Heuristic behavior tree planning with large language model reasoning. *arXiv preprint arXiv:2406.00965*, 2024b. ICRA 2025.

Frederick Callaway, Bas van Opheusden, Sayan Gul, Priyam Das, Paul M. Krueger, Thomas L. Griffiths, and Falk Lieder. Rational use of cognitive resources in human planning. *Nature Human Behaviour*, 6(8):1112–1125, 2022. doi: 10.1038/s41562-022-01332-8.

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

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, Deli Zhao, and Hao Chen. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Kaihua Chen, Deva Ramanan, and Tarasha Khurana. Using diffusion priors for video amodal segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Kejia Chen, Zheng Shen, Yue Zhang, Lingyun Chen, Fan Wu, Zhenshan Bing, Sami Haddadin, and Alois Christian Knoll. LEMMo-Plan: LLM-enhanced learning from multi-modal demonstration for planning sequential contact-rich manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11972–11978, 2025b.

Zixuan Chen, Ze Ji, Jing Huo, and Yang Gao. SCaR: Refining skill chaining for long-horizon robotic manipulation via dual regularization. In *Advances in Neural Information Processing Systems*, 2024.

Zixuan Chen, Junhui Yin, Yangtao Chen, Jing Huo, Pinzhuo Tian, Jieqi Shi, Yiwen Hou, Yinchuan Li, and Yang Gao. DeCo: Task decomposition and skill composition for zero-shot generalization in long-horizon 3d manipulation. *IEEE Robotics and Automation Letters*, 2026. arXiv:2505.00527.

Zhengxue Cheng, Yiqian Zhang, Wenkang Zhang, Haoyu Li, Keyu Wang, Li Song, and Hengdi Zhang. OmniVTLA: Vision-tactile-language-action model with semantic-aligned tactile sensing. *arXiv preprint arXiv:2508.08706*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems*, 2023. arXiv:2303.04137.

Carlos G. Correa, Mark K. Ho, Frederick Callaway, Nathaniel D. Daw, and Thomas L. Griffiths. Humans decompose tasks by trading off utility and computational cost. *PLOS Computational Biology*, 19(6):e1011087, 2023. doi: 10.1371/journal.pcbi.1011087.

Mark R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, 1989. doi: 10.1109/70.34763.

Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. RACER: Rich language-guided failure recovery policies for imitation learning. *arXiv preprint arXiv:2409.14674*, 2024. ICRA 2025.

Murtaza Dalal, Tarun Chiruvolu, Devendra Singh Chaplot, and Ruslan Salakhutdinov. Plan-seq-learn: Language model guided RL for solving long horizon robotics tasks. In *International Conference on Learning Representations*, 2024. arXiv:2405.01534.

Alan Dao, Dinh Bach Vu, and Bui Quang Huy. AlphaSpace: Enabling robotic actions through semantic tokenization and symbolic reasoning. *arXiv preprint arXiv:2503.18769*, 2025.

Jingwen Deng, Zihao Wang, Shaofei Cai, Anji Liu, and Yitao Liang. Open-world skill discovery from unsegmented demonstrations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Xin Dong, Weijian Deng, Lihan Zhang, Tianru Dai, Wenfeng Deng, and Yansong Tang. SAM3D-Phys: Towards multi-object interactive simulation in real world. *arXiv preprint arXiv:2605.30239*, 2026a.

Zibin Dong, Yicheng Liu, Shiduo Zhang, Baijun Ye, Yifu Yuan, Fei Ni, Jingjing Gong, Xipeng Qiu, Hang Zhao, Yinchuan Li, and Jianye Hao. ActionCodec: What makes for good action tokenizers. *arXiv preprint arXiv:2602.15397*, 2026b.

Kamil Dreczkowski, Pietro Vitiello, Vitalis Vosylius, and Edward Johns. Learning a thousand tasks in a day. *Science Robotics*, 10(108):eadv7594, 2025. doi: 10.1126/scirobotics.adv7594.

Shiying Duan, Pei Ren, Nanxiang Jiang, Zhengping Che, Jian Tang, Zhaoxin Fan, Yifan Sun, and wenjun wu. RoboPARA: Dual-arm robot planning with parallel allocation and recomposition across tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=b9GH5mMBfG.

Yiguo Fan, Pengxiang Ding, Shuanghao Bai, Xinyang Tong, Yuyang Zhu, Hongchao Lu, Fengqi Dai, Wei Zhao, Yang Liu, Siteng Huang, Zhaoxin Fan, Badong Chen, and Donglin Wang. Long-VLA: Unleashing long-horizon capability of vision language action model for robot manipulation. *arXiv preprint arXiv:2508.19958*, 2025. CoRL 2025.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Zeyu Feng, Hao Luan, Pranav Goyal, and Harold Soh. LTLDoG: Satisfying temporally-extended symbolic constraints for safe diffusion-based planning. *IEEE Robotics and Automation Letters*, 9(10):8571–8578, 2024a.

Zeyu Feng, Hao Luan, Kevin Yuchen Ma, and Harold Soh. DOPPLER: Diffusion meets options: Hierarchical generative skill composition for temporally-extended tasks. *arXiv preprint arXiv:2410.02389*, 2024b. ICRA 2025.

Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. FOCUS: Object-centric world models for robotics manipulation. *Frontiers in Neurorobotics*, 19:1585386, 2025.

Paul M. Fitts and Michael I. Posner. *Human Performance*. Brooks/Cole, 1967.

Kai Gao, Fan Wang, Erica Aduh, Dylan Randle, and Jane Shi. MuST: Multi-head skill transformer for long-horizon dexterous manipulation with skill progress. *arXiv preprint arXiv:2502.02753*, 2025. ICRA 2025.

Zeyu Gao, Yao Mu, Jinye Qu, Mengkang Hu, Shijia Peng, Chengkai Hou, Lingyue Guo, Ping Luo, Shanghang Zhang, and Yanfeng Lu. DAG-Plan: Generating directed acyclic dependency graphs for dual-arm cooperative planning. *arXiv preprint arXiv:2406.09953*, 2024.

Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. PDDLStream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 440–448, 2020.

Caelan Reed Garrett, Rohan Chitnis, Rachel Holladay, Beomjoon Kim, Tom Silver, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Integrated task and motion planning. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:265–293, 2021. doi: 10.1146/annurev-control-091420-084139.

Generalist AI Team. GEN-0: Embodied foundation models that scale with physical interaction. Generalist AI Blog, 2025. URL https://generalistai.com/blog/nov-04-2025-GEN-0.

Generalist AI Team. GEN-1: Scaling embodied foundation models to mastery. Generalist AI Blog, 2026. URL https://generalistai.com/blog/apr-02-2026-GEN-1.

Chen Geng, Guangzhao He, Yue Gao, Yunzhi Zhang, Shangzhe Wu, and Jiajun Wu. NeuROK: Generative 4D neural object kinematics. *arXiv preprint arXiv:2605.30347*, 2026.

James J. Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979.

Clemence Grislain, Hamed Rahimi, Olivier Sigaud, and Mohamed Chetouani. I-FailSense: Towards general robotic failure detection with vision-language models. *arXiv preprint arXiv:2509.16072*, 2025.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

Nolan B. Gutierrez and William J. Beksi. Movement primitives in robotics: A comprehensive survey. *arXiv preprint arXiv:2601.02379*, 2026.

Yuci Han and Alper Yilmaz. Enhancing policy learning with world-action model. *arXiv preprint arXiv:2603.28955*, 2026.

Ce Hao, Xuanran Zhai, Yaohua Liu, and Harold Soh. Abstracting robot manipulation skills via mixture-of-experts diffusion policies. *arXiv preprint arXiv:2601.21251*, 2026. ICLR 2026.

Sanjay Haresh, Daniel Dijkman, Apratim Bhattacharyya, and Roland Memisevic. ClevrSkills: Compositional language and visual reasoning in robotics. In *Advances in Neural Information Processing Systems*, 2024.

Benned Hedegaard, Yichen Wei, Ahmed Jaafar, Stefanie Tellex, George Konidaris, and Naman Shah. Beyond task and motion planning: Hierarchical robot planning with general-purpose skills. *arXiv preprint arXiv:2504.17901*, 2025.

Minho Heo, Youngwoon Lee, Doohyun Lee, and Joseph J. Lim. FurnitureBench: Reproducible real-world benchmark for long-horizon complex manipulation. *The International Journal of Robotics Research*, 44(10-11):1863–1891, 2025. doi: 10.1177/02783649241304789.

Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In *Conference on Robot Learning*, 2024. arXiv:2410.24090.

Carolina Higuera, Sergio Arnaud, Byron Boots, Mustafa Mukadam, Francois Robert Hogan, and Franziska Meier. Visuo-tactile world models. *arXiv preprint arXiv:2602.06001*, 2026.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=FFxkFMU89E.

Aoshen Huang, Jiaming Chen, Jiyu Cheng, Ran Song, Wei Pan, and Wei Zhang. SADiff: Skill-aware diffusion for generalizable robotic manipulation. *arXiv preprint arXiv:2601.11266*, 2026a.

Jialei Huang, Shuo Wang, Fanqi Lin, Yihang Hu, Chuan Wen, and Yang Gao. Tactile-VLA: Unlocking vision-language-action model's physical knowledge for tactile generalization. *arXiv preprint arXiv:2507.09160*, 2025a.

Suning Huang, Qianzhong Chen, Xiaohan Zhang, Jiankai Sun, and Mac Schwager. ParticleFormer: A 3D point cloud world model for multi-object, multi-material robotic manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2025b.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, Pierre Sermanet, Noah Brown, Tomas Jackson, Linda Luu, Sergey Levine, Karol Hausman, and Brian Ichter. Inner monologue: Embodied reasoning through planning with language models. In *arXiv preprint arXiv:2207.05608*, 2022.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning*, 2023. arXiv:2307.05973.

Wenlong Huang, Yu-Wei Chao, Arsalan Mousavian, Ming-Yu Liu, Dieter Fox, Kaichun Mo, and Li Fei-Fei. PointWorld: Scaling 3D world models for in-the-wild robotic manipulation. *arXiv preprint arXiv:2601.03782*, 2026b.

Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation*, 25(2):328–373, 2013. doi: 10.1162/NECO_a_00393.

