MapPolicy: Structure-Aware Imitation Learning for Robot Manipulation via Physically Constrained Scene Map

Zehao Du[1]
duzehao_2023jd@sjtu.edu.cn

Rui Li[*1]
lirui_wesley@sjtu.edu.cn

Hongyu Liu[*1]
430887267@sjtu.edu.cn

Zheng Huang[1]
zheng6.6@sjtu.edu.cn

Jiude Wei[1]
wjd_kznwl@sjtu.edu.cn

Jiachun Bao[1 2]
jiachunbao@sjtu.edu.cn

Cewu Lu[1 2]
lucewu@sjtu.edu.cn

Jianhua Sun[†1]
gothic@sjtu.edu.cn

Abstract

Robot manipulation under partial observability requires policies to act from incomplete visual evidence while still reasoning about object structure, spatial layout, and physical relations. Existing imitation learning methods mainly encode currently visible observations, which makes them brittle when task-critical parts, contact regions, or support relations are occluded. We introduce Scene Map, a structured scene representation that models object shape, part structure, spatial layout, and physical constraints as a graph of manipulation-relevant primitives and relations. Scene Map turns occluded but task-relevant structure into explicit prediction targets and provides structural priors for inferring unseen components from partial observations. Building on Scene Map, we propose MapPolicy, an imitation learning framework that fuses structured map features with visual and robot state features for action prediction. MapPolicy uses structure-modulated graph attention to propagate global structural and spatial information through primitive features, and a physical constraint loss to regularize learned features with geometric and kinematic consistency. Across 49 manipulation tasks in three simulation benchmarks and five real world tasks, MapPolicy consistently improves over strong imitation learning baselines on partial observability, articulated interaction, and structure-sensitive manipulation. Project page: https://zehao-du.github.io/MapPolicy-Homepage/.

1 Introduction

Partial observability is a core challenge for robot manipulation [73, 76]. Visual observations are partial projections of a 3D scene from limited viewpoints, so task-relevant information can be missing from the current input [26, 2, 69]. When the missing regions contain object parts, support relations, contact surfaces, or actionable regions, the problem is not simply perceptual noise. It becomes structural incompleteness, where the policy must act using scene structure that is only partially observed.

[1]Shanghai Jiao Tong University, Shanghai 200240, China.
[2]Shanghai Innovation Institute, Shanghai 200231, China.
[*]Rui Li and Hongyu Liu contributed equally as second authors.
[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

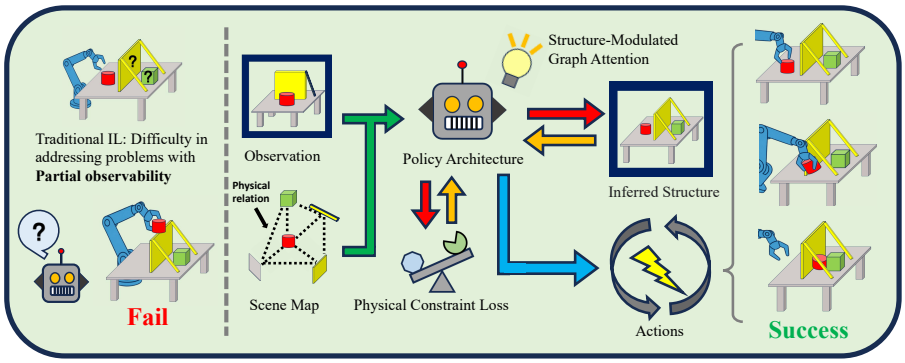

Figure 1: Overview of MapPolicy. Traditional imitation learning can fail under partial observability because the current observation omits task-relevant structure. MapPolicy augments the observation with a Scene Map that captures physical relations, uses structure-modulated graph attention to infer hidden structure, and regularizes learned features with physical constraint loss for action prediction.

Existing visuomotor imitation learning methods improve scene understanding with richer representations, including object-centric representations that focus on task-relevant objects [78, 77, 41, 51, 42] and structured representations such as keypoints, semantic fields, and geometric abstractions [10, 66, 11, 56, 45]. However, these representations are still largely derived from visible observations. They can make observed cues easier to use, but they do not explicitly model how visible evidence constrains unseen object structure. As illustrated in fig. 1, this limitation can make traditional imitation learning ambiguous and brittle when successful manipulation depends on hidden supports, physical relations, or spatial arrangements.

Humans are often more robust in similar situations because they use prior knowledge of object structure and physical relations to infer what is not visible. For example, when placing an object onto a shelf, a person does not need to see every layer of the shelf to infer the existence and approximate position of supporting surfaces. This observation suggests that robust manipulation under partial observability requires more than stronger encoding of visible pixels or point clouds. It requires a representation that exposes object structure and physical relations as explicit priors, consistent with the view that affordance and action are grounded in environmental structure [18].

To this end, we propose Scene Map, a structured scene representation for policy learning. Scene Map represents a manipulation scene as a graph of geometric primitives and physical relations. Nodes describe primitive-level object parts such as cuboids and cylinders through their shape, scale, 3D position, and rotation, while edges encode connectivity and physical constraints between primitives. This design follows work on hierarchical graphs, part decompositions, and compositional geometric primitives [35, 63, 79, 38, 39, 57]. By organizing object shape, part structure, spatial layout, and constraints in a graph, Scene Map provides structural cues that are not reliably available from the current observation alone.

Building on Scene Map, we introduce MapPolicy, an imitation learning framework that fuses map features with visual and robot state features for action prediction. MapPolicy uses Structure-Modulated Graph Attention to propagate global structural and spatial information through primitive features, allowing visible parts and physical relations to inform the representation of hidden components. It also uses Physical Constraint Loss to regularize learned map features with geometric and kinematic consistency. Together, these components help the policy infer task-relevant structure from partial observations and use the inferred structure for manipulation.

We evaluate MapPolicy on 49 simulation tasks across MetaWorld, RLBench, and ManiSkill, together with five real-world tasks. Across these settings, MapPolicy consistently improves over visual, diffusion-based, and 3D policy baselines on articulated interaction, contact-rich control, and precise spatial arrangement.

Our contributions can be summarized as follows: **(1)** We tackle partial observability by introducing Scene Map as a structural scaffold that models object shape, part structure, spatial layout, and physical constraints. By turning occluded but task-relevant structure into prediction targets and constraining these predictions with physical relations, Scene Map provides the policy with cues for inferring

unseen components from partial observations. **(2)** We propose MapPolicy, a policy framework that incorporates Scene Map into imitation learning by integrating structured scene representations with visual and robot state observations. **(3)** We design two mechanisms that make structural priors effective for policy learning: Structure-Modulated Graph Attention for propagating global structural and spatial information through primitive features, and Physical Constraint Loss for enforcing geometric and kinematic consistency in learned map features. **(4)** We conduct comprehensive evaluations across 49 simulation tasks in MetaWorld, RLBench, and ManiSkill, together with five real-world tasks, demonstrating consistent performance gains in partial-observability, articulated-interaction, contact-rich manipulation, and precise spatial-arrangement settings.

2 Related Works

Imitation Learning for Robot Manipulation Imitation Learning (IL) provides an efficient paradigm for robot manipulation by learning a mapping function from observations to actions from demonstrations. Classical approaches are closely related to inverse reinforcement learning [36, 1]. Modern behavior cloning trains policies with supervised objectives [47, 48, 44, 61, 12]. Multi-task and generalist manipulation policies, such as PerAct [53], CLIPort [52], Lift3D [27], Baku [23], Track2Act [5], Gen2Act [4], RoboAgent [3], and large-scale robot learning models [37], demonstrate the benefit of stronger perception backbones, larger corpora, and more expressive policy parameterizations.

Recent generative IL methods enhance generalization and robustness. OSVI-WM [19] introduced an end-to-end imitation learning architecture trained solely on in-domain data without large-scale pretraining and a world-model-guided trajectory generation module tailored for unseen tasks. Latent Policy Barrier [58] implicitly treats the latent expert demonstration distribution itself as a barrier that separates safe, in-distribution states from unsafe, out-of-distribution regions. 3D Equivariant Visuomotor Policy [24] introduces an SO(3)-equivariant policy learning framework that uses spherical projection from 2D RGB inputs to model 3D symmetries.

Object-Centric Representations for Manipulation Object-centric representations have been widely explored as a way to improve generalization in robot manipulation. Early methods use object poses, bounding boxes, or graph-structured object abstractions to decouple manipulation policies from raw pixels [54, 65, 14, 62, 64, 34]. A broader object-centric learning literature studies how to discover or refine object slots, keypoints, and structured object latents from visual data [30, 21, 15, 31, 32, 40, 72, 46, 8, 16, 49]. More recent works leverage learned, foundation-model, or pretrained object-centric features to construct stronger scene encoders for manipulation, including object proposal priors [78], object-centric 3D policy representations [77], 3D semantic fields [66], grounded spatial value maps [11], pretrained object-centric scene encoders [41], composed "what" and "where" object representations [51], and task-oriented hierarchical object decomposition [42]. These methods improve the policy's ability to localize task-relevant entities and reduce reliance on global image appearance.

Despite their effectiveness, these methods primarily improve the representation or learning of the observed part of the scene. In contrast, our approach equips the policy with the ability to reason about unseen structure, addressing partial observability from a more fundamental perspective.

3 Method

3.1 Method Overview

MapPolicy is a structured imitation learning framework that incorporates an explicit Scene Map. At each time step, the system first constructs a scene graph from the current RGB-D observation, then encodes the graph into compact map features, and finally fuses these features with visual evidence and robot state for action prediction. In compact form, $\mathcal{G}_t = F_{\text{ctor}}(I_t, D_t)$, $\mathbf{z}_{map} = F_{\text{enc}}(\mathcal{G}_t)$, and $\hat{a}_t = \pi_\theta(\mathbf{z}_{map}, \mathbf{z}_{vis}, \mathbf{z}_{state})$, where I_t and D_t denote RGB and depth observations, $\mathbf{z}_{vis} = F_{\text{vis}}(I_t)$ is the visual feature, and $\mathbf{z}_{state} = F_{\text{state}}(s_t)$ is the robot state feature.

