OSTRA: LEARNING MULTI-STAGE ROBOT MANIPULATION THROUGH OBJECT-STATE TRANSITION SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sparse supervision is a key challenge in multi-stage manipulation, with the lack of supervision over the past-to-future evolution of objects required by the policy being particularly critical. To address this issue, we introduce the Object-State Transition Map (**OST-Map**), a unified object-centric representation that organizes task-relevant objects as identity-consistent graphs and represents historical and future object states in a shared structured space. Building on OST-Map, we propose **OSTRA**, a diffusion-based Object-State Transition Reasoning Architecture that hierarchically predicts future object states, target TCP poses, and continuous action trajectories. OSTRA combines Transition-Guided Attention, which prioritizes historical object-state transitions during future-state prediction, with Staged Denoising, which progressively translates predicted object changes into robot subgoals and executable actions. Experiments on three simulation benchmarks and seven diverse real-world tasks demonstrate consistent improvements over representative imitation learning baselines. Further ablations validate the contributions of OST-Map and the proposed architectural designs.

1 INTRODUCTION

Sparse supervision is a key challenge in imitation learning, particularly for multi-stage robot manipulation tasks (Zare et al., 2024; Mandlekar et al., 2020; Chen et al., 2023; Jain et al., 2025). Completing such tasks requires a policy not only to understand the environment at the current time step, but also to retain knowledge of the past—for example, whether prior subtasks succeeded and how key objects changed (Figure 1, left) (Kang & Kuo, 2025; Mark et al., 2026). Because an environment consists of objects, the policy must understand how these objects have changed over time. Therefore, the currently observed environment alone is insufficient for the policy to predict the most appropriate action. The policy must also be aware of object changes within the environment. We refer to this challenge as the object-state transition supervision gap.

Early imitation learning methods improve short-term action prediction through action chunking or limited observation windows, but do not explicitly capture long-horizon task-relevant state transitions (Zhao et al., 2023; Chi et al., 2025). Recent approaches incorporate past information through observation histories, latent memories, or task progress (Liang & Boularias, 2021; Shi et al., 2026; Mark et al., 2026). Others use future observations, goal states, or latent plans as additional supervision (Lynch et al., 2020; Mandlekar et al., 2020; Belkhale & Sadigh, 2023). However, these methods represent temporal information at observation, latent, or task level, providing limited supervision on individual-object evolution during execution. Our method bridges this gap by unifying both temporal directions at the object level. Specifically, it represents past object-state transitions as structured historical context and predicts future object states as auxiliary supervision, enabling the policy to reason about how task-relevant objects have changed and how they should evolve next.

To provide such supervision over past-to-future object evolution, we introduce the **O**bject-**S**tate **T**ransition Map (OST-Map), an object-centric representation of observed and future states in a shared structured space. OST-Map avoids requiring policies to discover transition-relevant cues from visual histories using robot-centric action supervision alone. Specifically, OST-Map represents each task-relevant object as an identity-consistent graph of structural nodes. Each node encodes

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

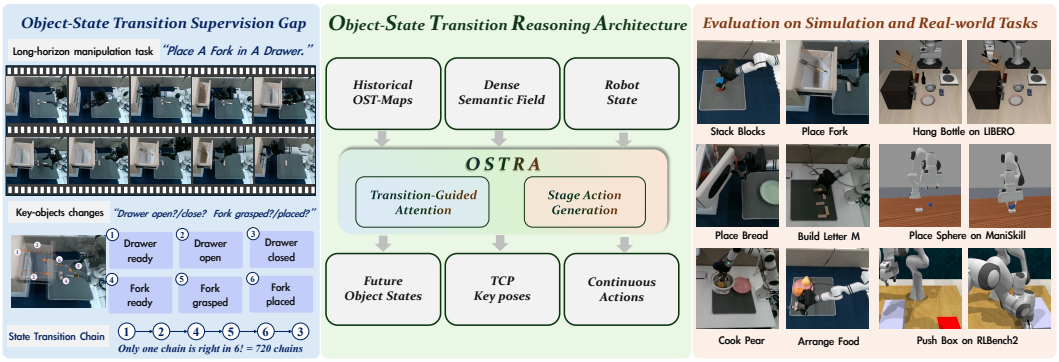


Figure 1: **Overview of OSTRA.** Multi-stage manipulation requires resolving the object-state transition supervision gap: the policy must infer the correct transition chain from historical object changes. OSTRA combines historical OST-Maps, a dense Semantic Field, and the robot state to predict future object states, TCP keyposes, and continuous actions, and is evaluated across simulation and real-world tasks, where explicit transition supervision supports long-horizon control.

its semantic identity, geometric parameters, and 6-DoF pose, while the graph edges capture structural and spatial relations among nodes and objects. By preserving object and node identities across time, OST-Map organizes observations into temporally ordered object-centric tokens that explicitly characterize how task-relevant objects evolve. It further represents future object states in the same structured space, allowing the transition from historical to future object states to be directly supervised. In this way, OST-Map turns otherwise implicit task progress and anticipated object changes into explicit intermediate targets between visual perception and robot action generation.

Building on OST-Map, we introduce OSTRA, an **O**bject-**S**tate **T**ransition **R**easoning **A**rchitecture for multi-stage manipulation (Figure 1, middle). OSTRA is a diffusion-based framework that hierarchically predicts future object states, target TCP poses, and continuous action trajectories. It incorporates two key designs. First, Transition-Guided Attention attends to historical OST-Map tokens before incorporating dense Semantic Field tokens, encouraging the model to prioritize task-relevant object-state transitions when predicting future states. Second, Staged Denoising follows an object-to-TCP-to-action hierarchy that progressively translates predicted object-state changes into robot subgoals and executable actions. Together, these designs connect object-level transition reasoning with robot-level planning and continuous control, enabling the policy to reason from how objects have changed to how they should evolve next and which actions should realize those changes.

We evaluate OSTRA on three simulation benchmarks and seven diverse real-world tasks. Across these settings, OSTRA consistently improves over representative imitation learning baselines on multi-stage manipulation tasks. Further ablation studies validate the contributions of OST-Map, Transition-Guided Attention, and Staged Denoising, demonstrating the effectiveness of structured object-state transition supervision for connecting scene evolution with robot action generation.

Our contributions are: (1) To tackle the object-state transition supervision gap in multi-stage imitation learning, we introduce OST-Map, an object-centric representation that aligns historical object-state sequences with future object-state targets, enabling direct supervision of object-state transitions. (2) Building on OST-Map, we introduce OSTRA, a diffusion-based architecture with two key designs. a) Transition-Guided Attention applies cross-attention to OST-Map tokens before Semantic Field tokens, encouraging the model to prioritize object-state transition information. b) Staged Denoising hierarchically predicts future object states, TCP poses, and action trajectories, progressively translating object-level state changes into robot subgoals and continuous control. (3) We conduct evaluations across three simulation benchmarks and seven diverse real-world tasks, demonstrating consistent performance gains over representative imitation learning baselines.

2 Related Work

2.1 Long-Horizon Visuomotor Imitation Learning

Long-horizon manipulation requires coherent behavior over extended execution. Existing methods shorten the control horizon through action chunking, autoregressive prediction, or diffusion-based

action generation (Zhao et al., 2023; Chi et al., 2025; Haldar et al., 2024; Reuss et al., 2024). Other approaches introduce higher-level task structure through subgoals, skill composition, progress estimation, or planning (Shiarlis et al., 2018; Gupta et al., 2020; Chen et al., 2023; Kang & Kuo, 2025; Jain et al., 2025; Liu et al., 2026; Zhang et al., 2026b; Feng et al., 2025; Zhou et al., 2026). These methods organize actions or task stages over time, but generally do not supervise how task-relevant objects evolve from historical to target states.

2.2 Temporal Modeling in Robot Policies

Temporal context is commonly incorporated through observation histories, latent memories, recurrent features, or retrieved past frames (Liang & Boularias, 2021; Fang et al., 2025; Shi et al., 2026; Mark et al., 2026). Recent policies further maintain recurrent or hierarchical memory to capture dependencies beyond the current observation (Ji et al., 2026; Xiao et al., 2026; Li et al., 2026).

A complementary direction uses future information as guidance. Existing methods condition policies on goal observations or latent plans (Lynch et al., 2020; Mandlekar et al., 2020), predict future visual states or joint video-action trajectories (Hu et al., 2025; Zhu et al., 2025), or learn predictive representations from future transitions and interaction targets (Zeng et al., 2024). Despite using historical and future context, these approaches represent time through holistic observations, latent memories, predicted frames, or task-level targets rather than identity-consistent state transitions of individual objects.

2.3 Object-Centric Representations for Manipulation

Object-centric policies organize perception and action around manipulable entities. Existing approaches use object proposals or 3D object features as policy inputs (Zhu et al., 2023b;a), learn affordance-, flow-, or contact-based interaction representations (Belkhale & Sadigh, 2023; Xu et al., 2025; Zhang et al., 2026a; Lee & Fazeli, 2025), or represent manipulation through explicit object motion, including SE(3) pose trajectories and object pointflow (Hsu et al., 2025; Yang et al., 2025). Recent work also introduces structured object-level representations based on physical concepts or semantic operations (Wei et al., 2026; Tang et al., 2026).

Object-centric world models use entity-level representations for dynamics and future prediction (Ferraro et al., 2025; Jeong et al., 2025), while graph-based approaches preserve explicit objects and relations for manipulation planning (Zhu et al., 2026). These methods typically represent objects as perceptual units, interaction interfaces, motion variables, or future prediction targets. In contrast, OST-Map maintains identity-consistent structural states across observed and future task stages, making past-to-future object-state transitions explicit supervision for action generation.

3 Method

3.1 Overview

Problem Formulation. We consider multi-stage robot manipulation from RGB-D observations. At time t, the policy receives an observation window $\mathcal{O}_t = \{(I_\tau, D_\tau, s_\tau)\}_{\tau=t-T_o+1}^{t}$ and predicts an action chunk $A_t = \{a_{t+k}\}_{k=0}^{H_a-1}$. Rather than directly mapping the current visual observation to robot actions, OSTRA explicitly models how task-relevant objects have evolved throughout the observation history and which object states should be achieved next.

Object-State Transition Modeling. As shown in Figure 2, each RGB-D observation is converted into an OST-Map whose identity-consistent nodes encode object semantics, structure, affordance, and 6-DoF pose. Tracking yields aligned historical maps that expose object-state transitions and supervise future states in the same structured space, while a dense Semantic Field preserves local scene evidence.