Riccardo Andrea Izzo, Gianluca Bardaro, and Matteo Matteucci. BTGenBot: Behavior tree generation for robotic tasks with lightweight LLMs. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024. arXiv:2403.12761.

Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. doi: 10.1109/LRA.2020.2974707.

Youngjoon Jeong, Junha Chun, Soonwoo Cha, and Taesup Kim. Object-centric world model for language-guided manipulation. *arXiv preprint arXiv:2503.06170*, 2025.

Hanxiao Jiang, Binghao Huang, Ruihai Wu, Zhuoran Li, Shubham Garg, Hooshang Nayyeri, Shenlong Wang, and Yunzhu Li. RoboEXP: Action-conditioned scene graph via interactive exploration for robotic manipulation. In *Proceedings of the Conference on Robot Learning (CoRL)*, 2024.

Lars Johannsmeier, Samuel Schneider, Yanan Li, Etienne Burdet, and Sami Haddadin. A process-centric manipulation taxonomy for the organization, classification and synthesis of tactile robot skills. *Nature Machine Intelligence*, 7:916–927, 2025. doi: 10.1038/s42256-025-01045-3.

Daniel Kahneman, Anne Treisman, and Brian J. Gibbs. The reviewing of object files: Object-specific integration of information. *Cognitive Psychology*, 24(2):175–219, 1992.

Hossein Karami, Antony Thomas, and Fulvio Mastrogiovanni. TMP-IDAN: A task and motion planning framework using iteratively deepened AND/OR graph networks. *Robotics and Autonomous Systems*, 189:104943, 2025.

Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *CoRL Workshop on Learning Robot Fine and Dexterous Manipulation: Perception and Control*, 2024. URL https://openreview.net/forum?id=eOtDTS4iMQ.

Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. UniSkill: Imitating human videos via cross-embodiment skill representations. *arXiv preprint arXiv:2505.08787*, 2025. CoRL 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. CompILE: Compositional imitation learning and execution. In *International Conference on Machine Learning (ICML)*, 2019.

George Konidaris and Andrew Barto. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2009.

George Konidaris, Scott Kuindersma, Roderic Grupen, and Andrew Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, 31(3):360–375, 2012. doi: 10.1177/0278364911428653.

Longxin Kou, Fei Ni, Yan Zheng, Jinyi Liu, Yifu Yuan, Zibin Dong, and Jianye Hao. KISA: A unified keyframe identifier and skill annotator for long-horizon robotics demonstrations. In *Proceedings of the 41st International Conference on Machine Learning*, pages 25441–25474, 2024.

Norbert Krüger, Christopher Geib, Justus Piater, Ronald Petrick, Mark Steedman, Florentin Wörgötter, Aleš Ude, Tamim Asfour, Dirk Kraft, Damir Omrcen, Alejandro Agostini, and Rüdiger Dillmann. Object–action complexes: Grounded abstractions of sensory–motor processes. *Robotics and Autonomous Systems*, 59(10):740–757, 2011. doi: 10.1016/j.robot.2011.05.009.

Youngwoon Lee, Joseph J Lim, Anima Anandkumar, and Yuke Zhu. Adversarial skill chaining for long-horizon robot manipulation via terminal state regularization. In *Conference on Robot Learning*, 2021. arXiv:2111.07999.

Chengmeng Li, Junjie Wen, Yaxin Peng, Yan Peng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *IEEE Robotics and Automation Letters*, 11:2506–2513, 2026a. URL https://api.semanticscholar.org/CorpusID:284756287.

Chuanyu Li, Renjun Dang, Xiang Li, Zhiyuan Wu, Jing Xu, Hamidreza Kasaei, Roberto Calandra, Nathan Lepora, Shan Luo, Hao Su, and Rui Chen. ManiSkill-ViTac 2025: Challenge on manipulation skill learning with vision and tactile sensing. *arXiv preprint arXiv:2411.12503*, 2024.

Haozhan Li, Yuxin Zuo, Jiale Yu, Yuhao Zhang, Zhaohui Yang, Kaiyan Zhang, Xuekai Zhu, Yuchen Zhang, Tianxing Chen, Ganqu Cui, Dehui Wang, Dingxiang Luo, Yuchen Fan, Youbang Sun, Jia Zeng, Jiangmiao Pang, Shanghang Zhang, Yu Wang, Yao Mu, Bowen Zhou, and Ning Ding. SimpleVLA-RL: Scaling VLA training via reinforcement learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026b.

Jinming Li, Yichen Zhu, Zhibin Tang, Junjie Wen, Minjie Zhu, Xiaoyu Liu, Chengmeng Li, Ran Cheng, Yaxin Peng, Yan Peng, and Feifei Feng. CoA-VLA: Improving vision-language-action models via visual-textual chain-of-affordance. In *International Conference on Computer Vision*, 2025a. arXiv:2412.20451.

Runze Li, Hongyin Zhang, Junxi Jin, Qixin Zeng, Zifeng Zhuang, Yiqi Tang, Shangke Lyu, and Donglin Wang. World-value-action model: Implicit planning for vision-language-action systems. *arXiv preprint arXiv:2604.14732*, 2026c.