Figure 2 summarizes the full pipeline. *Map Constructor* instantiates Scene Map from sensor observations, *Map Encoder* turns the graph into relation-aware map features, and the *Policy Backbone*

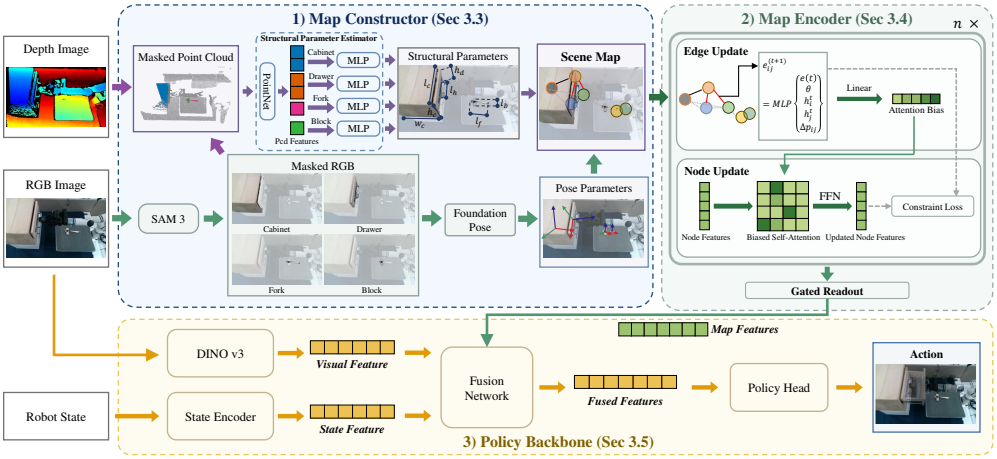

Figure 2: Overall pipeline of MapPolicy. The Map Constructor instantiates a Scene Map from RGB-D observations, the Map Encoder transforms the graph into map features, and the Policy Backbone fuses these features with visual and robot state information for action prediction.

combines these features with visual and proprioceptive features to predict actions. This decomposition makes the role of Scene Map explicit: it is not an external annotation consumed only during training, but the intermediate representation through which the policy reasons about scene organization under partial observability.

3.2 Scene Map

As shown in fig. 3, Scene Map is a structured representation. It represents a manipulation scene as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each object is represented as an object-level subgraph, and the full scene map is obtained by composing these subgraphs together with cross-object relations. In the Scene Map example in fig. 3, object parts are decomposed into primitives, and their connectivity forms a graph that preserves object structure and spatial layout even when some parts are not directly visible.

The nodes in Scene Map correspond to geometric primitives, including cuboids, cylinders, spheres, and rings as illustrated in fig. 3. A node stores the primitive pose, shape or scale parameters, semantic identity, and affordance-related geometry. In compact form, we write a node as $v_i = \langle \mathbf{p}_i, \mathbf{r}_i, X_i^{pos}, s_i, X_i^{aff} \rangle$, where X_i^{pos} samples the primitive geometry and X_i^{aff} samples manipulation-relevant regions. This primitive-level decomposition preserves object shape and part layout without collapsing them into a single object descriptor.

The edges in Scene Map represent physical constraints between primitives. Each edge is written as $e_{ij} = \langle c_{ij}, \theta_{ij}, a_{ij}, T_{ij} \rangle$, where c_{ij} is the constraint type, θ_{ij} contains continuous relation parameters, a_{ij} selects the geometric anchors used by the relation, and T_{ij} is the relative pose. These constraints are derived from the local geometry of primitives. For example, face anchors provide contact planes and normal directions, while axis anchors provide center lines and axial directions. The right part of fig. 3 describes the allowed motion of primitive B relative to primitive A, where red crosses denote forbidden translation or rotation and green arrows denote free degrees of freedom. For example, a cylindrical constraint is defined between two axis anchors: it enforces the two axes to be coaxial by constraining their direction mismatch and radial offset, while leaving translation along the axis and rotation around the axis unconstrained. Detailed primitive definitions and all six constraint types are provided in Appendix A.1.

3.3 Map Constructor

Scene Map defines *what* structural information should be represented, while the constructor defines *how* that information is instantiated from the current scene. Following the pipeline in fig. 2, the observation is first organized into object-level evidence, including object geometry and pose estimates.

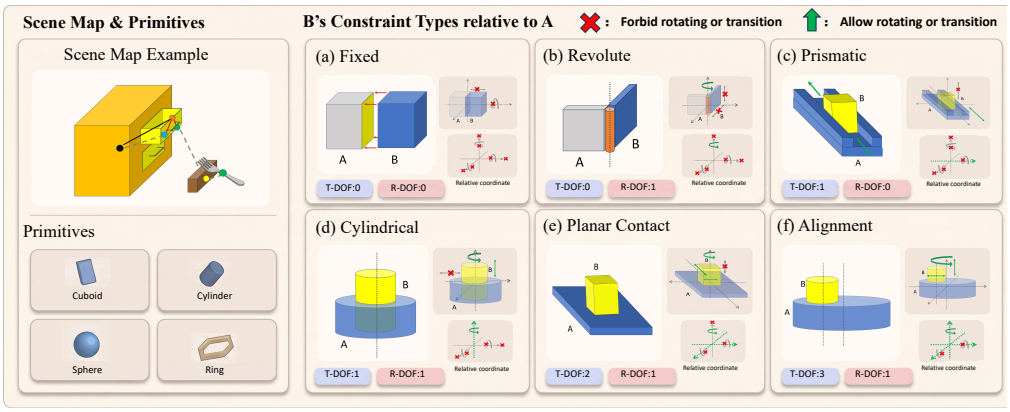

Figure 3: Scene Map. The left panel shows a scene-map example and the primitive set used to decompose object structure, including cuboid, cylinder, sphere, and ring. The right panel summarizes the motion of object B relative to object A under six physical constraint types, where red crosses indicate forbidden translation or rotation and green arrows indicate allowed degrees of freedom.

Firstly, SAM 3 [7] is used to segment foreground objects from the RGB image, and FoundationPose [68, 67] is then applied to estimate their pose parameters. In parallel, the depth image and object masks are projected into masked point clouds. A structural parameter estimator, implemented with PointNet [9] followed by separate MLP heads, encodes these point clouds and predicts the structural parameters. Finally, the estimated pose and structural parameters are combined to instantiate the Scene Map.

3.4 Map Encoder

As shown in fig. 2(2), Map Encoder transforms the instantiated Scene Map into compact map features $\mathbf{z}_{map} = F_{\text{enc}}(\mathcal{G}_t)$ for control. The encoder keeps node and edge streams separate at the input stage. For each node, primitive geometry, affordance samples, and semantic descriptors are embedded separately and fused into the initial node embedding $\mathbf{h}_i^0$. For each edge, relation type, continuous relation parameters, anchor descriptors, and relative pose are embedded and fused into the initial relation embedding $\mathbf{e}_{ij}^0$. This preserves both part-level attributes and explicit relation attributes before message passing.

For batched graph attention, the sparse Scene Map is packed into fixed-size node and edge tensors, $\mathbf{H}^0 \in \mathbb{R}^{B\times N\times d}$ and $\mathbf{E}^0 \in \mathbb{R}^{B\times N\times N\times d}$, where valid edge entries carry the relation attributes recovered by the constructor. We apply a stack of Graphormer-style layers on these tensors, adapted from graph transformer mechanisms [50, 70, 43]. In each layer, each relation embedding is updated from its previous state and the two endpoint node states. The updated relation is then projected into a head-specific additive attention bias, which modulates node self-attention:

$$b_{ij,r}^{\ell} = \mathbf{w}_{b,r}^{\top}\mathbf{e}_{ij}^{\ell}, \qquad \alpha_{ij,r}^{\ell} = \text{softmax}_j\left(\frac{(\mathbf{q}_{i,r}^{\ell})^{\top}\mathbf{k}_{j,r}^{\ell}}{\sqrt{d_h}} + b_{ij,r}^{\ell}\right), \tag{1}$$

where r indexes attention heads. Node features are updated by the biased multi-head attention block followed by a feed-forward block with residual connections. Consequently, information can propagate between parts according to both node content and the explicit relation connecting them. After the final layer, a gated readout computes attention weights over node states and sums them into $\mathbf{z}_{map}$. The resulting map feature is compact enough to fuse with visual and robot state features, while still being produced through edge-conditioned reasoning over the explicit Scene Map structure.

3.5 Policy Backbone

As shown in fig. 2(3), Policy Backbone predicts actions from the fused multimodal representation. In parallel with map encoding, a visual encoder extracts $\mathbf{z}_{vis}$ from the current RGB observation, and a state encoder extracts $\mathbf{z}_{state}$ from proprioceptive input. The map feature, visual feature, and robot

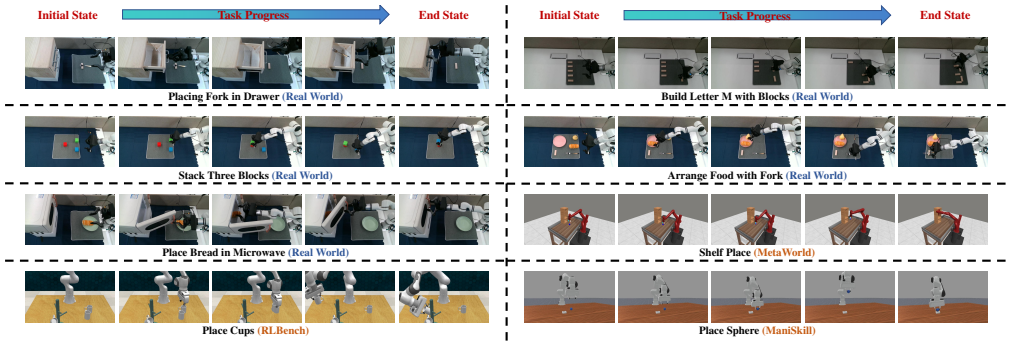

Figure 4: Demonstrations of the evaluation tasks. The figure shows all five real-world tasks and representative tasks from MetaWorld [71, 33], RLBench [25], and ManiSkill [22], illustrating the diversity of articulated interaction, object arrangement, and geometrically constrained manipulation considered in our experiments.

state feature are separately projected into modality tokens, fused by self-attention, and passed to an MLP policy head to predict $\hat{a}_t$. The three branches play complementary roles. The visual branch captures currently visible evidence, the map branch provides structural priors and relation-aware context, and the state branch provides the robot's internal configuration. The policy backbone does not reconstruct scene structure again. It consumes the map features produced by the Map Encoder and turns the fused representation into a manipulation action.

3.6 Loss Functions and Implementation Details

MapPolicy uses losses at both stages of training. For Map Constructor training, the objective is $\mathcal{L}_{ctor} = \mathcal{L}_{pcd} + \lambda_{struct}\mathcal{L}_{struct}$. Here $\mathcal{L}_{pcd}$ compares the point cloud rendered from the predicted Scene Map and projected into the camera frame with the observed masked point cloud, while $\mathcal{L}_{struct}$ is a mean-squared error on structural parameters. This supervision encourages the constructed Scene Map to be both geometrically consistent with RGB-D observations and structurally aligned with the task-specific map template.

For policy training, the objective is $\mathcal{L}_{policy} = \mathcal{L}_{act} + \lambda_{phys}\mathcal{L}_{phys}$. The action loss $\mathcal{L}_{act}$ is the imitation loss on expert actions. The physical constraint loss $\mathcal{L}_{phys}$ regularizes map features by decoding edge-conditioned representations into anchor states and penalizing violations of the corresponding geometric or kinematic constraints. Detailed constraint residuals are provided in Appendix A.1.