Transition-Guided Action Generation. Transition-Guided Attention first grounds prediction queries in historical OST-Map tokens and then incorporates dense Semantic Field tokens. Staged Action Generation follows the hierarchy *object state* $\rightarrow$ *TCP subgoal* $\rightarrow$ *action*: future OST-Map keyframes condition TCP-keyframe denoising, whose representation then conditions continuous-action generation.

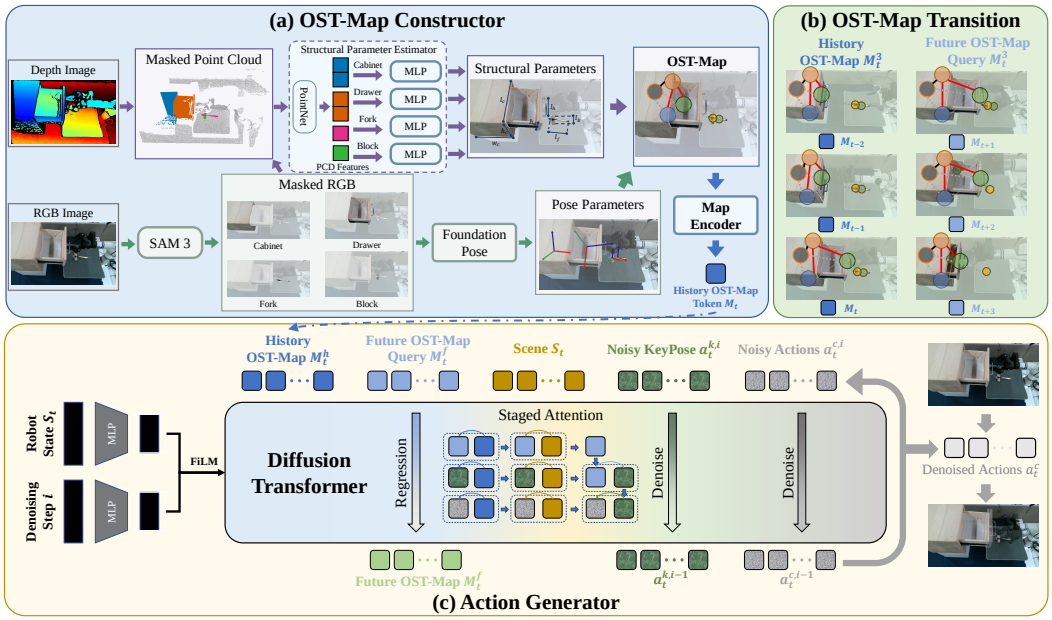


Figure 2: **Overview of OSTRA. (a) OST-Map Constructor:** RGB-D observations are converted into identity-consistent object-centric maps by estimating task-relevant object masks, structural parameters, and 6-DoF poses. **(b) Object-State Transition Modeling:** temporally aligned historical OST-Maps provide explicit object-state transition context and are used to query future object states. **(c) Transition-Guided Action Generation:** historical OST-Map tokens, future object-state queries, dense Semantic Field tokens, robot states, and the diffusion timestep are processed by OSTRA, which progressively predicts future object states, TCP keyposes, and continuous actions following the hierarchy *object state* $\rightarrow$ *TCP subgoal* $\rightarrow$ *action* for structured multi-stage control.

3.2 OST-Map: Object-State Transition Map

As illustrated in Figure 3, OST-Map composes parameterized primitives into object parts and identity-consistent structural nodes. Persistent node identities preserve fine-grained structure and enable temporal tracking.

For a task with N structural nodes, the OST-Map at time t is defined as an attributed graph $\mathcal{M}_t = (\mathcal{V}_t, \mathcal{E}_t)$. Each node $v_{t,i} \in \mathcal{V}_t$ is represented as

$$v_{t,i} = \big(c_i, \mathcal{P}_i, \mathcal{A}_i, p_{t,i}, q_{t,i}\big), \tag{1}$$

where c_i denotes its semantic identity, $\mathcal{P}_i$ and $\mathcal{A}_i$ denote its structural and affordance point sets, and $(p_{t,i}, q_{t,i}) \in \mathbb{R}^3 \times \mathbb{S}^3$ denotes its 6-DoF pose. Semantic identity and local structural attributes remain fixed within an episode, while pose evolves with the object state.

Each edge $e_{t,ij} \in \mathcal{E}_t$ stores its relation type, parameters, reference anchors, and relative pose. Free-relation edges connect pairs without task-specific relations. The sequence $\mathcal{M}_{t-T_o+1:t}$ captures object-state evolution in an identity-aligned space.

3.3 OSTRA: Object-State Transition Reasoning Architecture

OST-Map Construction and Transition Targets. As shown in Figure 2(a), OSTRA first constructs an OST-Map from each RGB-D observation. SAM 3 (Carion et al., 2026) segments prompted task objects, FoundationPose (Wen et al., 2024) registers and tracks their 6-DoF poses, and PointNet (Charles et al., 2017) with object-specific MLP heads estimates structural parameters from masked point clouds, producing a temporally aligned map sequence.

Historical OST-Maps encode observed transitions. During training, the structural-node positions at the next H_k task keyframes define future-state targets,

$$Y_t^M = \{p^\star_{t,h,i}\}_{h=1,i=1}^{H_k,N}. \tag{2}$$

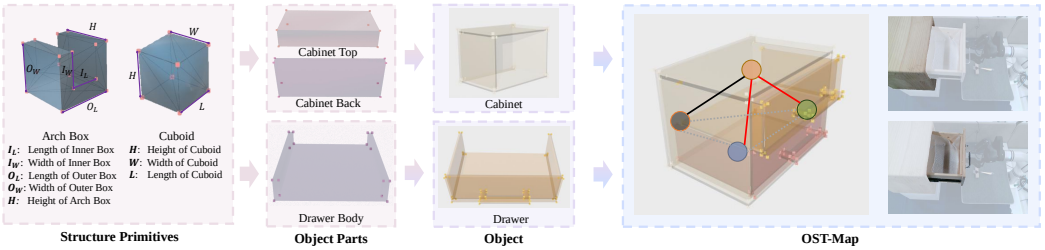


Figure 3: **Representation of OST-Map.** Parameterized structure primitives are composed into object parts and further assembled into task-relevant objects. Each object is represented by identity-consistent structural nodes and their relations, while node poses evolve with manipulation. The resulting OST-Map provides a compact object-centric representation of scene structure and object-state transitions for temporally grounded, multi-stage action generation.

paired with TCP keyframes Y_t^E and action chunk A_t. Future observations provide training-only supervision, while the shared OST-Map space supplies identity-aligned transition targets.

Multimodal Context Encoding. OSTRA combines a sparse OST-Map with a dense Semantic Field. Depth points $X_t = \{x_{t,j}\}_{j=1}^{P}$ are paired with image-aligned pretrained DINO features $d_{t,j}$. PointNet++ encodes $\mathrm{concat}(x_{t,j}, d_{t,j})$ into dense tokens F_t^S for cross-attention and downsampled tokens $\bar{F}_t^S$ for compact planning context.

For each historical map, node embeddings combine PointNet features of $\mathcal{P}_i$ and $\mathcal{A}_i$ with semantic, pose, temporal, and identity embeddings. Edge embeddings encode relation attributes and relative pose. A graph transformer produces coordinate-aware map tokens F_t^M. Map and scene tokens retain 3D coordinates for relative spatial attention. Robot state and diffusion timestep condition prediction blocks through adaptive normalization.

Transition-Guided Attention. A central challenge is the large imbalance between the sparse OST-Map tokens and the dense Semantic Field tokens. Directly mixing them in a single attention context can weaken the structured transition cues carried by the OST-Map. OSTRA therefore introduces *Transition-Guided Attention*, in which each prediction query first attends to historical OST-Map tokens and subsequently to the dense Semantic Field:

$$\widetilde{Z} = \mathrm{CA}(Z, F^M), \qquad Z^+ = \mathrm{CA}(\widetilde{Z}, F^S), \tag{3}$$

where CA denotes cross-attention. Cross-attention uses relative 3D rotary encodings computed from query and context coordinates.

For future object-state prediction, queries repeat the latest structural-node coordinates over the next H_k keyframes and augmenting them with learned keyframe-time and node-identity embeddings. TCP and action queries come from their noisy diffusion variables. The map-first order preserves transition grounding before introducing denser visual context.

Algorithm 1 Staged Action Generation in OSTRA

Require: Map context F^M, scene context F^S, robot state s_t
Ensure: Action chunk $\widehat{A}_t$
1: Construct future object queries Z^M
2: Initialize $Y^{E,N}, A^N \sim \mathcal{N}(0, I)$
3: **for** $n = N, \ldots, 1$ **do**
4: $\quad P^M, \widehat{Y}_n^M \leftarrow \textsc{ObjectStage}(Z^M, F^M, F^S, s_t, n)$
5: $\quad P^E, \widehat{\epsilon}_n^E \leftarrow \textsc{TCPStage}(Y^{E,n}, F^M, F^S, \mathrm{sg}(P^M), s_t, n)$
6: $\quad P^A, \widehat{\epsilon}_n^A, \widehat{G}_n \leftarrow \textsc{ActionGenerator}(A^n, F^M, F^S, \mathrm{sg}(P^E), s_t, n)$
7: $\quad Y^{E,n-1} \leftarrow \textsc{Denoise}(Y^{E,n}, \widehat{\epsilon}_n^E, n)$
8: $\quad A^{n-1} \leftarrow \textsc{Denoise}(A^n, \widehat{\epsilon}_n^A, n)$
9: **end for**
10: **return** A^0 with predicted gripper states

Staged Action Generation. As illustrated in Figure 2(c), OSTRA translates object-state transitions into executable robot motion through the hierarchy *object state* $\rightarrow$ *TCP subgoal* $\rightarrow$ *action*. At each

Table 1: **Summary of the main simulation results.** We report average task success rate (%) on each benchmark. Full per-task results are in Appendix A.4. **Bold** and underlining denote the best and second-best results, respectively, under the benchmark-specific evaluation protocols.