Xuanlin Li, Kyle Hsu, Jiayuan Gu, Oier Mees, Karl Pertsch, Homer Rich Walke, Chuyuan Fu, Ishikaa Lunawat, Isabel Sieh, Sean Kirmani, Sergey Levine, Jiajun Wu, Chelsea Finn, Hao Su, Quan Vuong, and Ted Xiao. Evaluating real-world robot manipulation policies in simulation. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 3705–3728. PMLR, 2025b. URL https://proceedings.mlr.press/v270/li25c.html.

Yaxuan Li, Zhongyi Zhou, Yefei Chen, Yanjiang Guo, Jiaming Liu, Shanghang Zhang, Jianyu Chen, and Yichen Zhu. Hi-WM: Human-in-the-world-model for scalable robot post-training. *arXiv preprint arXiv:2604.21741*, 2026d.

Yi Li, Yuquan Deng, Jesse Zhang, Joel Jang, Marius Memmel, Raymond Yu, Caelan Reed Garrett, Fabio Ramos, Dieter Fox, Anqi Li, Abhishek Gupta, and Ankit Goyal. HAMSTER: Hierarchical action models for open-world robot manipulation. In *International Conference on Learning Representations*, 2025c. arXiv:2502.05485.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation*, 2023. arXiv:2209.07753.

Zhixuan Liang, Yao Mu, Hengbo Ma, Masayoshi Tomizuka, Mingyu Ding, and Ping Luo. SkillDiffuser: Interpretable hierarchical planning via skill abstractions in diffusion-based task execution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. arXiv:2312.11598.

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

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems*, 2023.

Changyu Liu, Yiyang Liu, Taowen Wang, Qiao Zhuang, James Chenhao Liang, Wenhao Yang, Renjing Xu, Qifan Wang, Dongfang Liu, and Cheng Han. On-the-fly VLA adaptation via test-time reinforcement learning. *arXiv preprint arXiv:2601.06748*, 2026a.

Chaoqi Liu, Haonan Chen, Sigmund H. Høeg, Shaoxiong Yao, Yunzhu Li, Kris Hauser, and Yilun Du. Flexible multitask learning with factorized diffusion policy. *arXiv preprint arXiv:2512.21898*, 2025a.

Chenyv Liu, Wentao Tan, Lei Zhu, Fengling Li, Jingjing Li, Guoli Yang, and Heng Tao Shen. Self-correcting VLA: Online action refinement via sparse world imagination. *arXiv preprint arXiv:2602.21633*, 2026b.

Hanqing Liu, Jiahuan Long, Junqi Wu, Jiacheng Hou, Huili Tang, Tingsong Jiang, Weien Zhou, and Wenbiao Yao. Eva-VLA: Evaluating vision-language-action models' robustness under real-world physical variations. *arXiv preprint arXiv:2509.18953*, 2025b.

Yushan Liu, Peibo Sun, Shoujie Li, Yifan Xie, Lingfeng Zhang, Xintao Chao, Shiyuan Dong, Fang Chen, Xiao-Ping Zhang, and Wenbo Ding. OA-WAM: Object-addressable world action model for robust robot manipulation. *arXiv preprint arXiv:2605.06481*, 2026c.

Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. VLA-RL: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025a.

Guanxing Lu, Baoxiong Jia, Puhao Li, Yixin Chen, Ziwei Wang, Yansong Tang, and Siyuan Huang. GWM: Towards scalable gaussian world models for robotic manipulation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Jianlan Luo, Charles Xu, Jeffrey Wu, and Sergey Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105):eads5033, 2025. doi: 10.1126/scirobotics.ads5033. arXiv:2410.21845.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *Conference on Robot Learning (CoRL)*, 2020.

Baorui Ma, Jiahui Yang, Donglin Di, Xuancheng Zhang, Jianxun Cui, Hao Li, Yan Xie, and Wei Chen. MetricAnything: Scaling metric depth pretraining with noisy heterogeneous sources. *arXiv preprint arXiv:2601.22054*, 2026.

Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied AI. *arXiv preprint arXiv:2405.14093*, 2024.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. LeWorldModel: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026. doi: 10.48550/arXiv.2603.19312.

Weixin Mao, Weiheng Zhong, Zhou Jiang, Dong Fang, Zhongyue Zhang, Zihan Lan, Haosheng Li, Fan Jia, Tiancai Wang, Haoqiang Fan, and Osamu Yoshie. RoboMatrix: A skill-centric hierarchical framework for scalable robot task planning and execution in open-world. *arXiv preprint arXiv:2412.00171*, 2024a.

Xiaofeng Mao, Gabriele Giudici, Claudio Coppola, Kaspar Althoefer, Ildar Farkhatdinov, Zhibin Li, and Lorenzo Jamone. DexSkills: Skill segmentation using haptic data for learning autonomous long-horizon robotic manipulation tasks. *arXiv preprint arXiv:2405.03476*, 2024b. IROS 2024.

Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334, 2022.

Itamar Mishani, Yorai Shaoul, and Maxim Likhachev. MOSAIC: A skill-centric algorithmic framework for long-horizon manipulation planning. *arXiv preprint arXiv:2504.16738*, 2025.

Ruaridh Mon-Williams, Gen Li, Ran Long, Wenqian Du, and Christopher G Lucas. ELLMER: Embodied large language models enable robots to complete complex tasks in unpredictable environments. *Nature Machine Intelligence*, 7: 592–601, 2025.