MapPolicy is trained in two stages. We first train the structural parameter estimator in the Map Constructor, and then freeze the constructor while training the Map Encoder and Policy Backbone. At inference time, the system follows the same observation-to-map-to-action pipeline: RGB-D observations provide object masks, poses, and masked point clouds, which instantiate a Scene Map that is encoded and fused with visual and robot state features for action prediction. The policy backbone uses a linear state encoder, attention-based multimodal fusion, and an MLP policy head. Full implementation details are provided in Appendix A.4.

4 Experiments

This section evaluates whether Scene Map improves downstream manipulation policy learning beyond raw visual or geometric input. We first report main results on simulation benchmarks in sec. 4.1, then evaluate sim-to-real transfer in sec. 4.2. We further analyze Scene Map construction bottlenecks in sec. 4.3 and validate key design choices through ablations in sec. 4.4. Finally, we report the latency of MapPolicy in sec. 4.5

4.1 Main Experiments in Simulation Benchmarks

We evaluate MapPolicy on three simulation benchmarks with complementary characteristics: Meta-World [71, 33], RLBench [25], and ManiSkill [22]. Together, they cover classical single-task manipulation, large-scale multi-task interaction, and tabletop manipulation.

Table 1: Summary of main simulation results. We report average success rate (%) across all evaluated tasks in each benchmark. Full per-task results are provided in Appendix A.7.

MetaWorld [71, 33] (25 tasks)		RLBench [25] (18 tasks)		ManiSkill [22] (6 tasks)	
Method	Mean S.R. (%)	Method	Mean S.R. (%)	Method	Mean S.R. (%)
Diffusion Policy [13]	27.3	PerAct [53]	49.4	Diffusion Policy [13]	26.8
3D Diffusion Policy [74]	51.1	3D Diffuser Actor [28]	81.4	3D Diffusion Policy [74]	53.8
π_0 [6]	69.5	RVT-2 [20]	81.4	Octo [17]	16.8
Lift3D [27]	<u>78.8</u>	PDFactor [59]	<u>87.3</u>	π_0 [6]	<u>54.0</u>
MapPolicy	**93.2**	**MapPolicy**	**94.8**	**MapPolicy**	**73.0**

Table 2: Real-world results. We report success rate (%) over 100 trials for each real-world manipulation task. Best results are in **bold**, and second best are <u>underlined</u>.

Method	Avg.	Place Fork in Drawer	Stack Three Blocks	Arrange Food with Fork	Place Bread in Microwave	Build Letter M with Blocks
Diffusion Policy [13]	15.4	12	6	41	18	0
3D Diffusion Policy [74]	33.6	38	21	66	43	0
Lift3D [27]	36.2	41	25	69	46	0
π_0 [6]	<u>49.8</u>	<u>51</u>	<u>39</u>	<u>83</u>	<u>61</u>	<u>15</u>
MapPolicy (Ours)	**81.4**	**87**	**78**	**95**	**84**	**63**

Experiment Settings. We benchmark MapPolicy on three suites. **MetaWorld** [71, 33] contains 25 tasks in our evaluation and is used to test whether Scene Map helps on diverse classical MuJoCo-based manipulation problems requiring part-level interaction and precise spatial reasoning [60]. **RLBench** [25] contains 18 tasks in our evaluation and is used to test whether the benefits of Scene Map persist in visually richer and more diverse multi-task settings. **ManiSkill** [22] contains six tasks from the official *Table-Top 2 Finger Gripper Tasks* category, including *PegInsertionSide-v1*, *PlaceSphere-v1*, *PlugCharger-v1*, *PullCubeTool-v1*, *StackCube-v1*, and *StackPyramid-v1*, and is used to test MapPolicy on tabletop RGB-D manipulation with strong geometric and contact constraints. Across all three benchmarks, the main metric is task success rate. We report mean and standard deviation across five random seeds for the main simulation results. Detailed training and evaluation protocols are provided in Appendix A.5.

Baselines. We compare against strong baselines from three categories: diffusion-based policies, general-purpose visuomotor foundation policies, and 3D manipulation policies. On MetaWorld, we compare against *Diffusion Policy* [13], *3D Diffusion Policy* [74], π_0 [6], and *Lift3D* [27]. On RLBench, we compare against published baselines reported in prior work, including *PerAct* [53], *3D Diffuser Actor* [28], *RVT-2* [20], and the *PDFactor* family [59], using their reported numbers. On ManiSkill, we compare against *Diffusion Policy* [13], *3D Diffusion Policy* [74], *Octo* [17], and π_0 [6].

Results. MetaWorld [71, 33] tests classical manipulation tasks requiring actionable-part identification and local spatial reasoning. As shown in tab. 1, MapPolicy achieves 93.2% average success rate, outperforming the strongest prior baseline Lift3D at 78.8%. The gains are particularly meaningful on articulated interaction, contact-rich manipulation, and precise placement. Full per-task results are provided in Appendix A.7. **RLBench** [25] evaluates larger-scale multi-task scenes with richer visual variation and multi-object interaction. MapPolicy reaches 94.8% average success rate, improving over the strongest published baseline PDFactor at 87.3%, indicating that Scene Map benefits extend beyond classical manipulation settings. **ManiSkill** [22] focuses on tabletop tasks with strong geometric constraints such as insertion, plugging, stacking, and tool use. MapPolicy obtains 73.0% average success rate, outperforming π_0 at 54.0% and 3D Diffusion Policy at 53.8%, with the largest gains on tasks where explicit structure is directly relevant to action success.

Controlled Partial-Observability Evaluation. We evaluate *Shelf Place* and *Door Open* in MetaWorld [71, 33] under 0–80% occlusion using spatially contiguous masks on RGB and depth. At 60% and 80% occlusion, MapPolicy achieves 81% and 67% success on *Shelf Place*, and 91% and 82% on *Door Open*, substantially outperforming all baselines in tab. 3. At 60% occlusion, the door handle is nearly invisible, yet MapPolicy loses only 7 points and the Scene Map retains a 55.9% F-Score in tab. 4, demonstrating robust hidden-structure inference.

Table 3: Task success rate under controlled occlusion on MetaWorld. All values are percentages.

Task	Method	0%	20%	40%	60%	80%	Task	Method	0%	20%	40%	60%	80%
Shelf Place	Diffusion Policy	11	12	8	3	2	*Door Open*	Diffusion Policy	**100**	96	78	24	16
Shelf Place	3D Diffusion Policy	17	15	12	5	3	*Door Open*	3D Diffusion Policy	**100**	97	84	38	29
Shelf Place	π_0	35	34	32	16	10	*Door Open*	π_0	**100**	**100**	91	47	33
Shelf Place	Lift3D	42	36	30	18	5	*Door Open*	Lift3D	**100**	**100**	93	53	38
Shelf Place	**MapPolicy (Ours)**	**94**	**92**	**93**	**81**	**67**	*Door Open*	**MapPolicy (Ours)**	**100**	99	**98**	**91**	**82**

Table 4: Scene Map construction accuracy under controlled occlusion.

Task	Occlusion	CD ($\times 10^{-3}$) $\downarrow$	EMD ($\times 10^{-2}$) $\downarrow$	F-Score (%) $\uparrow$	Task	Occlusion	CD ($\times 10^{-3}$) $\downarrow$	EMD ($\times 10^{-2}$) $\downarrow$	F-Score (%) $\uparrow$
Shelf Place	0%	**9.12**	**4.83**	**60.8**	*Door Open*	0%	**8.97**	**4.78**	**61.1**
Shelf Place	20%	9.26	4.91	60.3	*Door Open*	20%	9.18	4.89	60.5
Shelf Place	40%	9.51	5.08	59.6	*Door Open*	40%	9.43	5.11	59.4
Shelf Place	60%	10.84	5.89	56.4	*Door Open*	60%	11.02	5.96	55.9
Shelf Place	80%	15.37	8.46	46.7	*Door Open*	80%	15.61	8.35	46.1

4.2 Real-World Experiments

We evaluate whether the Scene Map interface transfers from simulation to real RGB-D manipulation on a Realman 75-BF robot arm with 7 DoF and a pika gripper, with its low-level controller running at 30 Hz. RGB-D observations are captured by a RealSense D415, and point clouds are constructed with Open3D [75]. We reuse representative baselines from our simulation study for the real-world setting. In the current real-robot evaluation, we directly deploy *Diffusion Policy* [13] and *3D Diffusion Policy* [74] as the primary reproducible baselines, while π_0 [6] and *Lift3D* [27] provide additional reference points for method positioning. For each real-world task, we execute each method for 100 trials and report task success rate as the main metric.

Task Suite. We design five real-world tasks that cover articulated interaction, reflective-object grasping, multi-object spatial reasoning, and long-horizon arrangement. *Place Fork in Drawer* consists of four stages: opening a closed drawer, picking up a fork placed on a block, placing it into the drawer, and closing the drawer. This task is challenging because it combines articulated-object manipulation with grasping a thin, irregular, and reflective metal object. *Stack Three Blocks* requires first placing a green block on a red block and then placing a blue block on the green block. The main difficulty is maintaining the correct spatial relation across three objects during sequential stacking. *Arrange Food with Fork* consists of placing bread on a plate, placing a pear on the plate, and then placing a fork on a block stand. Although this is a more common tabletop arrangement task, it still tests compositional multi-object placement in a cluttered real scene. *Place Bread in Microwave* consists of opening the microwave door, transferring bread from a plate into the microwave, and closing the door. This task emphasizes articulated manipulation under partial occlusion and changing scene geometry. *Build Letter M with Blocks* requires arranging five blocks into a specified global shape. Unlike local pick-and-place tasks, this setting tests whether the policy can reason about a target spatial pattern and adjust placements online to satisfy the final global arrangement.

Results. As shown in tab. 2, we report success rate over 100 real-robot trials for each task and each method. MapPolicy transfers robustly to the real world and is particularly effective on tasks where success depends on understanding hidden structure, articulated state, or global spatial layout. Compared with Diffusion Policy [13] and 3D Diffusion Policy [74], MapPolicy produces more stable multi-step behavior on drawer and microwave tasks and better preserves the required inter-object arrangement on stacking and shape-construction tasks. These results are consistent with our central claim: explicit Scene Map helps the policy maintain a manipulation-oriented representation of object structure and spatial relations even when raw observations are noisy, partially occluded, or insufficient to reveal the full scene state.

4.3 System Error Breakdown

MapPolicy is a multi-stage system whose final performance depends not only on downstream policy learning, but also on the quality of Scene Map construction. To identify the dominant bottleneck, we perform a system error breakdown on MetaWorld [71, 33] using oracle replacements. Since several error sources in MapPolicy can be independently controlled, we use a combinatorial oracle analysis instead of a single sequential replacement protocol.