RLBench2 (James et al., 2020) (7 tasks)		ManiSkill (Gu et al., 2023) (6 tasks)		LIBERO (Liu et al., 2023) (40 tasks)	
Method	Mean S.R. (%)	Method	Mean S.R. (%)	Method	Mean S.R. (%)
ACT (Zhao et al., 2023)	15.5	ACT (Zhao et al., 2023)	21.3	ACT (Zhao et al., 2023)	67.7
DP3 (Ze et al., 2024)	26.0	Diffusion Policy (Chi et al., 2025)	26.8	GROOT (Zhu et al., 2023a)	66.2
3D Diffuser Actor (Ke et al., 2025)	64.7	DP3 (Ze et al., 2024)	53.8	Diffusion Policy (Chi et al., 2025)	72.4
PerAct^2 (Shridhar et al., 2023)	40.0	3D Diffuser Actor (Ke et al., 2025)	51.5	BAKU (Haldar et al., 2024)	93.0
PPI (Yang et al., 2025)	80.8	PPI (Yang et al., 2025)	56.8	MDT (Reuss et al., 2024)	76.1
Diffusion Policy (Chi et al., 2025)	42.6	BAKU (Haldar et al., 2024)	43.0	DP3 (Ze et al., 2024)	73.0
SAM2Act+ (Fang et al., 2025)	84.5	FlowPolicy (Zhang et al., 2025)	39.5	UWM (Zhu et al., 2025)	91.8
GROOT (Zhu et al., 2023a)	45.2	SPOT (Hsu et al., 2025)	47.5	MemoryVLA (Shi et al., 2026)	96.7
SPOT (Hsu et al., 2025)	82.3	Im2Flow2Act (Xu et al., 2025)	45.7	BPP (Mark et al., 2026)	90.9
OSTRA (Ours)	**89.8**	**OSTRA (Ours)**	**74.5**	**OSTRA (Ours)**	**97.4**

diffusion step, future-object queries yield an object plan P^M and predicted states $\widehat{Y}^M$. Noisy TCP queries then combine ordered map-to-scene attention with $\mathrm{sg}(P^M)$ to produce a TCP plan P^E and TCP-noise estimate. Finally, the Action Generator conditions noisy action queries on $\mathrm{sg}(P^E)$ to predict action noise and gripper openness. TCP keyframes and actions are progressively denoised from Gaussian noise, whereas future object states are re-predicted at every step. Position and rotation use separate schedules. Algorithm 1 summarizes the procedure.

Training Objective and Inference. OSTRA is trained with direct supervision at all three levels of the hierarchy. The objective combines future object-state prediction, TCP diffusion, action diffusion, and gripper supervision:

$$\begin{aligned}\mathcal{L} = \lambda_M \|\widehat{Y}^M - Y^M\|_1 + \lambda_E \|\widehat{\epsilon}_E - \epsilon_E\|_1 \\ + \lambda_A \|\widehat{\epsilon}_A - \epsilon_A\|_1 + \lambda_G \|\widehat{G} - G\|_1.\end{aligned} \tag{4}$$

The terms supervise object evolution, TCP subgoals, continuous control, and gripper state. Stop-gradient requires each stage to satisfy its own target.

At inference, map and field contexts are encoded once per replanning step. Each reverse step executes Algorithm 1. The policy executes the first predicted actions and replans from the updated history.

4 Experiments

Our experiments address five questions: **Q1:** Does OSTRA outperform baseline policies across simulation benchmarks (Section 4.1)? **Q2:** Do these benefits hold in real-world manipulation (Section 4.2)? **Q3:** How do errors and computational costs in the foundation-model components affect OSTRA? We study their impact on task success in Section 4.3 and their latency in Section 4.4. **Q4:** Which architectural designs contribute to performance? **Q5:** How sensitive is OSTRA to its key hyperparameters? The last two questions are addressed by the design ablations and hyperparameter analysis in Section 4.5.

4.1 Simulation Experiments

Benchmarks and Experimental Setup. We benchmark OSTRA on three manipulation suites, with representative task rollouts shown in Figure 4. **RLBench2**, based on RLBench (James et al., 2020), contains seven multi-stage tasks requiring long-horizon object-state reasoning. **ManiSkill** (Gu et al., 2023) contains six contact-rich insertion, placement, tool-use, and stacking tasks. **LIBERO** (Liu et al., 2023) contains four ten-task suites—Spatial, Object, Goal, and Long—for diverse language-conditioned manipulation. We use benchmark-specific datasets and protocols and report task success rate. Implementation and benchmark details are provided in Appendices A.1 and A.3.

Baselines. We compare OSTRA with benchmark-specific methods spanning action generation, history and memory, future prediction, and object-centric perception and motion (Zhao et al., 2023; Chi et al., 2025; Ze et al., 2024; Fang et al., 2025; Yang et al., 2025; Zhu et al., 2023a; Hsu et al., 2025). Detailed descriptions of all baselines are provided in Appendix A.3.

Simulation results. As summarized in Table 1, OSTRA achieves 89.8%, 74.5%, and 97.4% on RLBench2, ManiSkill, and LIBERO, improving over their strongest reported baselines by 5.3, 17.7,

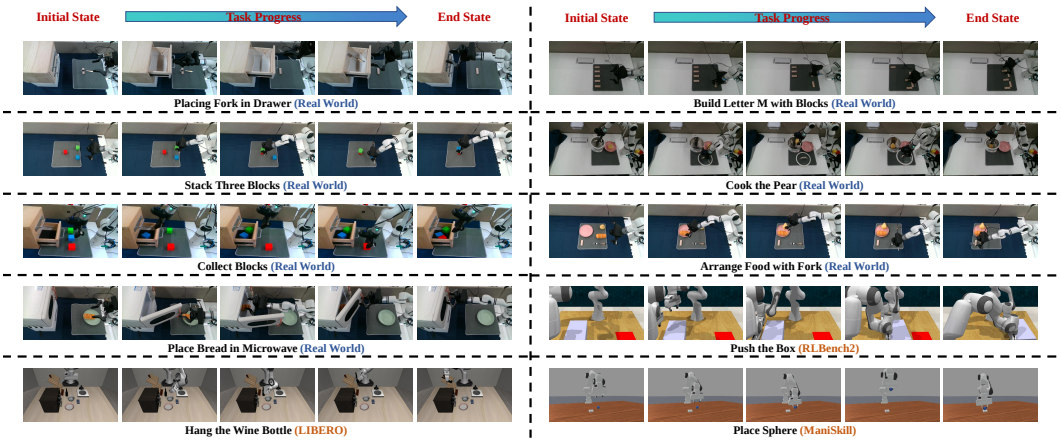


Figure 4: Task demonstrations from initial state to task completion. The seven real-world tasks are *Place Fork in Drawer*, *Stack Three Blocks*, *Collect Blocks*, *Place Bread in Microwave*, *Build Letter M with Blocks*, *Cook the Pear*, and *Arrange Food with Fork*. The figure also shows representative tasks from the three simulation benchmarks: *Push the Box* from RLBench2 (James et al., 2020), *Place Sphere* from ManiSkill (Gu et al., 2023), and *Hang the Wine Bottle* from LIBERO (Liu et al., 2023), together spanning articulated, contact-rich, and long-horizon robot manipulation challenges.

Table 2: **Real-world manipulation results.** We report full-task success rate (%) over 20 trials per task and the average across seven tasks. **Bold** and underlining denote the best and second-best results, respectively, using identical trial counts for every policy in the matched comparison.

Method	Avg. ↑	Place Fork in Drawer	Stack Three Blocks	Arrange Food with Fork	Place Bread in Microwave	Build Letter M with Blocks	Collect Blocks	Cook the Pear
ACT (Zhao et al., 2023)	19.3	15	10	35	20	0	30	25
Diffusion Policy (Chi et al., 2025)	20.0	10	5	40	20	0	35	30
DP3 (Ze et al., 2024)	40.7	40	20	65	45	0	60	55
PPI (Yang et al., 2025)	50.7	50	30	75	55	10	70	65
SAM2Act+ (Fang et al., 2025)	<u>59.3</u>	<u>55</u>	<u>40</u>	<u>85</u>	<u>65</u>	<u>15</u>	<u>80</u>	<u>75</u>
MemoryVLA (Shi et al., 2026)	54.3	50	35	80	60	10	75	70
MPI (Zeng et al., 2024)	44.3	45	25	70	50	0	65	55
SPOT (Hsu et al., 2025)	52.1	45	35	80	55	10	75	65
Im2Flow2Act (Xu et al., 2025)	42.9	40	25	70	45	5	60	55
OSTRA (Ours)	**84.3**	**85**	**80**	**95**	**85**	**65**	**90**	**90**

and 0.7 points. The gains are especially pronounced on representative multi-stage tasks. On RLBench2, OSTRA reaches 82.7% on Laptop and 77.7% on Handover, improving over the strongest baselines by 30.7 and 9.2 points, respectively. On ManiSkill, it achieves 79% on Pull Cube with Tool and 50% on Stack Pyramid, gains of 22 and 18 points. These tasks require tool-mediated interaction, articulated state changes, or sequential object arrangements, where success depends on reaching appropriate intermediate object states rather than predicting a single short-horizon motion. Complete per-task results and method comparisons across benchmarks are reported in Appendix A.4.

4.2 Real-World Experiments

We evaluate OSTRA on a 7-DoF Realman 75-BF arm with a Pika parallel gripper. An Intel RealSense D415 captures RGB-D observations, and Open3D (Zhou et al., 2018) reconstructs point clouds. The map constructor uses predicted masks, poses, and structural parameters, evaluating reasoning under errors accumulated by the OST-Map perception pipeline. Appendix A.5 provides the complete hardware, task, and evaluation protocol.

Task suite. We consider seven tasks spanning articulated interaction, long-horizon rearrangement, thin or reflective objects, and global spatial constraints. They require opening and closing containers, sequential stacking, cluttered placement, partial-occlusion reasoning, pattern construction, collection, and shared-workspace manipulation. As visualized in Figure 4, they cover articulated state, support, containment, occupancy, and multi-object layout transitions.

Table 3: System error breakdown for OSTRA. We use ground-truth replacements to identify the dominant bottleneck in OST-Map construction. Ground-truth quantities are used only for analysis and are not part of the inference interface. Δ reports the improvement over Full OSTRA.

Setting	GT Extraction	GT Object Pose	GT Structural Size	Avg. Success	Δ
Full OSTRA				89.8	0.0
+ GT Extraction	✓			90.3	+0.5
+ GT Object Pose		✓		91.9	+2.1
+ GT Structural Size			✓	93.4	+3.6
+ GT Extraction + GT Object Pose	✓	✓		92.1	+2.3
+ GT Extraction + GT Structural Size	✓		✓	94.2	+4.4
+ GT Object Pose + GT Structural Size		✓	✓	95.7	+5.9
+ GT Extraction + GT Object Pose + GT Structural Size	✓	✓	✓	96.4	+6.6

Table 4: **Inference latency (ms) and throughput.** We report mean component and end-to-end latency for the first frame and subsequent frames at batch size one. The first frame initializes OST-Map. Subsequent frames track objects and update the map for end-to-end comparison.