Luis Montesano, Manuel Lopes, Alexandre Bernardino, and José Santos-Victor. Learning object affordances: From sensory–motor coordination to imitation. *IEEE Transactions on Robotics*, 24(1):15–26, 2008. doi: 10.1109/TRO.2007.914848.

Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. RoboCasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems*, 2024.

Fei Ni, Jianye Hao, Shiguang Wu, Longxin Kou, Jiashun Liu, Yan Zheng, Bin Wang, and Yuzheng Zhuang. Generate subgoal images before act: Unlocking the chain-of-thought reasoning in diffusion model for robot manipulation with multimodal prompts. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Kazuma Obata, Tatsuya Aoki, Takato Horii, Tadahiro Taniguchi, and Takayuki Nagai. LiP-LLM: Integrating linear programming and dependency graph with large language models for multi-robot task planning. *IEEE Robotics and Automation Letters*, 2024. arXiv:2410.21040.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Alexandros Paraschos, Christian Daniel, Jan Peters, and Gerhard Neumann. Probabilistic movement primitives. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. pi-0.5: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. pi-0.7: A steerable generalist robotic foundation model with emergent capabilities. Technical report, Physical Intelligence, April 2026. https://www.pi.website/blog/pi07.

Zenon W. Pylyshyn. Visual indexes, preconceptual objects, and situated vision. *Cognition*, 80(1–2):127–158, 2001.

Carl Qi, Xiaojie Wang, Silong Yong, Stephen Sheng, Huitan Mao, Sriram Srinivasan, Manikantan Nambi, Amy Zhang, and Yesh Dattatreya. Self-refining vision language model for robotic failure detection and reasoning. *arXiv preprint arXiv:2602.12405*, 2026.

Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, and Xuelong Li. SpatialVLA: Exploring spatial representations for visual-language-action model. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. SayPlan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *Conference on Robot Learning*, 2023. arXiv:2307.06135.

Allen Z. Ren, Justin Lidard, Lars L. Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. In *International Conference on Learning Representations*, 2025. arXiv:2412.12953.

Yifei Simon Shao, Yuchen Zheng, Sunan Sun, Pratik Chaudhari, Vijay Kumar, and Nadia Figueroa. SymSkill: Symbol and skill co-invention for data-efficient and reactive long-horizon manipulation. *arXiv preprint arXiv:2510.01661*, 2025. ICRA 2026.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, Adrian Li-Bell, Danny Driess, Lachy Groom, Sergey Levine, and Chelsea Finn. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025. ICML 2025.

Daniel Sliwowski and Dongheui Lee. ConditionNET: Learning preconditions and effects for execution monitoring. *IEEE Robotics and Automation Letters*, 10(2), 2025. arXiv:2502.01167.

Wenxuan Song, Jiayi Chen, Wenxue Li, Xu He, Han Zhao, Can Cui, Pengxiang Ding, Shiyan Su, Feilong Tang, Xuelian Cheng, Donglin Wang, Zongyuan Ge, Xinhu Zheng, Zhe Liu, Hesheng Wang, and Haoang Li. RationalVLA: A rational vision-language-action model with dual system. *arXiv preprint arXiv:2506.10826*, 2025.

Elizabeth S. Spelke. Principles of object perception. *Cognitive Science*, 14(1):29–56, 1990. doi: 10.1207/s15516709cog1401_3.

Jingwen Sun, Wenyao Zhang, Zekun Qi, Shaojie Ren, Zezhi Liu, Hanxin Zhu, Guangzhong Sun, Xin Jin, and Zhibo Chen. VLA-JEPA: Enhancing vision-language-action model with latent world model. *arXiv preprint arXiv:2602.10098*, 2026a. doi: 10.48550/arXiv.2602.10098.

Xiaoquan Sun, Zetian Xu, Chen Cao, Zonghe Liu, Yihan Sun, Jingrui Pang, Ruijian Zhang, Zhen Yang, Kang Pang, Dingxin He, Mingqi Yuan, and Jiayu Chen. AtomVLA: Scalable post-training for robotic manipulation via predictive latent world models. *arXiv preprint arXiv:2603.08519*, 2026b.

Yu Sun, Meng Cao, Ping Yang, Rongtao Xu, Yunxiao Yan, Runze Xu, Liang Ma, Roy Gan, Andy Zhai, Qingxuan Chen, Zunnan Xu, Hao Wang, Jincheng Yu, Lucy Liang, Qian Wang, Ivan Laptev, Ian D. Reid, and Xiaodan Liang. ManipArena: Comprehensive real-world evaluation of reasoning-oriented generalist robot manipulation. *arXiv preprint arXiv:2603.28545*, 2026c.

Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

Stefan Tabakov, Asen Popov, Dimitar Dimitrov, S Ensiye Kiyamousavi, Vladimir Hristov, and Boris Kraychev. Atomic action slicing: Planner-aligned options for generalist VLA agents. In *ACM/SIGAPP Symposium on Applied Computing*, 2026. arXiv:2512.11584.