Table 5: System error breakdown on MetaWorld [71, 33]. We use oracle replacements to identify the dominant bottleneck in MapPolicy. Oracle quantities are used only for analysis and are not part of the inference interface. Δ reports the improvement over Full MapPolicy.

Setting	GT Extraction	GT Object Pose	GT Structural Size	Avg. Success	Δ
Full MapPolicy				93.2	0.0
+ GT Extraction	✓			93.4	+0.2
+ GT Object Pose		✓		94.0	+0.8
+ GT Structural Size			✓	94.6	+1.4
+ GT Extraction + GT Object Pose	✓	✓		93.4	+0.2
+ GT Extraction + GT Structural Size	✓		✓	95.1	+1.9
+ GT Object Pose + GT Structural Size		✓	✓	95.9	+2.7
+ GT Extraction + GT Object Pose + GT Structural Size	✓	✓	✓	96.0	+2.8

Table 6: Ablation study on Scene Map representation by varying node and edge attributes on MetaWorld [71, 33].

	Node				Edge				
	Geo	Pose	Sem	Aff	Type	Param	Anchor	RelPose	**Mean**
Ex1	✓	-	-	-	-	-	-	-	**72.4**
Ex2	✓	✓	-	-	-	-	-	-	**78.1**
Ex3	✓	✓	✓	✓	-	-	-	-	**83.5**
Ex4	✓	✓	✓	✓	✓	-	-	-	**86.7**
Ex5	✓	✓	✓	✓	✓	✓	-	-	**88.9**
Ex6	✓	✓	✓	✓	✓	✓	✓	-	**91.4**
Ex7	✓	✓	✓	✓	✓	✓	✓	✓	**93.2**

Table 7: Ablation study on map encoder and multimodal fusion design choices on MetaWorld [71, 33].

	Map Encoder				Fusion			
	Pool	PN	Trans.	Graph	Concat	MLP	Attn	**Mean**
Ex1	✓	-	-	-	-	-	✓	**75.6**
Ex2	-	✓	-	-	-	-	✓	**82.3**
Ex3	-	-	✓	-	-	-	✓	**89.1**
Ex4	-	-	-	✓	-	-	✓	**93.2**
Ex5	-	-	-	✓	✓	-	-	**84.2**
Ex6	-	-	-	✓	-	✓	-	**87.6**
Ex7	-	-	-	✓	-	-	✓	**93.2**

Note: Geo = part geometry, Sem = semantic identity, Aff = affordance attributes. Param = continuous relation parameters, RelPose = relative pose. PN = PointNet [9], Trans. = Transformer, Graph = graph encoder, Attn = attention fusion. Darker blue indicates better performance.

Setup. We analyze three independent sources of structural error: object extraction, object-pose estimation, and structural-size estimation. For each source, we replace the predicted quantity with its ground-truth counterpart and evaluate the resulting average success rate. All oracle quantities in this subsection are used only for analysis and are not part of the inference interface of MapPolicy.

Results. The main result is summarized in tab. 5, formatted as a 2D oracle matrix. The table contains three oracle columns, *GT Extraction*, *GT Object Pose*, and *GT Structural Size*, together with one performance column, *Avg. Success*. The rows enumerate all combinations of the three oracle sources. This analysis should be read in two stages. First, the single-factor replacements reveal which structural error source is the dominant independent bottleneck. Second, the joint replacements reveal whether different error sources interact strongly or contribute largely independent errors. This subsection therefore identifies which part of the MapPolicy pipeline deserves the highest priority for improvement. GT extraction, object pose, and structural size improve success by 0.2, 0.8, and 1.4 points, while replacing all three improves it by 2.8 points. These limited gains show that upstream errors do not accumulate catastrophically and identify structural-size estimation as the main bottleneck.

4.4 Ablation Studies

We design ablations to answer three questions: whether explicit Scene Map is necessary as a representation, how Scene Map should be encoded and fused with visual information, and which supervision terms make the learned map most useful for downstream control.

Scene Map Representation. We first evaluate which contents inside Scene Map are most important for policy learning. We conduct this ablation on MetaWorld [71, 33] and keep the policy backbone and training recipe fixed, changing only the attributes included in the map representation. Table 6 progressively adds node attributes, including part geometry, pose, semantic identity, and affordance-related information, and then edge attributes, including relation type, continuous relation parameters, anchor references, and relative pose. This comparison separates the contribution of part-level node descriptors from the contribution of explicit cross-part relations. If edge attributes provide additional gains beyond node-only representations, the conclusion is that Scene Map helps not only by describing object parts, but also by modeling the structural and physical relations among them.

Structure-Aware Encoder and Fusion. Given a Scene Map, the next question is how it should be encoded and fused with visual information. We therefore study both the choice of map encoder and the choice of fusion strategy. Table 7 compares four encoders while keeping the Scene Map representation fixed: *Pooling*, *PointNet* [9], *Transformer* [29], and *Graph Encoder*. This comparison tests whether relation-aware graph encoding is necessary to fully exploit the node-edge structure of

Table 8: Ablation study on training losses on MetaWorld [71, 33]. Map Attr. GT supervises the scene-dependent node and edge attributes used to instantiate Scene Map, Chamfer supervises map-induced point-cloud geometry, and Physical enforces relation-specific constraints.

	Loss Terms						Loss Terms				
Setting	Action	Map Attr. GT	Chamfer	Physical	Avg. Success	Setting	Action	Map Attr. GT	Chamfer	Physical	Avg. Success
Ex1	✓	-	-	-	80.6	Ex5	✓	✓	✓	-	88.7
Ex2	✓	✓	-	-	83.4	Ex6	✓	✓	-	✓	89.5
Ex3	✓	-	✓	-	84.1	Ex7	✓	-	✓	✓	92.4
Ex4	✓	-	-	✓	82.8	**Ex8**	✓	✓	✓	✓	**93.2**

Table 9: Inference latency. All results are measured on an NVIDIA GeForce RTX 4090 with batch size 1. Latency is reported in milliseconds (ms).

Inference Stage	SAM [7] (ms) ↓	FoundationPose [68] (ms) ↓	Map Instantiation (ms) ↓	Policy (ms) ↓	End-to-End (ms) ↓	FPS ↑
First Frame	1404.291	1394.299	20.764	14.975	2834.329	0.353
Subsequent Frames	96.008	95.263	20.764	14.975	227.010	4.405
Pose-Only Tracking	–	95.263	–	14.975	110.238	9.1

Scene Map. We then fix the best map encoder and compare three fusion strategies: *Concat*, *MLP fusion*, and *Attention fusion*. These results show whether the information in Scene Map can be effectively used by naive concatenation or whether a stronger fusion mechanism is needed to integrate structural and visual cues for downstream action prediction.

Loss Design. We keep action imitation enabled and ablate three auxiliary losses: map-attribute supervision, point-cloud Chamfer supervision, and physical constraint regularization. Table 8 shows that each auxiliary loss improves over action imitation alone, while combining Chamfer supervision with physical constraints gives the strongest partial combination and using all three losses achieves the best performance.

4.5 Inference Latency

Table 9 reports the latency of MapPolicy. The first frame constructs the Scene Map with a one-time cost of 2834.329 ms. Later frames track masks and poses at 227.010 ms, or 4.405 FPS. We use 30 warm-up runs and 100 timed runs per template. SAM [7] and FoundationPose [68] tracking latency is averaged over 100 frames. For higher control rates, pose-only tracking caches object geometry, topology, and physical constraints. Later frames run only FoundationPose [68] and the policy, reducing latency to 110.238 ms, or 9.1 FPS. Full construction restarts after tracking failure or when a new object appears.

5 Conclusion

We studied robot manipulation under partial observability and argued that the core challenge is missing task-relevant structure rather than missing visual evidence alone. We introduced Scene Map, a graph representation of object shape, part structure, spatial layout, and physical constraints, and proposed MapPolicy to fuse map, visual, and robot state features for imitation learning. Across 49 simulation tasks in MetaWorld, RLBench, and ManiSkill, together with five real-world tasks, MapPolicy consistently improves over strong baselines on partial observability, articulated interaction, contact-rich manipulation, and precise spatial arrangement. These results suggest that explicit structural priors can improve manipulation policy learning beyond raw visual input or unstructured geometry alone. Future work should improve the robustness and generality of Scene Map construction for more open-ended real-world environments.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62606317, the Science and Technology Major Project of Jiangsu Province (Grant No. BG2025035), Shanghai Artificial Intelligence Laboratory, and the XPLORER PRIZE. The authors declare no competing interests.

References

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 1, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015430. URL https://doi.org/10.1145/1015330.1015430.

[2] Iretiayo Akinola, Jacob Varley, and Dmitry Kalashnikov. Learning precise 3d manipulation from multiple uncalibrated cameras. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4616–4622, 2020. doi: 10.1109/ICRA40945.2020.9197181.

[3] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4788–4795, 2024. doi: 10.1109/ICRA57147.2024.10611293.

[4] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. In Joseph Lim, Shuran Song, and Hae-Won Park, editors, *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 3936–3951. PMLR, 27–30 Sep 2025. URL https://proceedings.mlr.press/v305/bharadhwaj25a.html.

[5] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2Act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *Computer Vision – ECCV 2024*, volume 15134 of *Lecture Notes in Computer Science*, pages 331–349, Cham, 2025. Springer. doi: 10.1007/978-3-031-73116-7_18.

[6] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, Laura Smith, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A Vision-Language-Action Flow Model for General Robot Control. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.010.

[7] 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, editors, *International Conference on Learning Representations*, volume 2026, pages 138846–138923, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e0982cbc81401df3430ee1ff780dc7a2-Paper-Conference.pdf.

[8] Michael Chang, Tom Griffiths, and Sergey Levine. Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 32694–32708. Curran Associates, Inc., 2022. doi: 10.52202/068431-2369. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/d301e2878a7ebadf1a95029e904fc7d0-Paper-Conference.pdf.

[9] 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)*, pages 77–85, 2017. doi: 10.1109/CVPR.2017.16.

[10] Boyuan Chen, Pieter Abbeel, and Deepak Pathak. Unsupervised learning of visual 3d keypoints for control. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*,

pages 1539–1549. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/chen21b.html.

[11] Yangtao Chen, Zixuan Chen, Junhui Yin, Jing Huo, Pinzhuo Tian, Jieqi Shi, and Yang Gao. Gravmad: Grounded spatial value maps guided action diffusion for generalized 3d manipulation. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Learning Representations*, volume 2025, pages 102042–102074, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/fce176458ff542940fa3ed16e6f9c852-Paper-Conference.pdf.

[12] 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, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 3809–3829. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/chen23e.html.

[13] 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.

[14] Coline Devin, Pieter Abbeel, Trevor Darrell, and Sergey Levine. Deep object-centric representations for generalizable robot learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7111–7118, 2018. doi: 10.1109/ICRA.2018.8461196.

[15] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations*, 2019.