Stage	Mask Seg. / Track. ↓	Pose Reg. / Track. ↓	Structure & Map ↓	Encoders & Policy ↓	Total (ms) ↓	FPS ↑
First Frame (Initialization)	1412.350	1386.425	21.340	15.210	2835.325	0.353
Later Frames (Tracking)	94.832	96.155	20.125	15.180	226.292	4.419

Real-world results. Across 20 trials per task (Table 2), OSTRA averages 84.3%, exceeds SAM2Act+ (59.3%) by 25.0 points, and ranks first on all tasks. It is more reliable on drawer and microwave tasks and better preserves relations in stacking and shape construction than Diffusion Policy (Chi et al., 2025) and DP3 (Ze et al., 2024). Its largest gain is on *Build Letter M with Blocks* (65% versus 15%), supporting OST-Map reasoning under noisy, partially occluded observations.

4.3 System Error Breakdown

We use ground-truth replacements to identify which OST-Map Constructor errors most affect task success. We replace outputs of SAM 3 extraction, FoundationPose estimation and tracking, and PointNet–MLP structural estimation with ground-truth masks, poses, or sizes while fixing the policy and evaluation. We test all eight combinations: none, each single or pairwise replacement, and all GT. Ground truth is used only for diagnosis. The complete intervention protocol is provided in Appendix A.6. Table 3 reports average success relative to Full OSTRA. Single and joint replacements reveal individual and interacting errors. The Semantic Field encoder and policy remain unchanged, so all-GT residual failures are not solely due to map construction.

4.4 Latency

We profile batch-one inference on an NVIDIA GeForce RTX 4090. The first frame segments masks, registers poses, estimates structure, and initializes OST-Map. later frames track objects and update the map. Both include encoding and policy inference. Detailed profiling definitions are provided in Appendix A.6. Table 4 reports 2835.325 ms for initialization and 226.292 ms (4.419 FPS) thereafter, a $12.5\times$ speedup. Segmentation and pose estimation dominate. Map construction, encoding, and policy inference add little overhead.

4.5 Ablation Studies

We ablate OST-Map temporal roles and attributes, map–visual attention order, staged denoising and stop-gradient, and temporal and loss hyperparameters. All variants use the same RLBench2 data, capacity, budget, and evaluation seeds. Appendix A.6 provides the complete ablation and sensitivity setups used for controlled training, evaluation, and hyperparameter comparisons in matched settings.

OST-Map. Table 5 compares no map or future loss (Ex1), history only (Ex2), future supervision from the latest map (Ex3), and the full model (Ex11). Ex4–Ex11 progressively add geometry, semantics, affordances, and edge attributes to pose-only nodes under the same graph encoder and node correspondence. Full OST-Map reaches 89.8%, 20.4 points above no map. history-only and future-only reach 84.5% and 85.8%. Their combination and the Ex4–Ex11 trend show that both temporal roles and attribute groups help.

Table 5: **OST-Map ablations on RLBench2.** We vary temporal supervision and map attributes under a fixed graph encoder. Ex11 is OSTRA.

	Temporal		Node				Edge				
	Hist.	Fut.	Geo	Pose	Sem	Aff	Type	Param	Anchor	RelPose	**Mean**
Ex1	–	–	–	–	–	–	–	–	–	–	69.4
Ex2	✓	–	✓	✓	✓	✓	✓	✓	✓	✓	84.5
Ex3	–	✓	✓	✓	✓	✓	✓	✓	✓	✓	85.8
Ex4	✓	✓	–	✓	–	–	–	–	–	–	75.1
Ex5	✓	✓	✓	✓	–	–	–	–	–	–	78.8
Ex6	✓	✓	✓	✓	✓	–	–	–	–	–	82.5
Ex7	✓	✓	✓	✓	✓	✓	–	–	–	–	85.2
Ex8	✓	✓	✓	✓	✓	✓	✓	–	–	–	86.6
Ex9	✓	✓	✓	✓	✓	✓	✓	✓	–	–	87.8
Ex10	✓	✓	✓	✓	✓	✓	✓	✓	✓	–	88.9
Ex11	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**89.8**

Table 6: **OSTRA ablations on RLBench2.** We vary attention order, prediction stages, and stop-gradient boundaries. Ex12 is OSTRA.

	Attention				Denoising			Stop-Grad.		
	Map	Vis.	$M \to V$	$V \to M$	Obj.	TCP	Hier.	Obj.	TCP	**Mean**
Ex1	✓	–	–	–	✓	✓	✓	✓	✓	76.5
Ex2	–	✓	–	–	✓	✓	✓	✓	✓	78.2
Ex3	✓	✓	–	–	✓	✓	✓	✓	✓	85.4
Ex4	✓	✓	–	✓	✓	✓	✓	✓	✓	87.1
Ex5	✓	✓	✓	–	–	–	–	–	–	75.8
Ex6	✓	✓	✓	–	✓	–	✓	✓	–	83.4
Ex7	✓	✓	✓	–	–	✓	✓	–	✓	84.6
Ex8	✓	✓	✓	–	✓	✓	–	–	–	81.5
Ex9	✓	✓	✓	–	✓	✓	✓	–	–	85.9
Ex10	✓	✓	✓	–	✓	✓	✓	✓	–	87.4
Ex11	✓	✓	✓	–	✓	✓	✓	–	✓	88.3
Ex12	✓	✓	✓	–	✓	✓	✓	✓	✓	**89.8**

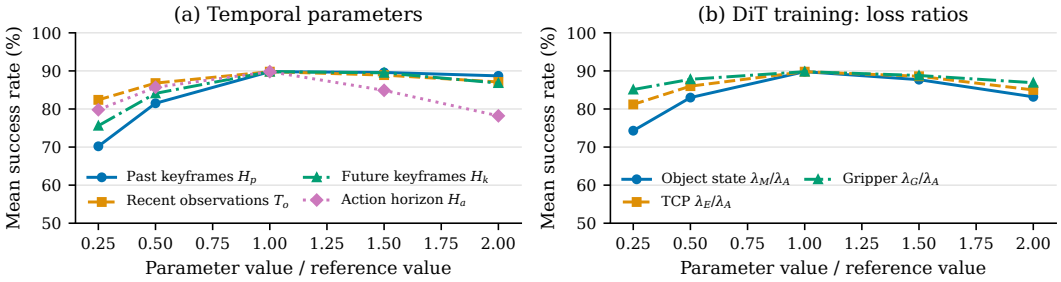


Figure 5: Hyperparameter sensitivity on RLBench2. (a) Past keyframes H_p, consecutive observations T_o, future keyframes H_k, and predicted actions H_a. (b) Object-state, TCP, and gripper loss weights relative to the action loss. Each curve varies one parameter with all others fixed at their reference values. The reference values are $H_p = T_o = H_k = 4$, $H_a = 16$, and $\lambda_M/\lambda_A = \lambda_E/\lambda_A = \lambda_G/\lambda_A = 1$. All sweeps use the same evaluation protocol.

Transition-Guided Attention. Ex1–Ex4 in Table 6 test map-only, visual-only, joint, and visual-first attention against map-first (Ex12), with the full hierarchy and stop-gradient boundaries. Map-first reaches 89.8%, exceeding these variants by 13.3, 11.6, 4.4, and 2.7 points, respectively.

Staged Denoising. With map-first attention fixed, Ex5–Ex8 test action-only, object-to-action, TCP-to-action, and flat joint prediction. Ex9–Ex12 test object/TCP stop-gradient combinations. Removed stages omit their heads and losses, while the flat variant retains all targets without hierarchical conditioning. The full hierarchy exceeds action-only by 14.0 points and flat prediction by 8.3 points. Both stop-gradient boundaries add 3.9 points over neither.

Hyperparameter sensitivity. Figure 5 varies four temporal counts—past keyframes H_p, observations T_o, future keyframes H_k, and actions H_a—and three loss ratios: λ_M/λ_A, λ_E/λ_A, and λ_G/λ_A. Each curve varies one parameter around the 89.8% full-model reference. Performance is stable nearby but falls at short or long horizons. H_a, H_p, and object-state loss are most sensitive, while gripper loss remains stable across the evaluated range and is less sensitive to its weighting.

5 Conclusion

We presented OSTRA, a framework for multi-stage manipulation that represents historical and future object states in a shared Object-State Transition Map. Transition-Guided Attention and Staged Action Generation progressively translate object-state changes into TCP subgoals and executable actions. Across RLBench2, ManiSkill, LIBERO, and seven real-world tasks, OSTRA consistently outperforms representative imitation-learning baselines, while the ablations validate the contributions of OST-Map and the proposed architectural designs. Together, these results show that explicit transition reasoning provides a structured bridge between perception and continuous control. Future work will improve map construction under perception errors and extend object-state reasoning to less structured, open-world manipulation with stronger transfer across embodiments and task domains, including deformable objects, dynamic scenes, and broader task families.

GENERATIVE AI USAGE DISCLOSURE

In this work, we used generative AI tools for no tasks with required disclosure. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in writing proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement the OSTRA method, or interpret results, and translation, dataset cleaning and reformatting, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools (Codex) to create or modify Figure 5, create or edit its plotting code, draft parts of the paper, edit the paper for readability, and adjust LaTeX formatting and layout. We have reviewed all AI-assisted work. All AI-assisted manuscript preparation was conducted under human direction and supervision; the AI-edited plotting code was executed, its output was inspected, and the manuscript was compiled and reviewed by the human authors. We take responsibility for the final content of this work, including **all** text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Suneel Belkhale and Dorsa Sadigh. Plato: Predicting latent affordances through object-centric play. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1424–1434. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/belkhale23a.html.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, RISHI HAZRA, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollar, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 138846–138923, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e0982cbc81401df3430ee1ff780dc7a2-Paper-Conference.pdf.

R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 77–85, 2017. doi: 10.1109/CVPR.2017.16.

Yuanpei Chen, Chen Wang, Li Fei-Fei, and Karen Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3809–3829. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/chen23e.html.

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. *Int. J. Rob. Res.*, 44(10–11):1684–1704, 2025. ISSN 0278-3649. doi: 10.1177/02783649241273668. URL https://doi.org/10.1177/02783649241273668.

Haoquan Fang, Markus Grotz, Wilbert Pumacay, Yi Ru Wang, Dieter Fox, Ranjay Krishna, and Jiafei Duan. SAM2Act: Integrating visual foundation model with a memory architecture for robotic manipulation. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 15925–15942. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/fang25c.html.

Yunhai Feng, Jiaming Han, Zhuoran Yang, Xiangyu Yue, Sergey Levine, and Jianlan Luo. Reflective planning: Vision-language models for multi-stage long-horizon robotic manipulation. In Joseph

Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2038–2062. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/feng25b.html.

Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: object-centric world models for robotic manipulation. *Frontiers in Neurorobotics*, Volume 19 - 2025, 2025. ISSN 1662-5218. doi: 10.3389/fnbot.2025.1585386. URL https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2025.1585386.

Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. ManiSkill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.

Abhishek Gupta, Vikash Kumar, Corey Lynch, Sergey Levine, and Karol Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1025–1037. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/gupta20a.html.

Siddhant Haldar, Zhuoran Peng, and Lerrel Pinto. Baku: An efficient transformer for multi-task policy learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 141208–141239. Curran Associates, Inc., 2024. doi: 10.52202/079017-4484. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ff887781480973bd3cb6026feb378d1e-Paper-Conference.pdf.

Cheng-Chun Hsu, Bowen Wen, Jie Xu, Yashraj Narang, Xiaolong Wang, Yuke Zhu, Joydeep Biswas, and Stan Birchfield. Spot: Se(3) pose trajectory diffusion for object-centric manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4853–4860, 2025. doi: 10.1109/ICRA55743.2025.11127562.

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/hu25g.html.

Shivam Jain, Sachit Sachdeva, and Rohan Paul. Enabling long(er) horizon imitation for manipulation tasks by modeling subgoal transitions. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 2219–2238. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/jain25b.html.

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. In *ICLR 2025 Workshops: World_Models*, 2025. URL https://mlanthology.org/iclrw/2025/jeong2025iclrw-objectcentric/.

Li Ji, Siyin Wang, Pengfang Qian, Xiaopeng Yu, Yihai Tian, Zhaoye Fei, Jingjing Gong, and Xipeng Qiu. Hime: Hierarchical embodied memory for long-horizon vision-language-action control. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Xuhui Kang and Yen-Ling Kuo. Incorporating task progress knowledge for subgoal generation in robotic manipulation through image edits. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7490–7499, 2025. doi: 10.1109/WACV61041.2025.00728.

Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. ke2024diffuseractor. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1949–1974. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/ke25a.html.

Jayjun Lee and Nima Fazeli. ViTaSCOPE: Visuo-tactile Implicit Representation for In-hand Pose and Extrinsic Contact Estimation. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.054.

Zaijing Li, Bing Hu, Rui Shao, Gongwei Chen, Dongmei Jiang, Pengwei Xie, Jianye Hao, and Liqiang Nie. Global prior meets local consistency: Dual-memory augmented vision-language-action model for efficient robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 35135–35145, June 2026.

Junchi Liang and Abdeslam Boularias. Self-supervised learning of long-horizon manipulation tasks with finite-state task machines. In Ali Jadbabaie, John Lygeros, George J. Pappas, Pablo A. Parrilo, Benjamin Recht, Claire J. Tomlin, and Melanie N. Zeilinger (eds.), *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pp. 484–497. PMLR, 07 – 08 June 2021. URL https://proceedings.mlr.press/v144/liang21a.html.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf.

Yuanzhe Liu, Jingyuan Zhu, Yuchen Mo, Gen Li, Xu Cao, Jin Jin, Yifan Shen, Zhengyuan Li, Tianjiao Yu, Wenzhen Yuan, Fangqiang Ding, and Ismini Lourentzou. Palm: Progress-aware policy learning via affordance reasoning for long-horizon robotic manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 28096–28110, June 2026.

Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura (eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1113–1132. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/lynch20a.html.

Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Silvio Savarese, and Li Fei-Fei. GTI: Learning to Generalize across Long-Horizon Tasks from Human Demonstrations. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi: 10.15607/RSS.2020.XVI.061.

Max Sobol Mark, Maria Attarian, Chuyuan Fu, Debidatta Dwibedi, Jacky Liang, Dhruv Shah, and Aviral Kumar. Long-Context Robot Imitation Learning by Focusing on Key History Frames. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.201.

Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal Diffusion Transformer: Learning Versatile Behavior from Multimodal Goals. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.121.

Hao Shi, Bin Xie, Yingfei Liu, Lin Sun, Fengrong Liu, Tiancai Wang, Erjin Zhou, Haoqiang Fan, Xiangyu Zhang, and Gao Huang. Memoryvla: Perceptual-cognitive memory in vision-language-action models for robotic manipulation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 18567–18602, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1fa3d6ccbcd15f7285fee666b2bd57be-Paper-Conference.pdf.

Kyriacos Shiarlis, Markus Wulfmeier, Sasha Salter, Shimon Whiteson, and Ingmar Posner. TACO: Learning task decomposition via temporal alignment for control. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4654–4663. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/shiarlis18a.html.

Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 785–799. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/shridhar23a.html.

Weiliang Tang, Jialin Gao, Jia-Hui Pan, Gang Wang, Li Erran Li, Yun-Hui Liu, Mingyu Ding, Pheng-Ann Heng, and Chi-Wing Fu. Rethinking intermediate representation for vlm-based robot manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29652–29662, June 2026.

Jiude Wei, Yuxuan Li, Cewu Lu, and Jianhua Sun. Physically ground commonsense knowledge for articulated object manipulation with analytic concepts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13519–13528, June 2026.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17868–17879, June 2024.

Lei Xiao, Jifeng Li, Juntao Gao, Feiyang Ye, Yan Jin, Jingjing Qian, Jing Zhang, Yong Wu, and Xiaoyuan Yu. Ava-vla: Improving vision-language-action models with active visual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13453–13463, June 2026.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2475–2499. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/xu25a.html.

Yuyin Yang, Zetao Cai, Yang Tian, Jia Zeng, and Jiangmiao Pang. Gripper Pose and Object Pointflow as Interfaces for Robotic Bimanual Manipulation. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.160.

Maryam Zare, Parham M. Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 54(12):7173–7186, 2024. doi: 10.1109/TCYB.2024.3395626.

Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.067.

Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, Heming Cui, Bin Zhao, Xuelong Li, Yu Qiao, and Hongyang Li. Learning Manipulation by Predicting Interaction. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.123.

Jiawei Zhang, Kaizhe Hu, Yingqian Huang, Yuanchen Ju, Zhengrong Xue, and Huazhe Xu. Affordgen: Generating diverse demonstrations for generalizable object manipulation with affordance correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15966–15975, June 2026a.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20743–20754, June 2026b.

Qinglun Zhang, Zhen Liu, Haoqiang Fan, Guanghui Liu, Bing Zeng, and Shuaicheng Liu. Flowpolicy: enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press, 2025. ISBN 978-1-57735-897-8. doi: 10.1609/aaai.v39i14.33617. URL https://doi.org/10.1609/aaai.v39i14.33617.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi: 10.15607/RSS.2023.XIX.016.

Pengfei Zhou, Liliang Chen, Shengcong Chen, Di Chen, Wenzhi Zhao, Rongjun Jin, Guanghui Ren, and Jianlan Luo. Act2goal: From world model to general goal-conditioned policy. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.015.

Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.015.

Yifeng Zhu, Zhenyu Jiang, Peter Stone, and Yuke Zhu. Learning generalizable manipulation policies with object-centric 3d representations. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3418–3433. PMLR, 06–09 Nov 2023a. URL https://proceedings.mlr.press/v229/zhu23b.html.

Yifeng Zhu, Abhishek Joshi, Peter Stone, and Yuke Zhu. Viola: Imitation learning for vision-based manipulation with object proposal priors. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 1199–1210. PMLR, 14–18 Dec 2023b. URL https://proceedings.mlr.press/v205/zhu23a.html.

Yifeng Zhu, Arisrei Lim, Peter Stone, and Yuke Zhu. Vision-based manipulation from single human video with open-world object graphs. *Auton. Robots*, 50(2), 2026. ISSN 0929-5593. doi: 10.1007/s10514-026-10253-8. URL https://doi.org/10.1007/s10514-026-10253-8.

A Appendix

Anonymous supplementary material. The anonymous supplementary material is available at https://osf.io/rc3hd/?view_only=d5754078e7c642459cda6d95c8d7b421 and is organized into two parts. The demo folder contains seven real-world manipulation videos, with long inactive periods removed and the remaining footage shown at three times the recorded speed. The code folder provides a review-oriented implementation snapshot of the core method, covering OST-Map representation and construction, structural and semantic scene encoding, relation-aware map reasoning, and the transition-guided prediction model. The snapshot omits datasets, model checkpoints, data-collection infrastructure, baseline implementations, and third-party source distributions.

A.1 Implementation Details

Perception and structural parameter estimation. OSTRA receives RGB-D observations and robot states. SAM 3 segments prompted task objects, and FoundationPose registers their initial 6-DoF poses and tracks them in subsequent frames. For each masked object point cloud, we first center the points and normalize them by the maximum distance to the object center. A PointNet-style structural estimator applies shared point-wise layers with widths 64, 128, and 256, each followed by

batch normalization and ReLU, and max-pools over the points to obtain a 256-dimensional object descriptor. A task-specific regressor with widths $256 \to 256 \to 128 \to D_\theta$ predicts the structural parameters, where the two hidden layers use ReLU and D_θ depends on the task template. A Softplus output transform ensures positive geometric dimensions. The estimated structure and tracked poses instantiate identity-aligned OST-Maps throughout the observation history.

Semantic Field encoder. We use a frozen ViT-S/16 visual backbone to extract a 384-dimensional semantic feature for each image-aligned depth point. Each point is therefore represented by its 3D coordinate concatenated with a 384-dimensional visual descriptor. The Semantic Field uses a two-level PointNet++ encoder. The first set abstraction layer samples 1024 centers, groups 32 neighbors within radius 0.04, and applies point-wise channels $(64, 64, 128)$. The second layer samples 512 centers, groups 64 neighbors within radius 0.08, and applies channels $(128, 128, 240)$. Batch normalization and ReLU follow each point-wise convolution. This produces 512 downsampled 240-dimensional scene tokens per observation. In parallel, a two-layer MLP $384 \to 240 \to 240$ with ReLU projects the visual descriptor of every input point, preserving a dense full-resolution Semantic Field for cross-attention.

OST-Map encoder. All OST-Map attributes are embedded to width $d = 240$. Structural and affordance point sets are encoded separately. Each point first passes through an MLP $3 \to 64 \to 64$, node-wise max pooling aggregates the point features, and a second MLP $64 \to 240 \to 240$ produces one geometry or affordance embedding per node. Layer normalization and ReLU are used in both MLPs. Node semantics are projected or embedded to 240 dimensions, while the 7D node pose is encoded by $7 \to 240 \to 240$. The geometry, affordance, semantic, and pose streams are concatenated and fused by a $960 \to 240$ projection with layer normalization and ReLU.