Bingjie Tang, Michael A Lin, Iretiayo Akinola, Ankur Handa, Gaurav S Sukhatme, Fabio Ramos, Dieter Fox, and Yashraj S Narang. IndustReal: Transferring contact-rich assembly tasks from simulation to reality. In *Robotics: Science and Systems*, 2023.

Bingjie Tang, Iretiayo Akinola, Jie Xu, Bowen Wen, Ankur Handa, Karl Van Wyk, Dieter Fox, Gaurav S Sukhatme, Fabio Ramos, and Yashraj Narang. AutoMate: Specialist and generalist assembly policies over diverse geometries. In *Robotics: Science and Systems*, 2024.

Tutian Tang, Xingyu Ji, Wanli Xing, Ce Hao, Wenqiang Xu, Lin Shao, Cewu Lu, Qiaojun Yu, Jiangmiao Pang, and Kaifeng Zhang. MoDE-VLA: Towards human-like manipulation through RL-augmented teleoperation and mixture-of-dexterous-experts VLA. *arXiv preprint arXiv:2603.08122*, 2026.

Stone Tao, Fanbo Xiang, Arth Shukla, Yuzhe Qin, Xander Hinrichsen, Xiaodi Yuan, Chen Bao, Xinsong Lin, Yulin Liu, Tse-Kai Chan, Yuan Gao, Xuanlin Li, Tongzhou Mu, Nan Xiao, Arnav Gurha, Viswesh N, Yong Woo Choi, Yen-Ru Chen, Zhiao Huang, Roberto Calandra, Rui Chen, Shan Luo, and Hao Su. Maniskill3: GPU parallelized robot simulation and rendering for generalizable embodied AI. In *7th Robot Learning Workshop: Towards Robots with Human-Level Abilities*, 2025. URL https://openreview.net/forum?id=GgTxudXaU8.

Geraud Nangue Tasse, Devon Jarvis, Steven James, and Benjamin Rosman. Skill machines: Temporal logic composition in reinforcement learning. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022. URL https://openreview.net/forum?id=VJ6JTmW5dY.

Qiangxing Tian, Shanshan Zhang, Donglin Wang, Jinxin Liu, and Shuyu Yang. A graph-based skill composition framework for robot learning. *Robotics and Autonomous Systems*, 2024. doi: 10.1016/j.robot.2024.104787.

Yuxuan Tian, Yurun Jin, Bin Yu, Yukun Shi, Hao Wu, Chi Harold Liu, Kai Chen, and Cong Huang. STARRY: Spatial-temporal action-centric world modeling for robotic manipulation. *arXiv preprint arXiv:2604.26848*, 2026.

Chenrui Tie, Shengxiang Sun, Jinxuan Zhu, Yiwei Liu, Jingxiang Guo, Yue Hu, Haonan Chen, Junting Chen, Ruihai Wu, and Lin Shao. Manual2Skill: Learning to read manuals and acquire robotic skills for furniture assembly using vision-language models. *arXiv preprint arXiv:2502.10090*, 2025. RSS 2025.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Ruisen Tu, Arth Shukla, Sohyun Yoo, Xuanlin Li, Junxi Li, Jianwen Xie, Hao Su, and Zhuowen Tu. SG-VLA: Learning spatially-grounded vision-language-action models for mobile manipulation. *arXiv preprint arXiv:2603.22760*, 2026.

Shivam Vats, Devesh K. Jha, Maxim Likhachev, Oliver Kroemer, and Diego Romeres. RecoveryChaining: Learning local recovery policies for robust manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9776–9783, 2025. doi: 10.1109/IROS60139.2025.11245856.

Weikang Wan, Yifeng Zhu, Rutav Shah, and Yuke Zhu. LOTUS: Continual imitation learning for robot manipulation through unsupervised skill discovery. In *IEEE International Conference on Robotics and Automation*, 2024. arXiv:2311.02058.

Weikang Wan, Jiawei Fu, Xiaodi Yuan, Yifeng Zhu, and Hao Su. LodeStar: Long-horizon dexterity via synthetic data augmentation from human demonstrations. *arXiv preprint arXiv:2508.17547*, 2025. CoRL 2025.

Siyin Wang et al. World action models: The next frontier in embodied AI. *arXiv preprint arXiv:2605.12090*, 2026.

Yating Wang, Haoyi Zhu, Mingyu Liu, Jiange Yang, Hao-Shu Fang, and Tong He. VQ-VLA: Improving vision-language-action models via scaling vector-quantized action tokenizers. *arXiv preprint arXiv:2507.01016*, 2025. ICCV 2025.

Philipp Wu, Yide Shentu, Qiayuan Liao, Ding Jin, Menglong Guo, Koushil Sreenath, Xingyu Lin, and Pieter Abbeel. RoboCopilot: Human-in-the-loop interactive imitation learning for robot manipulation. *arXiv preprint arXiv:2503.07771*, 2025.

Yi Wu, Zikang Xiong, Yiran Hu, Shreyash Iyengar, Nan Jiang, Aniket Bera, Lin Tan, and Suresh Jagannathan. SELP: Generating safe and efficient task plans for robot agents with large language models. *arXiv preprint arXiv:2409.19471*, 2024. ICRA 2025 Best Paper Finalist.

X Square Robot. WALL-WM: Carving world action modeling at the event joints. Technical report and open-source release, 2026. https://github.com/X-Square-Robot/wall-x.