[16] Ke Fan, Zechen Bai, Tianjun Xiao, Dominik Zietlow, Max Horn, Zixu Zhao, Carl-Johann Simon-Gabriel, Mike Zheng Shou, Francesco Locatello, Bernt Schiele, Thomas Brox, Zheng Zhang, Yanwei Fu, and Tong He. Unsupervised open-vocabulary object localization in videos. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13701–13709, 2023. doi: 10.1109/ICCV51070.2023.01264.

[17] Dibya Ghosh, Homer Rich Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Quan Vuong, Ted Xiao, Pannag R Sanketi, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An Open-Source Generalist Robot Policy. In *Proceedings of Robotics: Science and Systems*, July 2024. doi: 10.15607/RSS.2024.XX.090.

[18] James Jerome Gibson. *The Ecological Approach to Visual Perception*. Houghton Mifflin, 1979. doi: 10.4324/9781315740218.

[19] Raktim Goswami, Prashanth Krishnamurthy, Yann LeCun, and Farshad Khorrami. Osvi-wm: One-shot visual imitation for unseen tasks using world-model-guided trajectory generation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 54725–54745. Curran Associates, Inc., 2025. doi: 10.52202/085713-1830. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/4eb32fef14a4b0dea2ab997ce5b05281-Paper-Conference.pdf.

[20] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. RVT-2: Learning Precise Manipulation from Few Demonstrations. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.055.

[21] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2424–2433. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/greff19a.html.

[22] 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.

[23] 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, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 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.

[24] Boce Hu, Dian Wang, David Klee, Heng Tian, Xupeng Zhu, Haojie Huang, Robert Platt, and Robin Walters. 3d equivariant visuomotor policy learning via spherical projection. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 5685–5715. Curran Associates, Inc., 2025. doi: 10.52202/085713-0201. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/08a4d011c7b518b7d1d5b8fad6ddb4c9-Paper-Conference.pdf.

[25] 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.

[26] Rishabh Jangir, Nicklas Hansen, Sambaran Ghosal, Mohit Jain, and Xiaolong Wang. Look closer: Bridging egocentric and third-person views with transformers for robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):3046–3053, 2022. doi: 10.1109/LRA.2022.3144512.

[27] Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilve Wang, Longzan Luo, Xiaoqi Li, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, and Shanghang Zhang. Lift3d policy: Lifting 2d foundation models for robust 3d robotic manipulation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17347–17358, 2025. doi: 10.1109/CVPR52734.2025.01617.

[28] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 1949–1974. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/ke25a.html.

[29] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Transformer-based deep imitation learning for dual-arm robot manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8965–8972, 2021. doi: 10.1109/IROS51168.2021.9636301.

[30] Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2019.

[31] Tejas D Kulkarni, Ankush Gupta, Catalin Ionescu, Sebastian Borgeaud, Malcolm Reynolds, Andrew Zisserman, and Volodymyr Mnih. Unsupervised learning of object keypoints for perception and control. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/dae3312c4c6c7000a37ecfb7b0aeb0e4-Paper.pdf.

[32] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/8511df98c02ab60aea1b2356c013bc0f-Paper.pdf.

[33] Reginald McLean, Evangelos Chatzaroulas, Luc McCutcheon, Frank Röder, Tianhe Yu, Zhanpeng He, K.R. Zentner, Ryan Julian, J Terry, Isaac Woungang, Nariman Farsad, and Pablo Samuel Castro. Meta-world+: An improved, standardized, rl benchmark. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference. Curran Associates, Inc., 2025. doi: 10.52202/085713-4960. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/da295a7037612bb3bfc1334a6f8fc8bd-Paper-Datasets_and_Benchmarks_Track.pdf.

[34] Toki Migimatsu and Jeannette Bohg. Object-centric task and motion planning in dynamic environments. *IEEE Robotics and Automation Letters*, 5(2):844–851, 2020. doi: 10.1109/LRA.2020.2965875.

[35] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy J. Mitra, and Leonidas J. Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. *ACM Transactions on Graphics*, 38(6), 2019.

[36] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

[37] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Booher, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi Jim Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick Tree Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundaresan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Rohan Baijal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shuran Song, Sichun Xu, Siddhant Haldar, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay

Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, and Zipeng Lin. Open x-embodiment: Robotic learning datasets and rt-x models : Open x-embodiment collaboration0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024. doi: 10.1109/ICRA57147.2024.10611477.

[38] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10336–10345, 2019. doi: 10.1109/CVPR.2019.01059.

[39] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1057–1067, 2020. doi: 10.1109/CVPR42600.2020.00114.

[40] Jiahui Qian, Andreas Panagopoulos, and Dinesh Jayaraman. Discovering deformable keypoint pyramids. In *Computer Vision – ECCV 2022*, volume 13686 of *Lecture Notes in Computer Science*, pages 528–545, Cham, 2022. Springer. doi: 10.1007/978-3-031-19809-0_31.

[41] Jianing Qian, Anastasios Panagopoulos, and Dinesh Jayaraman. Recasting generic pretrained vision transformers as object-centric scene encoders for manipulation policies. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 17544–17552, 2024. doi: 10.1109/ICRA57147.2024.10610131.

[42] Jianing Qian, Yunshuang Li, Bernadette Bucher, and Dinesh Jayaraman. Task-oriented hierarchical object decomposition for visuomotor control. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 1891–1909. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/qian25b.html.

[43] Ladislav Rampášek, Michael Galkin, Vijay Prakash Dwivedi, Anh Tuan Luu, Guy Wolf, and Dominique Beaini. Recipe for a general, powerful, scalable graph transformer. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 14501–14515. Curran Associates, Inc., 2022. doi: 10.52202/068431-1054. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/5d4834a159f1547b267a05a4e2b7cf5e-Paper-Conference.pdf.

[44] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

[45] Hyunwoo Ryu, Jiwoo Kim, Hyunseok An, Junwoo Chang, Joohwan Seo, Taehan Kim, Yubin Kim, Chaewon Hwang, Jongeun Choi, and Roberto Horowitz. Diffusion-edfs: Bi-equivariant denoising generative modeling on se(3) for visual robotic manipulation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18007–18018, 2024. doi: 10.1109/CVPR52733.2024.01705.

[46] Bruno Sauvalle and Arnaud de La Fortelle. Unsupervised multi-object segmentation using attention and soft-argmax. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3267–3276, 2023. doi: 10.1109/WACV56688.2023.00328.

[47] Stefan Schaal. Learning from demonstration. In M.C. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9.

MIT Press, 1996. URL https://proceedings.neurips.cc/paper_files/paper/1996/file/68d13cf26c4b4f4f932e3eff990093ba-Paper.pdf.

[48] Stefan Schaal, Jan Peters, Jun Nakanishi, and A.J. Ijspeert. Learning movement primitives. *International Journal of Robotic Research - IJRR*, 15:561–572, 01 2003. doi: 10.1007/11008941_60.

[49] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *International Conference on Learning Representations*, 2022.

[50] Ahsan Shehzad, Feng Xia, Shagufta Abid, Ciyuan Peng, Shuo Yu, Dongyu Zhang, and Karin Verspoor. Graph transformers: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 37(7):3052–3071, 2026. doi: 10.1109/TNNLS.2025.3646122.

[51] Junyao Shi, Jianing Qian, Yecheng Jason Ma, and Dinesh Jayaraman. Composing pre-trained object-centric representations for robotics from "what" and "where" foundation models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15424–15432, 2024. doi: 10.1109/ICRA57147.2024.10610695.

[52] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Cliport: What and where pathways for robotic manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 894–906. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/shridhar22a.html.

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

[54] Maximilian Sieb, Zhou Xian, Audrey Huang, Oliver Kroemer, and Katerina Fragkiadaki. Graph-structured visual imitation. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 979–989. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/sieb20a.html.

[55] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seung Eun Yi, Michael Ramamonjisoa, Francisco Massa, Daniel HAZIZA, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Herve Jegou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=2N1GyqNjns.

[56] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B. Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se(3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400, 2022. doi: 10.1109/ICRA46639.2022.9812146.

[57] Jianhua Sun, Yuxuan Li, Longfei Xu, Nange Wang, Jiude Wei, Yining Zhang, and Cewu Lu. Conceptfactory: Facilitate 3d object knowledge annotation with object conceptualization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 75454–75467. Curran Associates, Inc., 2024. doi: 10.52202/079017-2402. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/89d19544d314740d11c0974ca3ddaf70-Paper-Datasets_and_Benchmarks_Track.pdf.

[58] Zhanyi Sun and Shuran Song. Latent policy barrier: Learning robust visuomotor policies by staying in-distribution. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz,

M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 174280–174305. Curran Associates, Inc., 2025. doi: 10.52202/085713-5797. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/feaf23597f3c93803d1b4230aadc8f44-Paper-Conference.pdf.

[59] Jingyi Tian, Le Wang, Sanping Zhou, Sen Wang, Jiayi Li, Haowen Sun, and Wei Tang. Pdfactor: Learning tri-perspective view policy diffusion field for multi-task robotic manipulation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15757–15767, 2025. doi: 10.1109/CVPR52734.2025.01469.

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

[61] Faraz Torabi, Garrett Warnell, and Peter Stone. Recent advances in imitation learning from observation. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 6325–6331. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/882. URL https://doi.org/10.24963/ijcai.2019/882.

[62] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 306–316. PMLR, 29–31 Oct 2018. URL https://proceedings.mlr.press/v87/tremblay18a.html.

[63] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1466–1474, 2017. doi: 10.1109/CVPR.2017.160.

[64] Stephen Tyree, Jonathan Tremblay, Thang To, Jia Cheng, Terry Mosier, Jeffrey Smith, and Stan Birchfield. 6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 13081–13088, 2022. doi: 10.1109/IROS47612.2022.9981838.

[65] Dequan Wang, Coline Devin, Qi-Zhi Cai, Fisher Yu, and Trevor Darrell. Deep object-centric policies for autonomous driving. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8853–8859, 2019. doi: 10.1109/ICRA.2019.8794224.

[66] Yixuan Wang, Guang Yin, Binghao Huang, Tarik Kelestemur, Jiuguang Wang, and Yunzhu Li. Gendp: 3d semantic fields for category-level generalizable diffusion policy. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard, editors, *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 4866–4878. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/wang25m.html.

[67] Bowen Wen, Jonathan Tremblay, Valts Blukis, Stephen Tyree, Thomas Müller, Alex Evans, Dieter Fox, Jan Kautz, and Stan Birchfield. Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 606–617, 2023. doi: 10.1109/CVPR52729.2023.00066.

[68] 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)*, pages 17868–17879, June 2024.

[69] Xinchen Yan, Jasmined Hsu, Mohammad Khansari, Yunfei Bai, Arkanath Pathak, Abhinav Gupta, James Davidson, and Honglak Lee. Learning 6-dof grasping interaction via deep geometry-aware 3d representations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3766–3773, 2018. doi: 10.1109/ICRA.2018.8460609.