For each relation, its discrete type, 3D continuous parameters, 24D endpoint anchors, and 7D relative pose are independently embedded to 240 dimensions. Their concatenation is fused by the same $960 \to 240$ design. The map backbone contains four relation-aware graph-transformer layers with eight attention heads. Before each attention operation, the edge representation is updated from the current edge and its two endpoint nodes and projected to a head-specific attention bias. The node feed-forward network expands $240 \to 960 \to 240$, uses GELU, and is wrapped by pre-layer normalization and residual connections. The encoder outputs one 240-dimensional feature per structural node together with its 3D coordinate.

Transition-guided prediction backbone. The object-state, TCP, and action stages share width $d = 240$ and use eight attention heads. Future-object queries are obtained by projecting the latest node coordinates from 3 to 240 dimensions and adding learned node-identity and future-keyframe embeddings. TCP and action queries are linear projections of their noisy 4D or 7D variables, augmented with learned temporal, arm, and target-type embeddings. Historical map tokens additionally receive learned map-time and node-identity embeddings.

Each prediction stage follows the same map-first attention pattern. Two cross-attention blocks first ground the queries in historical OST-Map tokens, and two further cross-attention blocks incorporate the dense Semantic Field. Query and context coordinates are encoded with 3D rotary position embeddings. Each cross-attention block consists of eight-head attention, a residual connection and layer normalization, followed by a $240 \to 240 \to 240$ feed-forward network with ReLU and a second residual connection. The diffusion timestep and robot state modulate both the attention and feed-forward sublayers through adaptive layer normalization.

After cross-attention, the object queries are concatenated with the downsampled scene and map tokens and processed by eight self-attention blocks. The resulting object plan conditions the TCP stage. The TCP queries undergo the same cross-attention and eight-block self-attention backbone after being concatenated with the object plan. Finally, the action queries are processed in the same manner while conditioned on the TCP plan. Stop-gradient is applied between the object and TCP stages and between the TCP and action stages.

Conditioning and prediction heads. The diffusion timestep is encoded by a 240-dimensional sinusoidal embedding followed by $240 \to 240 \to 240$ layers with SiLU. Robot state is encoded by an MLP $D_s \to 240 \to 240$ with ReLU and averaged across the observation window. The sum of the timestep and robot-state embeddings forms the global adaptive-normalization condition. A final

modulation layer $240 \rightarrow 480$ predicts scale and shift parameters before the output heads. Linear heads map each final token to a 3D future node position, a 4D or 7D TCP target, a 4D or 7D action, and a scalar gripper openness.

Policy inputs, training, and inference. Each policy input contains T_o consecutive observations and H_p historical OST-Map keyframes. The policy predicts H_k future object-state keyframes, TCP subgoals, and an action chunk of length H_a. Unless varied in the sensitivity study, we use $H_p = T_o = H_k = 4$ and $H_a = 16$. Training jointly supervises future structural-node positions, TCP diffusion, action diffusion, and gripper state. The reference loss ratios are $\lambda_M/\lambda_A = \lambda_E/\lambda_A = \lambda_G/\lambda_A = 1$. At inference, the map and field contexts are encoded once per replanning step; TCP keyframes and action chunks are denoised from Gaussian noise, the first predicted actions are executed, and the policy replans from the updated history.

A.2 OST-Map Construction Accuracy

We directly evaluate the accuracy of the OST-Map Constructor independently of the downstream policy on the simulation benchmarks. Ground-truth object masks, 6-DoF poses, geometric parameters, and persistent object identities are read from the corresponding simulator states and task definitions. These quantities are used only for evaluation and are not provided to the constructor. Object extraction is measured by mask mean intersection over union (mIoU). Pose accuracy is reported as the mean translation and rotation error of matched task objects. Structural accuracy is measured by the normalized mean absolute error (NMAE) of the predicted primitive dimensions, where each error is normalized by the corresponding simulator-provided dimension. Temporal identity accuracy measures the fraction of tracked object and structural-node identities that remain consistent over the full observation window.

We additionally report complete-map accuracy, a strict frame-level metric. An OST-Map is counted as correct only when all task-relevant objects are detected, their identities are correct, and every structural node satisfies the translation, rotation, and structural-error thresholds. This metric therefore captures whether the complete structured state supplied to OSTRA is reliable, rather than averaging independently correct map attributes. For complete-map accuracy, we use mask IoU threshold $\tau_{\text{mask}} = 0.75$, translation threshold $\tau_p = 2.0$ cm, rotation threshold $\tau_R = 10^\circ$, and structural NMAE threshold $\tau_\theta = 10\%$. We do not report these constructor metrics for the real-world experiments, where the required dense masks, exact object poses, and geometric parameters are not available as independent ground truth.

Analysis. The constructor recovers accurate and temporally stable OST-Maps across all three simulation benchmarks. It achieves an average mask mIoU of 89.2%, while the mean pose errors remain below one centimeter in translation and 4° in rotation. The 5.2% structural NMAE further shows that the estimated primitive dimensions closely match the simulator geometry. These accuracies are important because OST-Map represents object state through both pose and fine-grained structure; small errors in either component would directly perturb the transition targets and spatial relations supplied to the policy.

Identity accuracy reaches 95.4%, indicating that the same objects and structural parts are consistently aligned across the observation history. Under the stricter complete-map criterion, which requires every task-relevant element in a frame to satisfy all extraction, pose, structure, and identity conditions simultaneously, the constructor remains correct in 86.2% of cases. The relatively small gap between the individual metrics and this all-or-nothing measure indicates that constructor errors are not dominated by frequent catastrophic map failures. Overall, the results show that OSTRA receives a reliable structured representation on which to model object-state transitions.

A.3 Benchmarks and Baseline Details

Simulation benchmarks. RLBench2 is based on RLBench (James et al., 2020) and contains seven multi-stage tasks requiring long-horizon object-state reasoning. **RLBench2 protocol.** We follow PPI (Yang et al., 2025), using its regenerated RLBench2 data for Box, Ball, Drawer, Laptop, Dustpan, Tray, and Handover Easy. Each task uses 100 demonstrations of approximately 150–250 steps.

Table 7: **Accuracy of the OST-Map Constructor.** Translation and rotation errors are measured on matched task objects. Structural NMAE averages over all predicted primitive dimensions. Identity accuracy evaluates correspondence over the complete observation window, and complete-map accuracy requires every task-relevant map element to satisfy the evaluation thresholds.

Benchmark	Mask mIoU (%) ↑	Translation (cm) ↓	Rotation (°) ↓	Structural NMAE (%) ↓	Identity accuracy (%) ↑	Complete-map accuracy (%) ↑
RLBench2	90.4	0.85	3.1	4.8	96.2	88.5
ManiSkill	88.2	1.08	4.2	5.6	94.7	84.1
LIBERO	89.1	0.94	3.7	5.1	95.3	86.0
Simulation average	**89.2**	**0.96**	**3.7**	**5.2**	**95.4**	**86.2**

Evaluation uses 100 rollouts per task with varying initial states. We report the mean and standard deviation over three checkpoints and average success across the seven tasks.

ManiSkill protocol. ManiSkill (Gu et al., 2023) contains the six contact-rich insertion, placement, tool-use, and stacking tasks used in our evaluation. Following the official imitation-learning setup, we train with 100 successful motion-planning demonstrations per task. We retain each environment's official randomized initialization, episode horizon, and success condition, evaluate 100 episodes per task, and report task and average success rates.

LIBERO protocol. LIBERO (Liu et al., 2023) contains the Spatial, Object, Goal, and Long suites used in our evaluation. We follow MemoryVLA (Shi et al., 2026): each task uses 50 demonstrations after failed trajectories are removed, and each task is evaluated for 50 trials from the official initial states. We use 10 initial no-op steps for object stabilization and maximum episode lengths of 220, 280, 300, and 520 steps for Spatial, Object, Goal, and Long, respectively. We use no action ensembling and report success rates at the best validation checkpoint.

Table 8: **Quantitative results on RLBench2.** We report task success rate (%) as mean ± standard deviation over three checkpoints, with 100 evaluation rollouts per task for each checkpoint. **Bold** and underlining denote the best and second-best results, respectively, across all seven tasks.

Method	Avg. ↑	Box	Ball	Drawer	Laptop	Dustpan	Tray	Handover
ACT (Zhao et al., 2023)	15.5	67.0±7.0	38.3±10.0	1.7±2.1	0.0±0.0	0.0±0.0	1.3±1.5	0.0±0.0
DP3 (Ze et al., 2024)	26.0	39.3±3.1	27.0±6.6	0.0±0.0	6.0±2.6	**98.7**±0.6	6.3±0.6	4.7±1.5
3D Diffuser Actor (Ke et al., 2025)	64.7	54.7±3.8	87.3±1.9	52.7±14.1	40.7±6.1	96.7±2.6	76.0±4.3	44.7±3.3
PerAct^2 (Shridhar et al., 2023)	40.0	62.0±26.2	50.0±8.7	49.7±16.6	36.7±5.7	2.0±3.5	60.0±6.2	19.7±6.0
PPI (Yang et al., 2025)	80.8	96.7±1.5	89.3±1.5	79.7±3.8	46.3±1.2	**98.7**±1.5	**92.0**±1.0	62.7±2.5
DP (Chi et al., 2025)	42.6	61.0±2.0	59.7±3.5	38.5±4.0	10.3±3.1	69.7±2.5	67.0±3.2	2.0±1.0
SAM2Act+ (Fang et al., 2025)	<u>84.5</u>	<u>95.2</u>±1.2	88.0±2.0	<u>81.5</u>±3.0	**52.0**±4.5	97.0±1.1	<u>89.5</u>±1.8	<u>68.5</u>±3.2
GROOT (Zhu et al., 2023a)	45.2	65.0±5.1	58.3±4.2	35.0±6.0	22.1±3.8	45.0±5.0	50.0±4.0	41.2±3.5
SPOT (Hsu et al., 2025)	82.3	94.0±1.8	<u>86.5</u>±2.2	78.0±4.1	<u>48.5</u>±3.9	96.5±1.5	88.0±2.1	64.8±3.0
OSTRA (Ours)	**89.8**	**98.3**±0.6	**93.0**±1.5	**87.0**±7.5	**82.7**±9.6	<u>97.7</u>±1.0	**92.0**±0.6	**77.7**±3.8

Baseline details. We compare OSTRA with benchmark-specific baselines from four categories.