Zhou Xian, Nikolaos Gkanatsios, Theophile Gervet, Tsung-Wei Ke, and Katerina Fragkiadaki. ChainedDiffuser: Unifying trajectory diffusion and keypose prediction for robotic manipulation. In *Proceedings of the 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 2323–2339, 2023.

Feng Xu, Guangyao Zhai, Xin Kong, Tingzhong Fu, Daniel F. N. Gordon, Xueli An, and Benjamin Busam. STARE-VLA: Progressive stage-aware reinforcement for fine-tuning vision-language-action models. *arXiv preprint arXiv:2512.05107*, 2025.

Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. XSkill: Cross embodiment skill discovery. In *Conference on Robot Learning*, 2023.

Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, and Alex Wong. Binding touch to everything: Learning unified multimodal tactile representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. arXiv:2401.18084.

Ran Yang, Zijian An, Lifeng Zhou, and Yiming Feng. SeqVLA: Sequential task execution for long-horizon manipulation with completion-aware vision-language-action model. *arXiv preprint arXiv:2509.14138*, 2025.

Yue Yang, Shuo Cheng, Yu Fang, Homanga Bharadhwaj, Mingyu Ding, Gedas Bertasius, and Daniel Szafir. LiLo-VLA: Compositional long-horizon manipulation via linked object-centric policies. *arXiv preprint arXiv:2602.21531*, 2026a.

Zhaoyang Yang, Yurun Jin, Lizhe Qi, Cong Huang, and Kai Chen. EA-WM: Event-aware generative world model with structured kinematic-to-visual action fields. *arXiv preprint arXiv:2605.06192*, 2026b.

Jinhui Ye, Fangjing Wang, Ning Gao, Junqiu Yu, Zhu Yangkun, Bin Wang, Jinyu Zhang, Weiyang Jin, Yanwei Fu, Feng Zheng, Yilun Chen, and Jiangmiao Pang. Spatially guided training for vision-language-action model. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=eKhOrQWAVJ.

Zewei Ye, Weifeng Lu, Minghao Ye, Tao Lin, Shuo Yang, Junchi Yan, and Bo Zhao. RoboFAC: A comprehensive framework for robotic failure analysis and correction. *arXiv preprint arXiv:2505.12224*, 2025.

Chunlin Yu, Hanqing Wang, Ye Shi, Haoyang Luo, Sibei Yang, Jingyi Yu, and Jingya Wang. SeqAfford: Sequential 3D affordance reasoning via multimodal large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a.

Jiawen Yu, Hairuo Liu, Qiaojun Yu, Jieji Ren, Ce Hao, Haitong Ding, Guangyu Huang, Guofan Huang, Yan Song, Panpan Cai, Wenqiang Zhang, and Cewu Lu. ForceVLA: Enhancing VLA models with a force-aware MoE for contact-rich manipulation. In *Advances in Neural Information Processing Systems*, 2025b. URL https://openreview.net/forum?id=2845H8Ua5D.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1094–1100. PMLR, 2020.

Tianyuan Yuan, Yicheng Liu, Chenhao Lu, Zhuoguang Chen, Tao Jiang, and Hang Zhao. DepthVLA: Enhancing vision-language-action models with depth-aware spatial reasoning. *arXiv preprint arXiv:2510.13375*, 2025.

Michal Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. *arXiv preprint arXiv:2407.08693*, 2024.

Xuanran Zhai, Zekai Huang, Longyan Wu, Qianyou Zhao, Qiaojun Yu, Jieji Ren, Ce Hao, and Harold Soh. SkillVLA: Tackling combinatorial diversity in dual-arm manipulation via skill reuse. *arXiv preprint arXiv:2603.03836*, 2026.

Borong Zhang, Jiahao Li, Jiachen Shen, Yishuai Cai, Yuhao Zhang, Yuanpei Chen, Juntao Dai, Jiaming Ji, and Yaodong Yang. VLA-Arena: An open-source framework for benchmarking vision-language-action models. *arXiv preprint arXiv:2512.22539*, 2025a.

Chaofan Zhang, Peng Hao, Xiaoge Cao, Xiaoshuai Hao, Shaowei Cui, and Shuo Wang. VTLA: Vision-tactile-language-action model with preference learning for insertion manipulation. *arXiv preprint arXiv:2505.09577*, 2025b.

Hongyin Zhang, Shuo Zhang, Junxi Jin, Qixin Zeng, Runze Li, and Donglin Wang. RobustVLA: Robustness-aware reinforcement post-training for vision-language-action models. *arXiv preprint arXiv:2511.01331*, 2025c.

Kaidi Zhang, Heng Zhang, Zhengtong Xu, Zhiyuan Zhang, Md Rakibul Islam Prince, Xiang Li, Xiaojing Han, Yuhao Zhou, Arash Ajoudani, and Yu She. TacVLA: Contact-aware tactile fusion for robust vision-language-action manipulation. *arXiv preprint arXiv:2603.12665*, 2026a.

Kaifeng Zhang, Baoyu Li, Kris Hauser, and Yunzhu Li. AdaptiGraph: Material-adaptive graph-based neural dynamics for robotic manipulation. In *Robotics: Science and Systems (RSS)*, 2024a.