[70] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances*

in Neural Information Processing Systems, volume 34, pages 28877–28888. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/f1c1592588411002af340cbaedd6fc33-Paper.pdf.

[71] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In Leslie Pack Kaelbling, Danica Kragic, and Komei Sugiura, editors, *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pages 1094–1100. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

[72] Wentao Yuan, Chris Paxton, Karthik Desingh, and Dieter Fox. Sornet: Spatial object-centric representations for sequential manipulation. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 148–157. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/yuan22a.html.

[73] 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.

[74] 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.

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

[76] Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 77799–77830. Curran Associates, Inc., 2024. doi: 10.52202/079017-2473. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8e5dc5969a6174fcaaececd890c7f59b-Paper-Datasets_and_Benchmarks_Track.pdf.

[77] 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, editors, *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 3418–3433. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/zhu23b.html.

[78] 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, editors, *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 1199–1210. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/zhu23a.html.

[79] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 900–909, 2017. doi: 10.1109/ICCV.2017.103.

A Technical appendices and supplementary material

Appendix organization. This appendix follows the order of the main paper. Appendix A.1 details the Scene Map representation and physical constraint loss used in sec. 3.2 and sec. 3.6. Appendix A.2 discusses computational complexity, Appendix A.3 summarizes the full MapPolicy procedure, and Appendix A.4 provides implementation details for the method. Appendix A.5 and Appendix A.7 provide experimental protocols and full per-task simulation results. Appendix A.8 and Appendix A.9 discuss limitations, future work, and broader impact.

The supplementary archive `supp_mappolicy.tar` contains a `supp/code/` directory with the released MapPolicy code, README, dependency list, Hydra configurations, Scene Map templates, dataset/environment wrappers, model implementations, and reproduction scripts, together with a `supp/demo/` directory containing five real-world demonstration videos.

A.1 Scene Map Representation and Constraint Loss

Scene Map representation. Scene Map represents each manipulation scene as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ built from task-relevant geometric primitives and their relations. Each object is represented as an object-level subgraph, and the complete scene map is formed by composing object subgraphs and cross-object relations. Each node stores a primitive-level descriptor

$$v_i = \langle \mathbf{p}_i, \mathbf{r}_i, X_i^{pos}, s_i, X_i^{aff} \rangle, \tag{2}$$

where $\mathbf{p}_i$ and $\mathbf{r}_i$ denote the primitive position and orientation, X_i^{pos} is a point set sampled from the primitive geometry, s_i is a semantic descriptor, and X_i^{aff} is an affordance point set sampled from manipulation-relevant regions. This representation keeps pose, geometry, semantics, and affordance information separate before they are embedded by the map encoder.

Each edge stores a relation between two primitives,

$$e_{ij} = \langle c_{ij}, \boldsymbol{\eta}_{ij}, a_{ij}, T_{ij} \rangle. \tag{3}$$

Here c_{ij} is the relation type, $\boldsymbol{\eta}_{ij} = (\delta, \alpha, \phi)$ contains continuous relation parameters, a_{ij} selects the geometric anchors used by the relation, and T_{ij} is the relative pose between the two primitives. We use six standard relation types: *Fixed*, *Revolute*, *Prismatic*, *Cylindrical*, *Planar-Contact*, and *Alignment*. These relations cover rigid attachment, common kinematic pairs, and spatial contact or alignment constraints.

Primitive definitions. Primitives provide a compact geometric basis for object structure. A cuboid primitive is parameterized by a pose and three side lengths, and it provides six face anchors whose centers, normals, tangents, and bitangents are determined by the local cuboid frame. A cylinder primitive is parameterized by a pose, radius, and height, and it provides an axis anchor along the cylinder center line as well as circular or side-surface contact regions when needed. A sphere primitive is parameterized by a pose and radius, and its geometry is sampled on the spherical surface. A ring primitive is parameterized by a pose, major radius, minor radius, and thickness, and it provides an axis anchor through the ring center together with surface samples around the annulus. Other task-specific primitives follow the same interface: they provide sampled geometry points, optional affordance samples, and the local anchors required by their incident edges.

Geometric anchors. Relations are expressed through local geometric anchors attached to primitives. A face anchor is represented by a local coordinate frame

$$a^F = \langle \mathbf{p}, \mathbf{n}, \mathbf{t}, \mathbf{b} \rangle, \tag{4}$$

where $\mathbf{p}$ is the face center, $\mathbf{n}$ is the normal direction, $\mathbf{t}$ is a tangent direction, and $\mathbf{b} = \mathbf{n} \times \mathbf{t}$ is the bitangent direction. An axis anchor is represented as

$$a^A = \langle \mathbf{p}, \mathbf{d} \rangle, \tag{5}$$

where $\mathbf{p}$ is a point on the axis and $\mathbf{d}$ is the axis direction. The anchor reference a_{ij} specifies which face or axis should be extracted from each endpoint primitive. This design makes each edge geometrically concrete, since relation parameters are evaluated in the coordinate frames of the referenced anchors.

Constraint types. The six constraint types describe which relative degrees of freedom between two primitives are restricted. *Fixed* represents rigid attachment and locks the relative pose between two anchors. *Revolute* aligns two axes and preserves only rotation around the shared axis, as in a hinge. *Prismatic* preserves the contact orientation and normal offset while allowing translation along a sliding direction. *Cylindrical* aligns two axes and removes radial displacement, while allowing both axial translation and rotation around the axis. *Planar-Contact* keeps two faces in contact by constraining the normal direction and normal offset, while leaving in-plane translation and in-plane rotation free. *Alignment* only enforces parallel directions and does not constrain relative position. These definitions correspond to the constrained and free translational and rotational degrees of freedom shown in fig. 3.

Auxiliary constraint prediction. The physical constraint loss is applied during training as an auxiliary regularizer on the map encoder. Before graph readout, a lightweight constraint head decodes edge-conditioned features into predicted anchor states for both endpoints. For face-based constraints, the predicted state is $\hat{a}^F = \langle \hat{\mathbf{p}}, \hat{\mathbf{n}}, \hat{\mathbf{t}}, \hat{\mathbf{b}} \rangle$. For axis-based constraints, the predicted state is $\hat{a}^A = \langle \hat{\mathbf{p}}, \hat{\mathbf{d}} \rangle$. We first enforce validity of the predicted face frames with

$$\begin{aligned} \mathcal{L}_{ortho} = \sum_{k \in \{i,j\}} \Bigg[\sum_{\mathbf{u} \in \{\hat{\mathbf{n}}_k, \hat{\mathbf{t}}_k, \hat{\mathbf{b}}_k\}} & (\|\mathbf{u}\|_2^2 - 1)^2 \\ & + (\hat{\mathbf{n}}_k^\top \hat{\mathbf{t}}_k)^2 + (\hat{\mathbf{t}}_k^\top \hat{\mathbf{b}}_k)^2 + (\hat{\mathbf{b}}_k^\top \hat{\mathbf{n}}_k)^2 \\ & + \|\hat{\mathbf{n}}_k \times \hat{\mathbf{t}}_k - \hat{\mathbf{b}}_k\|_2^2 \Bigg]. \end{aligned} \tag{6}$$

This term encourages the decoded normal, tangent, and bitangent vectors to form a unit, orthogonal local frame.

Relation-specific residuals. Let $\Delta\hat{\mathbf{p}} = \hat{\mathbf{p}}_j - \hat{\mathbf{p}}_i$. For each edge, the relation type determines which residuals are active. Fixed relations lock relative orientation and position:

$$\mathcal{R}_{fixed} = \left\{ \hat{\mathbf{n}}_i^\top \hat{\mathbf{n}}_j + 1, \hat{\mathbf{t}}_i^\top \hat{\mathbf{t}}_j - \cos\phi, \hat{\mathbf{t}}_i^\top \hat{\mathbf{b}}_j + \sin\phi, \Delta\hat{\mathbf{p}}^\top \hat{\mathbf{n}}_i - \delta, \Delta\hat{\mathbf{p}}^\top \hat{\mathbf{t}}_i, \Delta\hat{\mathbf{p}}^\top \hat{\mathbf{b}}_i \right\}. \tag{7}$$

Revolute relations align two axes and constrain their centers to lie on the same hinge line:

$$\mathcal{R}_{rev} = \left\{ \hat{\mathbf{d}}_i \times \hat{\mathbf{d}}_j, \Delta\hat{\mathbf{p}} \times \hat{\mathbf{d}}_i, \Delta\hat{\mathbf{p}}^\top \hat{\mathbf{d}}_i \right\}. \tag{8}$$

Prismatic relations preserve face orientation and normal contact while leaving motion along the sliding direction free:

$$\mathcal{R}_{pris} = \left\{ \hat{\mathbf{n}}_i^\top \hat{\mathbf{n}}_j + 1, \hat{\mathbf{t}}_i^\top \hat{\mathbf{t}}_j - 1, \Delta\hat{\mathbf{p}}^\top \hat{\mathbf{n}}_i - \delta, \Delta\hat{\mathbf{p}}^\top \hat{\mathbf{b}}_i \right\}. \tag{9}$$

Cylindrical relations align two axes and constrain radial displacement while leaving axial translation and twist free:

$$\mathcal{R}_{cyl} = \left\{ \hat{\mathbf{d}}_i \times \hat{\mathbf{d}}_j, \Delta\hat{\mathbf{p}} \times \hat{\mathbf{d}}_i \right\}. \tag{10}$$

Planar-contact relations enforce opposing normals and a fixed normal offset, while in-plane translation and in-plane rotation remain unconstrained:

$$\mathcal{R}_{planar} = \left\{ \hat{\mathbf{n}}_i^\top \hat{\mathbf{n}}_j + 1, \Delta\hat{\mathbf{p}}^\top \hat{\mathbf{n}}_i - \delta \right\}. \tag{11}$$

Alignment relations only constrain directions:

$$\mathcal{R}_{align} = \left\{ \hat{\mathbf{d}}_i \times \hat{\mathbf{d}}_j \right\}. \tag{12}$$

The final physical regularizer sums squared residuals over valid edges:

$$\mathcal{L}_{phys} = \lambda_{ortho} \mathcal{L}_{ortho} + \sum_{(i,j) \in \mathcal{E}} \lambda_{c_{ij}} \sum_{\mathbf{r} \in \mathcal{R}_{c_{ij}}} \|\mathbf{r}\|_2^2. \tag{13}$$

Only the residuals associated with the edge type are evaluated. Free degrees of freedom are intentionally omitted from the residual set, which lets Scene Map encode both hard geometric constraints and allowable relative motion.

A.2 Complexity Discussion

For an instantiated scene-map graph with N nodes and hidden width d, one attention layer costs $\mathcal{O}(N^2 d)$. Over L layers, the total complexity is

$$\mathcal{O}(LN^2 d). \tag{14}$$

In practice, N is determined by the task-specific structural template and remains moderate for the manipulation tasks considered in this work, making the additional structural reasoning cost practical.