Action-generation policies. ACT (Zhao et al., 2023) uses a conditional variational transformer to predict temporally coherent action chunks. Diffusion Policy (DP) (Chi et al., 2025) generates continuous action sequences through conditional denoising. DP3 (Ze et al., 2024) conditions action diffusion on compact 3D point-cloud representations. 3D Diffuser Actor (Ke et al., 2025) performs policy diffusion over actions using language-conditioned 3D scene features. PerAct^2 (Shridhar et al., 2023) predicts discretized 6-DoF actions from language-conditioned voxel observations with a Perceiver-based policy. BAKU (Haldar et al., 2024) is an efficient transformer policy designed for multi-task imitation learning. FlowPolicy (Zhang et al., 2025) learns a fast 3D visuomotor policy through consistency flow matching. MDT (Reuss et al., 2024) is a diffusion transformer conditioned on multiple forms of task-goal input.

History and memory policies. SAM2Act+ (Fang et al., 2025) augments visual foundation-model features with a memory architecture for manipulation. MemoryVLA (Shi et al., 2026) equips a vision-language-action policy with perceptual and cognitive memory. BPP (Mark et al., 2026) handles long observation histories by selecting and attending to key past frames.

Future-prediction methods. PPI (Yang et al., 2025) represents manipulation through predicted gripper poses and object point flows. UWM (Zhu et al., 2025) jointly models future video and actions with coupled diffusion objectives. MPI (Zeng et al., 2024) pretrains manipulation representations by predicting interactions and therefore requires a separately specified downstream action head.

Object-centric perception and motion methods. GROOT (Zhu et al., 2023a) learns generalizable manipulation policies from object-centric 3D representations. SPOT (Hsu et al., 2025) generates object-centric $\mathrm{SE}(3)$ pose trajectories with diffusion. Im2Flow2Act (Xu et al., 2025) uses predicted object flow as a cross-domain interface from videos to robot actions.

A.4 Detailed Simulation Benchmark Scores

This section reports the full per-task results summarized in Table 1.

Table 9: **Quantitative results on ManiSkill.** We report success (%) over 100 episodes per task and the six-task mean. A dagger marks an unmatched published score, "–" an unavailable result, and **bold** the best result under the official ManiSkill protocol.

Method	Avg. Success ↑	Peg Insertion Side	Place Sphere	Plug Charger	Pull Cube with Tool	Stack Cube	Stack Pyramid
ACT (Zhao et al., 2023)	21.3	0	2	28	38	49	11
Diffusion Policy (Chi et al., 2025)	26.8	0	4	36	47	61	13
DP3 (Ze et al., 2024)	53.8	55	62	58	48	72	28
3D Diffuser Actor (Ke et al., 2025)	51.5	48	60	55	46	71	29
PPI (Yang et al., 2025)	56.8	57	65	60	52	75	32
BAKU (Haldar et al., 2024)	43.0	25	44	48	55	66	20
FlowPolicy (Zhang et al., 2025)	39.5	14	32	45	53	68	25
SPOT (Hsu et al., 2025)	47.5	33	51	51	57	71	22
Im2Flow2Act (Xu et al., 2025)	45.7	29	47	52	56	69	21
OSTRA (Ours)	**74.5**	**70**	**84**	**75**	**79**	**89**	**50**

Table 10: **Quantitative results on LIBERO.** We report task success rate (%), with 50 evaluation trials per task. **Bold** and underlining denote the best and second-best results, respectively.

Method	LIBERO-Spatial	LIBERO-Object	LIBERO-Goal	LIBERO-Long	Avg. Success ↑
ACT (Zhao et al., 2023)	82.0	78.8	66.1	44.0	67.7
GROOT (Zhu et al., 2023a)	79.1	81.5	63.2	41.0	66.2
Diffusion Policy (Chi et al., 2025)	78.3	92.5	68.3	50.5	72.4
BAKU (Haldar et al., 2024)	94.0	95.0	96.0	87.0	93.0
MDT (Reuss et al., 2024)	78.5	87.5	73.5	64.8	76.1
DP3 (Ze et al., 2024)	86.5	84.0	71.5	50.0	73.0
UWM (Zhu et al., 2025)	93.5	93.8	94.2	85.5	91.8
MemoryVLA (Shi et al., 2026)	<u>98.4</u>	<u>98.4</u>	<u>96.4</u>	<u>93.4</u>	<u>96.7</u>
BPP (Mark et al., 2026)	92.0	94.5	93.0	84.0	90.9
OSTRA (Ours)	**98.5**	**98.7**	**97.3**	**95.0**	**97.4**

A.5 Real-World Details

Hardware and perception. The real-world platform consists of a 7-DoF Realman 75-BF arm, a Pika parallel gripper, and an Intel RealSense D415 RGB-D camera. Open3D reconstructs point clouds from the RGB-D stream. All reported evaluations use predicted masks, poses, and structural parameters rather than ground-truth map inputs.

Tasks and protocol. We evaluate seven tasks: *Place Fork in Drawer*, *Stack Three Blocks*, *Arrange Food with Fork*, *Place Bread in Microwave*, *Build Letter M with Blocks*, *Collect Blocks*, and *Cook the Pear*. Together they cover articulated interaction, sequential stacking, cluttered placement, partial-occlusion reasoning, pattern construction, collection, and shared-workspace manipulation. We run 20 trials per method and task and report full-task success rate.

A.6 Detailed Setups for Ablations and Diagnostic Experiments

System error breakdown. We diagnose the OST-Map Constructor by replacing the outputs of SAM 3 extraction, FoundationPose pose estimation and tracking, and PointNet–MLP structural estimation with ground-truth masks, poses, or sizes. We evaluate all eight combinations: no replacement, each of the three individual replacements, all three pairwise replacements, and replacement of all three components. The policy and Semantic Field encoder remain fixed, and ground-truth quantities are used only for this diagnostic experiment.

System error analysis. Table 3 shows that structural estimation is the largest individual source of recoverable error. Replacing structural sizes increases success from 89.8% to 93.4%, a gain of 3.6 points, compared with 2.1 points from ground-truth object poses and 0.5 points from ground-truth extraction. This ordering indicates that coarse object discovery is already relatively reliable, whereas errors in the geometric parameters used to instantiate OST-Map nodes more directly affect downstream state reasoning. The pairwise results reinforce this interpretation. Correcting pose and structural size together reaches 95.7%, which recovers 5.9 points and outperforms either correction alone. Adding ground-truth extraction to all other corrections further raises success to 96.4%. The remaining 3.6% failure rate under all ground-truth replacements cannot be attributed to these three constructor outputs. It instead reflects residual difficulty in the Semantic Field, policy prediction, control execution, or other unmodeled sources of variation.

Latency profiling. We profile batch-one inference on an NVIDIA GeForce RTX 4090. First-frame latency includes object segmentation, pose registration, structural estimation, OST-Map initialization, encoding, and policy inference. Later-frame latency replaces segmentation and registration with tracking and includes map updates, encoding, and policy inference. Throughput is computed from sequential end-to-end latency and does not denote the robot control frequency.

Latency analysis. Table 4 attributes nearly all initialization cost to the two foundation-model perception components. Segmentation and pose registration together require 2798.775 ms, or 98.7% of the 2835.325 ms first-frame latency. In contrast, structure and map construction require 21.340 ms, and the encoders and policy require 15.210 ms. Temporal tracking reduces mask processing from 1412.350 ms to 94.832 ms and pose processing from 1386.425 ms to 96.155 ms. Meanwhile, structure and map processing changes by only 1.215 ms, and encoder and policy latency is effectively unchanged. The resulting 226.292 ms later-frame latency is $12.5\times$ lower than initialization latency. Perception still accounts for 84.4% of later-frame latency, so faster segmentation and pose tracking offer substantially more headroom than simplifying OST-Map encoding or policy inference.

Ablations. All ablations use the same RLBench2 data, model capacity, training and evaluation budgets, and evaluation seeds. The OST-Map study varies historical and future transition supervision and progressively adds node geometry, semantics, affordances, and edge attributes under a fixed graph encoder and node correspondence. The architecture study varies map/visual attention, prediction stages, and object/TCP stop-gradient boundaries. Removed stages omit their prediction heads and losses; the flat variant retains all targets without hierarchical conditioning.

Historical and future OST-Maps. The temporal variants in Table 5 isolate the two roles of OST-Map. Removing the map and future-state loss yields 69.4%. Using the complete map attributes only for history raises success to 84.5%, while using them only for future supervision reaches 85.8%. Thus, observed state history and predicted future states independently recover 15.1 and 16.4 points over the no-map variant. Combining both roles reaches 89.8%, adding 5.3 points over history alone and 4.0 points over future supervision alone. The gain from their combination supports using one identity-aligned representation on both sides of the current observation rather than treating history encoding and future prediction as separate auxiliary mechanisms.

OST-Map attributes. Ex4–Ex11 form a cumulative attribute study with temporal supervision and the graph encoder held fixed. Pose-only nodes obtain 75.1%. Adding node geometry raises success to 78.8%, and adding semantics then raises it to 82.5%. Affordances provide a further 2.7-point gain to 85.2%. Collectively, these node attributes contribute 10.1 points beyond pose alone, showing that object state cannot be represented adequately by rigid transforms without information

about shape, identity, and interaction regions. Edge attributes add another 4.6 points in total. Relation type, parameters, anchors, and relative pose progressively raise success from 85.2% to 86.6%, 87.8%, 88.9%, and 89.8%. Because the attributes are added cumulatively, these increments measure marginal gains in this ordering rather than isolated importance. Nevertheless, the monotonic trend shows that each relation descriptor provides complementary information.

Map–visual attention order. The first four rows of Table 6 retain the full prediction hierarchy and stop-gradient boundaries while changing how map and visual tokens interact. Map-only and visual-only attention reach 76.5% and 78.2%, respectively. Joint access to both modalities improves success to 85.4%, an 8.9-point gain over map-only attention and a 7.2-point gain over visual-only attention. Ordering the interaction also matters. Visual-first attention reaches 87.1%, whereas the map-first design used by OSTRA reaches 89.8%. The 2.7-point advantage of map-first attention is consistent with first establishing a transition-conditioned object representation and then using dense visual tokens to recover local scene detail.

Staged prediction and gradient boundaries. With map-first attention fixed, action-only prediction reaches 75.8%. Introducing an object-state stage raises success to 83.4%, while introducing a TCP stage reaches 84.6%. A flat model that predicts object state, TCP, and actions without hierarchical conditioning reaches only 81.5%. The complete object-to-TCP-to-action hierarchy achieves 89.8%, improving over action-only prediction by 14.0 points and over flat joint prediction by 8.3 points. These comparisons show that the auxiliary targets are most useful when their predictions become ordered conditioning variables rather than parallel output heads. The stop-gradient variants further separate supervision between stages. Without either boundary, success is 85.9%. Applying only the object boundary raises it to 87.4%, applying only the TCP boundary raises it to 88.3%, and applying both reaches 89.8%. The two individual gains sum to the 3.9-point improvement of the complete design, indicating complementary protection against later-stage action losses overriding intermediate object-state and subgoal supervision.