Li Zhang, Xianhui Meng, Liu Liu, Haonan Jiang, Jianan Wang, Rujing Wang, Cewu Lu, Jun Liu, and Hong Zhang. Probing effective and efficient category-level articulated object pose perception. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–18, 2026b. doi: 10.1109/TPAMI.2026.3682504.

Likui Zhang, Tao Tang, Zhihao Zhan, Xiuwei Chen, Zisheng Chen, Jianhua Han, Jiangtong Zhu, Pei Xu, Hang Xu, Hefeng Wu, Liang Lin, and Xiaodan Liang. AtomicVLA: Unlocking the potential of atomic skill learning in robots. *arXiv preprint arXiv:2603.07648*, 2026c. CVPR 2026.

Mingtong Zhang, Kaifeng Zhang, and Yunzhu Li. Dynamic 3D gaussian tracking for graph-based neural dynamics modeling. In *Conference on Robot Learning (CoRL)*, 2024b.

Peng-Fei Zhang, Ying Cheng, Xiaofan Sun, Shijie Wang, Fengling Li, Lei Zhu, and Heng Tao Shen. A step toward world models: A survey on robotic manipulation. *arXiv preprint arXiv:2511.02097*, 2025d.

Siyang Zhang, Bin Li, Jingtao Qi, Xueying Wang, Fu Li, Jianan Wang, En Zhu, and Jinjing Sun. Code-BT: A code-driven approach to behavior tree generation for robot tasks planning with large language models. In *International Joint Conference on Artificial Intelligence*, pages 8814–8822, 2025e.

Zhanguang Zhang, Zhiyuan Li, Behnam Rahmati, Rui Heng Yang, Yintao Ma, Amir Rasouli, Sajjad Pakdamansavoji, Yangzheng Wu, Lingfeng Zhang, Tongtong Cao, Feng Wen, Xinyu Wang, Xingyue Quan, and Yingxue Zhang. Do world action models generalize better than VLAs? A robustness study. *arXiv preprint arXiv:2603.22078*, 2026d.

Haibo Zhao, Yu Qi, Boce Hu, Yizhe Zhu, Ziyan Chen, Heng Tian, Xupeng Zhu, Owen Howell, Haojie Huang, Robin Walters, Dian Wang, and Robert Platt. Generalizable hierarchical skill learning via object-centric representation. *arXiv preprint arXiv:2510.21121*, 2025a.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. CoT-VLA: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1702–1713, 2025b. doi: 10.1109/CVPR52734.2025.00166.

Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023. arXiv:2304.13705.

Zhigen Zhao, Shuo Cheng, Yan Ding, Ziyi Zhou, Shiqi Zhang, Danfei Xu, and Ye Zhao. A survey of optimization-based task and motion planning: From classical to learning approaches. *IEEE Transactions on Automation Science and Engineering*, 2024. arXiv:2404.02817.

Jinliang Zheng, Jianxiong Li, Dongxiu Liu, Yinan Zheng, Zhihao Wang, Zhonghong Ou, Yu Liu, Jingjing Liu, Ya-Qin Zhang, and Xianyuan Zhan. UniAct: Universal actions for enhanced embodied foundation models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. arXiv:2501.10105.

Zhide Zhong, Haodong Yan, Junfeng Li, Xiangchen Liu, Xin Gong, Tianran Zhang, Wenxuan Song, Jiayi Chen, Xinhu Zheng, Hesheng Wang, and Haoang Li. FlowVLA: Visual chain of thought-based motion reasoning for vision-language-action models. *arXiv preprint arXiv:2508.18269*, 2025.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning (ICML)*, 2025a.

Jiacong Zhou, Jiaxu Miao, Xianyun Wang, and Jun Yu. Enhancing LLM planning for robotics manipulation through hierarchical procedural knowledge graphs. In *Advances in Neural Information Processing Systems*, 2025b.

Jiayu Zhou, Qiwei Wu, Jian Li, Zhe Chen, Xiaogang Xiong, and Renjing Xu. Gentle manipulation policy learning via demonstrations from VLM planned atomic skills. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18855–18863, 2026. doi: 10.1609/aaai.v40i22.38955.

Zihan Zhou, Animesh Garg, Dieter Fox, Caelan Garrett, and Ajay Mandlekar. SPIRE: Synergistic planning, imitation, and reinforcement learning for long-horizon manipulation. In *Conference on Robot Learning*, 2024. arXiv:2410.18065.

Chongyu Zhu, Mithun Vanniasinghe, Jiayu Chen, and Chi-Guhn Lee. LOKI: Offline discovery of interpretable skills from multi-task trajectories. *arXiv preprint arXiv:2602.01018*, 2026.

Yifeng Zhu, Peter Stone, and Yuke Zhu. Bottom-up skill discovery from unsegmented demonstrations for long-horizon robot manipulation. *IEEE Robotics and Automation Letters*, 7(2):4126–4133, 2022. doi: 10.1109/lra.2022.3146589.

Yihang Zhu, Weiqing Wang, Shijie Wu, Ye Shi, and Jingya Wang. Learning semantic atomic skills for multi-task robotic manipulation. *arXiv preprint arXiv:2512.18368*, 2025.