A.3 MapPolicy Procedure

The overall training and inference procedure of MapPolicy is summarized in Algorithm 1. The pipeline consists of three stages. First, the Map Constructor is trained to reconstruct object structures from masked point clouds, using point-cloud supervision and structural-parameter regression. Second, the trained Map Constructor is frozen and used to generate Scene Maps from RGB-D observations. The Map Encoder extracts structural features from these maps, which are fused with visual and proprioceptive features for action prediction. During this stage, the policy is optimized jointly with the imitation learning objective and the physical constraint loss. Finally, during inference, MapPolicy continuously constructs Scene Maps from incoming observations and uses the resulting fused features to predict and execute robot actions.

Algorithm 1 MapPolicy training and inference pipeline

Stage 1: Map Constructor Training
Input RGB-D demonstrations with object masks, object poses, and structural targets
Project depth and masks into object-level masked point clouds
Predict structural parameters with the structural parameter estimator
Combine pose and structural parameters to instantiate Scene Map
Optimize the constructor with point-cloud supervision and structural-parameter MSE
Stage 2: Policy Training
Freeze the trained Map Constructor
for each demonstration step **do**
    Construct Scene Map from RGB-D observations
    Encode Scene Map into map features with the Map Encoder
    Extract visual features from RGB observations and robot state features from proprioception
    Fuse map, visual, and robot state features, then predict the action with the policy head
    Optimize action imitation loss together with the physical constraint loss
end for
Inference
while the task is not finished **do**
    Obtain object masks, object poses, and masked point clouds from the current RGB-D observation
    Instantiate Scene Map and encode it into map features
    Fuse map features with visual and robot state features
    Execute the predicted action
end while

A.4 Implementation Details

Map Constructor. The structural parameter estimator operates on object-level masked point clouds. Each object point cloud is sampled to 1024 points and encoded by a PointNet encoder with input channels for point coordinates and normal-related attributes. The PointNet stem applies shared one-dimensional layers with widths 64, 128, and 1024, followed by max pooling over points. The pooled feature is passed through fully connected layers of widths 512 and 256 with batch normalization and ReLU activations, producing a 256-dimensional object feature. A task-specific linear prediction head maps this feature to the structural parameters required by the Scene Map template. The output is split into primitive sizes, primitive positions, and primitive rotations according to the task-specific template dimensions.

Map Encoder. The Map Encoder uses a graph hidden width of 768. For each node, the primitive geometry point set and affordance point set are separately encoded by PointNet-style point-wise MLPs with hidden width 64 and output width 768. The semantic descriptor is a 512-dimensional

object-level feature projected to 768 dimensions. These three node streams are concatenated and fused by a linear layer followed by layer normalization and ReLU, yielding a 768-dimensional node embedding.

For each edge, the relation type is embedded with a 10-entry embedding table of width 768. Continuous relation parameters of dimension 3, anchor descriptors of dimension 24, and relative pose descriptors of dimension 9 are each projected to 768 dimensions. The four edge streams are concatenated and fused by a linear layer followed by layer normalization and ReLU, yielding a 768-dimensional edge embedding.

The graph backbone contains 6 Graphormer-style layers with 24 attention heads and dropout 0.1. Each layer first updates edge embeddings with endpoint node features, projects the updated edge embedding into a head-specific attention bias, and applies biased multi-head self-attention over nodes. The feed-forward block expands the hidden width by a factor of 4, uses GELU activation, and projects back to 768 dimensions. The final graph representation is produced by gated readout: a linear gate scores each node, softmax normalizes node weights, and the weighted sum gives a 768-dimensional map feature.

Policy Backbone. The visual encoder is a frozen DINO v3 image encoder [55] that extracts the visual feature used by the policy. Scene Map text semantics are encoded separately with a frozen CLIP text encoder. RGB observations are resized to the visual encoder input resolution before feature extraction. The map feature has dimension 768. The map feature, visual feature, and robot state feature are separately projected into modality tokens and fused by self-attention. The fused tokens are pooled and passed to an MLP policy head with hidden widths 1024, 512, and 256. The policy head uses SELU activations and orthogonal initialization, with the final layer scaled by 0.01, and outputs a 7-dimensional action.

Training and inference. Training is performed in two stages. The first stage trains the structural parameter estimator with point-cloud supervision and structural-parameter supervision. The predicted parameters and object poses instantiate a Scene Map, from which a scene point cloud is rendered and projected into the camera frame. The point-cloud loss $\mathcal{L}_{pcd}$ compares this projected point cloud with the observed masked point cloud, while $\mathcal{L}_{struct}$ is a mean-squared error between predicted and ground-truth structural parameters. The first-stage objective is $\mathcal{L}_{ctor} = \mathcal{L}_{pcd} + \lambda_{struct}\mathcal{L}_{struct}$. The second stage freezes the trained constructor and optimizes the Map Encoder and Policy Backbone. For each demonstration step, RGB observations, canonical object point clouds, and object poses instantiate a Scene Map, which is encoded into map features and used for action prediction. The policy training objective is $\mathcal{L} = \mathcal{L}_{act} + \lambda_{phys}\mathcal{L}_{phys}$, where $\mathcal{L}_{act}$ supervises the predicted action and $\mathcal{L}_{phys}$ regularizes the learned map features with physical constraints. At inference time, foreground object masks, object poses, and masked point clouds are extracted from RGB-D observations to instantiate the Scene Map. For later frames, tracked masks and tracked object poses provide the object-level evidence, and the same map construction, map encoding, and action prediction pipeline is applied.

A.5 Experimental Protocols

Task suites and metrics. For MetaWorld, we follow the evaluation metric of Lift3D and use task success rate as the main metric. Each task uses a dataset of 30 episodes, split into 25 training episodes and 5 validation episodes. We train each model for 300 epochs, perform rollout evaluation every 25 epochs with 25 test rollouts, and save the checkpoint with the best rollout success rate. Results are reported as mean and standard deviation across five random seeds.

For RLBench, we follow the task selection and evaluation metric of PDFactor. Each task uses a dataset of 100 demonstrations, and task success rate is reported under the benchmark's official success conditions. Each task is evaluated with 25 episodes per seed, and results are reported as mean and standard deviation across five random seeds.

For ManiSkill, we follow the same evaluation protocol as RLBench. Each task uses a dataset of 100 demonstrations, and task success rate is computed with the official success condition of each environment. We evaluate six official tasks from the *Table-Top 2 Finger Gripper Tasks* category. Results are reported as mean and standard deviation across five random seeds.

For real-world experiments, each method is executed for 100 trials on each task, and success rate is computed with the task-level completion criteria used to annotate the real-robot trials.

Training protocol. MapPolicy is trained in two stages. The first stage trains the structural parameter estimator used by the Map Constructor, and the second stage trains the Map Encoder and Policy Backbone with the constructor fixed. For both stages, we use Adam-style optimization with gradient clipping, cosine learning-rate decay, and a 20-epoch warmup. Unless otherwise stated, training runs use 300 epochs and seed 0. The map-constructor stage uses learning rate 4×10^{-4}, weight decay 10^{-4}, and batch size 512. The policy stage uses learning rate 2×10^{-4} and batch size 24 for MetaWorld, learning rate 2×10^{-4} and batch size 128 for RLBench, and learning rate 4×10^{-4}, weight decay 10^{-4}, and batch size 256 for ManiSkill. We set the map loss weight and physical constraint loss weight to 1 unless otherwise specified by an ablation.

Inputs and model selection. RGB observations are resized before being passed to the visual encoder. We use 224×224 images for MetaWorld and RLBench policy training and 128×128 images for ManiSkill policy training. The structural estimator receives object-level point clouds derived from RGB-D observations and masks, as described in sec. 3.3. For simulation experiments, checkpoints are selected by validation success or rollout success on the corresponding benchmark split. For ablations, we keep the policy backbone, optimizer, training schedule, and evaluation protocol fixed, changing only the ablated representation, encoder, fusion strategy, or loss term.

Baselines. For MetaWorld and ManiSkill, diffusion-based and 3D policy baselines are trained and evaluated under the same task suites and success-rate metric whenever reproducible implementations are available. For RLBench, we use the published numbers of established baselines and match their reporting convention of task-level success rates with mean and standard deviation where available. In the real-world experiments, Diffusion Policy and 3D Diffusion Policy are directly deployed as the primary reproducible baselines using the same robot, camera, task definitions, and trial budget as MapPolicy, while π_0 and Lift3D are included as additional reference points for method positioning.

Compute resources. Simulation experiments are run on local GPU servers equipped with NVIDIA RTX 4090 and NVIDIA H200 GPUs. Each training run uses one GPU and approximately 20 CPU cores. On an H200 GPU, a single task-seed run takes approximately 4–8 GPU-hours for MetaWorld, RLBench, and ManiSkill. The main MapPolicy simulation results cover 49 tasks across five random seeds, corresponding to roughly 980–1960 GPU-hours, with an approximate midpoint of 1.5k GPU-hours. Baseline, ablation, and system-breakdown experiments use the same per-run resource allocation, but their total compute was not logged separately. Real-world data collection takes approximately 20 robot-hours, and the reported real-world evaluations take approximately 50 robot-hours in total.

A.6 Additional Representation Ablation

Table 10: Controlled representation ablation on MetaWorld.

Representation	Visual-only	Mask	Keypoints	Object Pose	3D BBox	Scene Map
Mean S.R. (%) ↑	74.8	81.3	84.9	88.2	82.7	**93.2**
Δ vs. Visual-only	–	+6.5	+10.1	+13.4	+7.9	**+18.4**

We isolate the input representation while keeping the policy backbone, demonstrations, optimization, and evaluation protocol fixed. Table 10 shows that Scene Map achieves 93.2% mean success, outperforming object pose by 5.0 points and visual-only input by 18.4 points. This result highlights the benefit of explicit part structure and physical relations.

A.7 Full Per-Task Simulation Results

This section provides the full per-task results summarized in tab. 1.

Table 11: Main results on MetaWorld [71, 33]. We report success rate (%) on the full MetaWorld evaluation suite containing 25 tasks. Results are averaged over **5 random seeds** and shown as mean and standard deviation. Best results are in **bold**, and second best are underlined.