Hyperparameter sensitivity. We vary one parameter at a time around the full-model reference while fixing all others. The temporal study varies H_p, T_o, H_k, and H_a; the loss study varies λ_M/λ_A, λ_E/λ_A, and λ_G/λ_A. The reference configuration is $H_p = T_o = H_k = 4$, $H_a = 16$, and $\lambda_M/\lambda_A = \lambda_E/\lambda_A = \lambda_G/\lambda_A = 1$.

Hyperparameter analysis. Figure 5 shows a consistent maximum at the reference configuration, with different sensitivities across parameter groups. Reducing the number of past keyframes to one quarter of the reference produces the largest temporal drop, from 89.8% to 70.2%. Reducing future keyframes to the same ratio reaches 75.6%, while reducing recent observations reaches 82.4%. This gap suggests that sparse keyframe history contains information that cannot be replaced by a similarly short window of consecutive observations. Action horizon is asymmetric. A quarter-length horizon reaches 79.8%, and doubling the horizon reaches 78.2%, whereas doubling the other temporal contexts retains at least 86.8%. The policy therefore benefits from enough action context to represent a coherent subgoal, but long open-loop chunks make control substantially harder.

The loss-weight sweeps show the same ordering as the prediction hierarchy. At one quarter of the reference weight, object-state, TCP, and gripper supervision reach 74.3%, 81.2%, and 85.1%, respectively. Object-state supervision is therefore the most sensitive because errors at the first prediction stage propagate to both TCP and action generation. TCP weighting is moderately sensitive, while gripper weighting remains comparatively stable. Moderate over-weighting is less damaging than severe under-weighting. At $1.5\times$ the reference, the three loss sweeps remain between 87.7% and 88.8%, while at $0.25\times$ they lose between 4.7 and 15.5 points. This pattern supports balanced supervision at the reference setting without requiring precise tuning of the lower-level gripper loss.

B OST-Map Representation Details

This section provides a detailed description of the OST-Map introduced in Section 3.2. The representation is designed around three requirements. First, it should separate task-relevant objects from irrelevant scene detail. Second, it should expose the internal geometry and interaction structure that distinguish object states beyond a single rigid pose. Third, the same structural entities should re-

main identifiable across time so that observed and future object-state transitions can be expressed in a shared space.

B.1 Hierarchical Scene Representation

OST-Map represents a scene through the hierarchy *scene* $\rightarrow$ *object* $\rightarrow$ *structural node*. At time t, the map is written as

$$\mathcal{M}_t = \big(\mathcal{O}, \mathcal{V}_t, \mathcal{E}_t, \Pi\big), \tag{5}$$

where $\mathcal{O}$ is the set of task-relevant objects, $\mathcal{V}_t$ and $\mathcal{E}_t$ are the flattened structural nodes and edges, and Π assigns every node and relation to its parent object. This hierarchy preserves object membership while presenting the complete scene as one relational graph.

An object may contain one or several structural nodes. The decomposition is chosen to expose the geometry that matters for manipulation rather than to reconstruct every visual detail. A simple block is represented by one cuboid node. In contrast, a charger is decomposed into a body and two prongs, while its receptacle is decomposed into a center divider and a rectangular face loop. The latter decomposition makes the relative placement of the prongs, slots, and insertion opening explicit even though the charger and receptacle are each perceived as a single object.

OST-Map separates attributes according to how they vary over time. Object identity, node semantics, primitive geometry, and the graph schema are fixed within an episode. Node poses and pairwise relative poses vary as the scene evolves. This factorization provides stable structural correspondence without assuming that the objects themselves remain stationary.

B.2 Structural Nodes

Each structural node describes one task-relevant geometric part:

$$v_{t,i} = \big(o_i, c_i, \tau_i, \theta_i, \mathcal{P}_i, \mathcal{A}_i, \mathcal{F}_i, \mathcal{D}_i, p_{t,i}, q_{t,i}\big). \tag{6}$$

Here, o_i is the persistent parent-object identity and c_i is the semantic identity of the structural part. The primitive type τ_i specifies a geometric family, such as a cuboid, cylinder, sphere, ring, or rectangular ring. Its parameters θ_i describe the corresponding dimensions, such as side lengths, radius, height, or inner and outer extent.

The structural parameters generate several complementary geometric descriptions. $\mathcal{P}_i$ is a set of points sampled from the primitive surface and provides shape evidence without requiring a dense object mesh. $\mathcal{A}_i$ is an optional set of affordance points identifying regions that are particularly relevant to interaction, such as grasping, contact, or insertion regions. $\mathcal{F}_i$ and $\mathcal{D}_i$ respectively denote candidate faces and axes. They provide local geometric references for relations between structural parts.

The time-dependent state of a node is its pose $(p_{t,i}, q_{t,i}) \in \mathbb{R}^3 \times \mathbb{S}^3$, where $p_{t,i}$ is the node position and $q_{t,i}$ is a unit quaternion. Thus, two observations of the same node share o_i, c_i, τ_i, and θ_i, but may differ in position and orientation. Structural parameters are not treated as an additional unstructured state vector; their effect is expressed through the resulting geometry, affordances, and anchors.

B.3 Relation Edges

An edge relates two structural nodes:

$$e_{t,ij} = \big(i, j, r_{ij}, \eta_{ij}, a^i_{ij}, a^j_{ij}, \Delta p_{t,ij}, \Delta q_{t,ij}\big). \tag{7}$$

The relation type r_{ij} describes how the endpoint parts may be connected. OST-Map supports seven relation types:

- *Fixed* relations preserve a rigid relative transformation.
- *Revolute* relations allow rotation around a shared axis.
- *Prismatic* relations allow translation along a designated direction.
- *Cylindrical* relations allow both translation and rotation along a shared axis.

- *Planar-Contact* relations preserve contact between two surfaces while allowing motion within the plane.
- *Alignment* relations constrain directions without fixing relative position.
- *Free* relations impose no task-specific constraint.

The vector $\eta_{ij} \in \mathbb{R}^3$ stores relation-dependent continuous parameters. The references a_{ij}^i and a_{ij}^j select a face or axis from each endpoint. A face anchor is described by its center, normal, tangent, and bitangent; an axis anchor is described by a point and its direction, with unused components set to zero. Finally, $(\Delta p_{t,ij}, \Delta q_{t,ij})$ records the current relative translation and quaternion orientation.

Known structural relations can be specified by the task template. Every other node pair is connected by a *Free* relation. A free relation has no constraint parameters or anchors, but it still carries the measured relative pose. Consequently, the graph represents both explicit geometric structure and unconstrained spatial context through the same edge form.

B.4 Task Templates

Each task defines a structural template specifying the task-relevant objects, their primitive decomposition, the ordering of structural nodes, and the episode-level geometric parameters. The same template is instantiated at every time step. Perception estimates the object poses and structural parameters, while tracking updates the poses without changing node identity or ordering.

Table 11 gives representative templates from the tasks considered in this work. In *Stack Cube*, each cube is adequately described by one cuboid node. In *Plug Charger*, two perceived objects expand into five structural nodes. The charger contains a body and two prongs, whose dimensions and separation determine the insertion geometry. The receptacle contains a center divider and a rectangular face loop, exposing the two slots and their boundary. In RLBench2 *Push Box*, one cuboid node captures the manipulated object because no finer decomposition is required.

B.5 Attribute Encoding

Table 12 summarizes the numerical attributes used to encode one OST-Map. Node geometry and affordances are represented as variable-size point sets, allowing simple and complex primitives to use different sampling densities. Each point remains associated with its structural node. Node semantics may be represented by learned identifiers or pretrained semantic embeddings. Relations use a shared fixed-width description regardless of relation type.

For each node, the structural and affordance point sets are aggregated and combined with semantic and pose attributes. For each relation, relation type, continuous parameters, anchors, and relative pose provide complementary descriptions of how the endpoint nodes are arranged. Relational propagation then produces one coordinate-aware token per structural node. Because the tokens retain both their 3D positions and persistent node identities, they can be compared across time and used directly by Transition-Guided Attention.

B.6 Temporal Alignment and Transition Targets

OST-Map turns the per-frame structure graph into a transition representation by preserving the template, object ordering, and node ordering across time. For an observation window, OSTRA therefore receives

$$\mathcal{H}_t^M = [\mathcal{M}_{t-T_o+1}, \ldots, \mathcal{M}_t], \tag{8}$$

where the i-th node always refers to the same structural part. Tracking updates $p_{t,i}$ and $q_{t,i}$ while leaving the node identity and graph schema unchanged. Differences across aligned node poses directly expose object-state evolution, including rigid motion, articulated displacement, part alignment, and changes in inter-object configuration.

Future transition supervision uses the same node ordering. For the reported OSTRA model, each selected future keyframe provides the target position of every structural node:

$$Y_t^M = \left\{ p_{t,h,i}^\star \right\}_{h=1,i=1}^{H_k,N}. \tag{9}$$

Table 11: **Representative task-specific OST-Map templates.** The listed parameters describe episode-level geometry, while node poses vary over time.

Task	Objects	Nodes	Parameters	Primitive decomposition
Stack Cube	2	2	6 size	red cuboid, green cuboid
Plug Charger	2	5	14 size + 1 prong spacing	body, two prongs, divider, ring
RLBench2 Push Box	1	1	3 size	box cuboid

Table 12: **Numerical attributes of an OST-Map.** N, M, P, and K denote the numbers of nodes, relations, structural points, and affordance points in one map.

Attribute	Shape	Interpretation
Node position	$N \times 3$	Current structural-node centers
Node orientation	$N \times 4$	Unit quaternions
Node semantics	N or $N \times D_s$	Semantic identities or embeddings
Structural points	$P \times 3$	Primitive surface samples assigned to nodes
Affordance points	$K \times 3$	Interaction-region samples assigned to nodes
Relation connectivity	$2 \times M$	Endpoint node indices
Relation type	$M \times 1$	Discrete relation identities
Relation parameters	$M \times 3$	Continuous geometric parameters
Relation anchors	$M \times 24$	Two 12D endpoint frames
Relative pose	$M \times 7$	Relative position and quaternion

The prediction target is therefore expressed in the same structured space as the observed history rather than as an unrelated latent vector. Orientation remains part of each observed OST-Map, while future node positions provide the transition targets used in the reported model. This shared correspondence allows the predicted object-state sequence to condition TCP-subgoal generation and, subsequently, continuous action generation.