Method	Avg.	Basket ball	Bin Picking	Box Close	Coffee Pull	Coffee Push	Dis assemble	Hammer	Hand Insert	Handle Pull	Handle Pull Side	Lever Pull	Peg Insert Side	Peg Unplug Side
Diffusion Policy [13]	27.3±1.2	85±3.2	15±2.1	30±4.5	34±5.1	67±3.8	43±6.2	15±2.4	9±1.5	27±4.1	23±3.5	49±5.0	34±4.2	74±2.9
3D Diffusion Policy [74]	51.1±1.5	**98±0.8**	34±4.5	42±5.2	87±2.4	**94±1.2**	69±4.7	76±3.1	14±2.8	53±6.0	85±2.3	79±3.4	69±5.1	75±4.0
π_0 [6]	69.5±1.8	88±2.5	45±5.8	70±4.1	92±1.8	85±3.2	65±5.5	80±2.9	40±6.4	88±2.1	82±3.7	85±2.8	50±7.2	85±2.4
Lift3D [27]	78.8±1.4	96±1.5	92±2.0	**92±1.4**	90±2.2	88±2.7	80±3.6	94±1.8	64±5.1	**100±0.0**	**96±1.1**	86±3.2	84±4.0	98±0.9
MapPolicy (Ours)	**93.2±0.9**	96±1.2	**98±1.0**	82±3.5	**100±0.0**	74±6.2	**100±0.0**	**100±0.0**	**100±0.0**	**100±0.0**	**100±0.0**	94±2.1	**94±1.8**	**100±0.0**	**100±0.0**

Method	Avg.	Pick Out of Hole	Pick Place	Pick Place Wall	Push	Push Back	Push Wall	Reach Wall	Shelf Place	Soccer	Stick Pull	Sweep	Sweep Into
Diffusion Policy [13]	27.3±1.2	0±0.0	0±0.0	5±1.2	30±4.8	0±0.0	20±3.5	59±6.1	11±2.4	14±3.1	11±2.9	18±4.5	10±2.2
3D Diffusion Policy [74]	51.1±1.5	14±2.5	12±2.8	35±5.4	51±4.2	0±0.0	49±5.1	68±3.3	17±3.8	18±2.9	27±5.0	96±1.4	15±3.1
π_0 [6]	69.5±1.8	25±4.2	55±6.5	45±5.9	88±2.1	60±7.2	65±4.8	85±2.4	35±5.5	75±3.8	60±6.2	**100±0.0**	90±1.9
Lift3D [27]	78.8±1.4	60±5.8	65±5.1	62±4.9	85±3.0	65±5.4	44±6.2	88±2.1	42±5.9	76±4.2	61±5.8	90±3.2	72±4.1
MapPolicy (Ours)	**93.2±0.9**	**94±1.8**	**82±3.2**	**94±2.0**	**92±1.5**	**84±3.9**	**78±4.5**	**96±1.2**	**94±2.4**	**80±3.5**	**100±0.0**	**100±0.0**	**98±1.0**

Table 12: Main results on RLBench [25]. We evaluate 25 episodes per task on 18 challenging tasks from RLBench and report mean and standard deviation of success rates (%) across 5 random seeds. Best results are in **bold**, and second best are underlined.

Method	Avg. Success	Push Buttons	Slide Block	Sweep to Dustpan	Meat off Grill	Turn Tap	Put in Drawer	Close Jar	Drag Stick	Put in Safe
PerAct [53]	49.4	92.8±3.0	74.0±13.0	52.0±0.0	70.4±2.0	88.0±4.4	51.2±4.7	55.2±4.7	89.6±4.1	84.0±3.6
3D Diffuser Actor [28]	81.4	98.4±2.0	97.6±3.2	84.0±4.4	96.8±1.6	99.2±1.6	96.0±3.6	96.0±2.5	**100.0±0.0**	97.6±2.0
RVT-2* [20]	81.4	**100.0±0.0**	92.0±2.8	**100.0±0.0**	99.0±1.7	99.0±1.7	96.0±2.0	**100.0±0.0**	99.0±1.7	96.0±2.8
PDFactor [59]	87.3	**100.0±0.0**	98.4±2.2	77.6±6.1	96.0±4.0	96.8±1.8	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	96.8±1.8
MapPolicy	**94.8**	**100.0±0.0**	**99.2±1.6**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**100.0±0.0**	**99.2±1.6**

Method	Avg. Success	Place Wine	Screw Bulb	Open Drawer	Stack Blocks	Stack Cups	Put in Cupboard	Insert Peg	Sort Shape	Place Cups
PerAct [53]	49.4	44.8±7.8	17.6±2.0	88.0±5.7	26.4±3.2	2.4±2.0	28.0±4.4	5.6±4.1	16.8±4.7	2.4±3.2
3D Diffuser Actor [28]	81.4	93.6±4.8	82.4±2.0	89.6±4.1	68.3±3.3	47.2±8.5	85.6±4.1	65.6±4.1	44.0±4.4	24.0±7.6
RVT-2* [20]	81.4	95.0±3.3	88.0±4.9	74.0±11.8	80.0±2.8	69.0±5.9	66.0±4.5	40.0±0.0	35.0±7.1	38.0±4.5
PDFactor [59]	87.3	98.4±2.2	92.8±3.3	97.6±3.6	68.8±3.3	89.6±2.2	77.6±5.4	69.6±4.6	48.0±8.0	64.0±5.7
MapPolicy	**94.8**	**100.0±0.0**	**96.0±2.0**	**99.2±1.6**	**88.8±3.3**	**95.2±2.5**	**92.0±3.6**	**85.6±4.1**	**68.0±5.4**	**82.4±4.4**

A.8 Limitation and Future Work

Limitation MapPolicy relies on the quality of Scene Map construction. Errors in object segmentation, pose estimation, depth projection, or structural parameter estimation can propagate to the map encoder and degrade downstream policy execution, especially with reflective surfaces or noisy RGB-D observations. In addition, the current Scene Map uses a predefined primitive vocabulary and a fixed set of physical constraint types, which makes it suitable for rigid and articulated manipulation but less expressive for deformable objects, fluids, ropes, cloth, or objects with unknown topology. Finally, although we evaluate on 49 simulation tasks and five real-world tasks, the experiments still focus on structured manipulation settings, and broader open-world deployment remains future work.

Future Work Future work should improve Scene Map construction under challenging sensing conditions and reduce the dependence on predefined object templates. One direction is to learn more adaptive primitive and constraint vocabularies that can expand beyond the current rigid and articulated settings. Another direction is to combine Scene Map with long-horizon memory and online map refinement, so that the robot can update structural beliefs during execution and handle more open-ended manipulation environments.

A.9 Broader Impact

MapPolicy aims to improve the reliability of robot manipulation under partial observability by making object structure and physical relations explicit. Potential positive impacts include more robust robotic assistance in controlled household, warehouse, manufacturing, and laboratory environments, as well as improved interpretability through the intermediate Scene Map representation. At the same time, stronger manipulation policies may create risks if they are deployed without appropriate safeguards. Errors in perception, Scene Map construction, or policy prediction can lead to unsafe actions, and improved manipulation capability could be misused outside controlled settings. We therefore

Table 13: Main results on ManiSkill [22]. We report success rate (%) on six official tabletop manipulation tasks from the Table-Top 2 Finger Gripper category. Results are averaged over **5 random seeds** and shown as mean and standard deviation. Best results are in **bold**, and second best are underlined.

Method	Avg.	PegInsertion Side-v1	Place Sphere-v1	Plug Charger-v1	PullCube Tool-v1	Stack Cube-v1	Stack Pyramid-v1
Diffusion Policy [13]	$26.8_{\pm 1.5}$	$0_{\pm 0.0}$	$4_{\pm 1.2}$	$36_{\pm 4.5}$	$47_{\pm 5.2}$	$61_{\pm 3.8}$	$13_{\pm 2.4}$
3D Diffusion Policy [74]	$53.8_{\pm 1.2}$	$\underline{55_{\pm 4.1}}$	$62_{\pm 3.5}$	$\underline{58_{\pm 4.8}}$	$48_{\pm 3.9}$	$72_{\pm 2.7}$	$28_{\pm 4.6}$
Octo [17]	$16.8_{\pm 0.9}$	$0_{\pm 0.0}$	$6_{\pm 1.8}$	$12_{\pm 2.5}$	$34_{\pm 4.1}$	$45_{\pm 5.0}$	$4_{\pm 1.1}$
π_0 [6]	$\underline{54.0_{\pm 1.8}}$	$35_{\pm 5.2}$	$\underline{70_{\pm 4.3}}$	$45_{\pm 6.1}$	$\underline{65_{\pm 3.4}}$	$\underline{75_{\pm 2.9}}$	$\underline{34_{\pm 5.5}}$
MapPolicy (Ours)	$\mathbf{73.0_{\pm 1.1}}$	$\mathbf{68_{\pm 3.2}}$	$\mathbf{82_{\pm 2.9}}$	$\mathbf{74_{\pm 3.7}}$	$\mathbf{78_{\pm 2.8}}$	$\mathbf{88_{\pm 1.5}}$	$\mathbf{48_{\pm 4.2}}$

view MapPolicy as suitable for supervised and safety-constrained deployment, with standard robot safeguards such as workspace limits, speed and force limits, emergency stops, runtime monitoring, and human oversight.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are briefly summarized in the contributions. The abstract and introduction independently elaborate on these contributions, which are subsequently validated in the experiments section.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations related to Scene Map construction errors, the predefined primitive and constraint vocabulary, and the current experimental scope in Appendix A.8.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper describes the method architecture in sec. 3, benchmark tasks and metrics in sec. 4, and training, evaluation, model-selection, and baseline protocols in Appendix A.5.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The anonymous supplemental archive `supp_mappolicy.tar` provides the released code, configuration files, Scene Map templates, smoke tests, and scripts for Meta-World data generation, training, and evaluation, with additional RLBench and ManiSkill code paths. Full generated simulator datasets and trained checkpoints are not included, but the supplemental README documents how to regenerate the data and reproduce the main MetaWorld experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies benchmark task suites, metrics, dataset sizes, train-validation splits, rollout protocol, checkpoint selection, optimizer settings, and key hyperparameters in sec. 4 and Appendices A.5 and A.4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main simulation benchmark results report mean and standard deviation of success rates across five random seeds in tab. 11, tab. 12, and tab. 13. Real-world experiments report success rates over 100 physical trials per task because repeated multi-seed robot evaluations are prohibitively costly.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A.5 reports the GPU types, CPU allocation, per-run training time, approximate total GPU-hours for the main simulation experiments, and robot-hours for real-world data collection and evaluation.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. The experiments involve simulation benchmarks and controlled robot manipulation tasks, and do not involve human subjects, private personal data, sensitive attributes, or deployment in safety-critical real-world settings.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix A.9 discusses potential positive impacts in reliable robot manipulation and possible risks from unsafe deployment or perception errors, together with deployment safeguards such as controlled settings and human supervision.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

Justification: All the assets used in the paper is properly credited and are the license and terms of use explicitly mentioned and properly respected.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The released new assets are provided in `supp_mappolicy.tar`, including the anonymous MapPolicy code, Scene Map task templates, configuration files, data generation and reproduction scripts, and five real-world demonstration videos. These assets are documented in the supplemental README, asset documentation, and Appendix description of the supplementary archive.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects. The real-world experiments are controlled robot manipulation evaluations and do not collect data from human participants.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so IRB approval or equivalent review for human-subject research is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.