Local World Models for Articulated Object Manipulation

Thijmen Nijdam[1] Mikel Zhobro[2] Christian Gumbsch[1] Rahaf Aljundi[3] Stratis Gavves[1] Andrii Zadaianchuk[4]
[1]University of Amsterdam [2]University of Tübingen [3]Toyota Motor Europe [4]New Theory

***Abstract*—Articulated objects such as cabinets, drawers, and doors are widespread in human environments, yet learning predictive models that generalize across their diverse geometries and joint configurations remains challenging. Existing world models typically represent the full scene, forcing dynamics prediction and planning to operate in high-dimensional state spaces containing substantial task-irrelevant variation. We propose a local, affordance-centric world model for articulated object manipulation. Rather than modeling the full scene, the representation consists of the isolated moving part, a single affordance point on its surface, and the robot state. The model predicts rigid-body transformations directly in this compact geometric space, where representation, dynamics, and task goals share a common coordinate frame. Combined with sampling-based Model Predictive Control, the learned model opens drawers and doors from closed states without task-specific policy training, achieving up to 41% success on unseen objects. Despite training on interactions from only eight object instances, performance remains similar on seen and unseen objects, suggesting that articulated-object dynamics transfer naturally in the proposed local representation. Scaling analyses further reveal how model capacity and interaction-data scale affect affordance and agent-state prediction. Together, these results highlight local, affordance-centric world models as a promising alternative to scene-level dynamics modeling for articulated manipulation.**

I. Introduction

General-purpose robots operating in homes and industrial environments must interact with a wide variety of articulated objects, including cabinets, drawers, and doors. Generalizing across these objects remains challenging because their geometry and joint configurations vary substantially, even within a single category [18, 9, 25, 35, 8].

World models provide a promising foundation for articulated-object manipulation by supporting planning, policy learning, and synthetic experience generation [21, 12]. Existing approaches typically model the entire scene, whether through global latent representations [3], tokenized observations [1, 14], rendered representations [2, 16], spatial latent grids [45, 44], or scene point clouds [13]. For articulated objects, however, the relevant dynamics are inherently local: contact at a small surface region drives a single rigid part along the motion permitted by its joint. The remainder of the scene introduces task-irrelevant variation, such as changing backgrounds, lighting conditions, occlusions, and surrounding objects, that a scene-level model must either explain or ignore, reducing sample efficiency. This locality is the premise of *object-centric* world models, which decompose scenes into object entities to predict dynamics compositionally [2, 10, 34]; we take this bias to its extreme, isolating the single task-relevant part rather than tracking a full set of object slots.

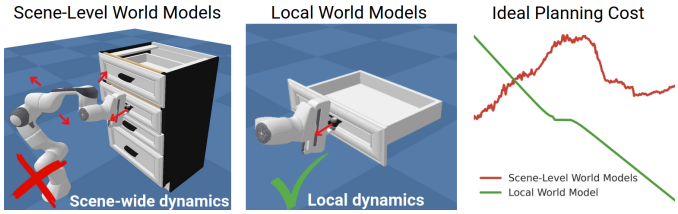

Fig. 1: **Left:** scene-level world models operate on full scenes. **Middle:** our model uses a local representation of the moving part and affordance point. **Right:** task progress along an expert trajectory. Scene-level costs (DINO-WM [45], computed as MSE in DINOv2 [26] feature space) are dominated by task-irrelevant motion and become non-monotonic, while affordance-point distance provides a smooth signal.

Motivated by this observation, we introduce a *local, affordance-centric world model*. Leveraging recent advances in 3D reconstruction [4, 40] and affordance prediction [36], we represent the local world using the isolated moving part, a single affordance point indicating where interaction should occur, and the robot state. The model predicts rigid-body transformations of the moving part directly in this space. The affordance point serves as the unifying abstraction: it specifies where to act, the surrounding geometry determines how the part can move, and the task objective becomes a target location for the same point. Representation, dynamics, and goals therefore share a common geometric space in which distance to the goal provides a direct measure of task progress (Figure 1).

This paper takes a first step toward evaluating this representation. We ask whether a world model learned in this local space is sufficiently accurate for planning and whether the resulting dynamics transfer across unseen object instances. We answer these questions using sampling-based Model Predictive Control (MPC) in simulation. To our knowledge, this is the first demonstration of sampling-based MPC over a learned world model solving the task of opening fully closed articulated objects.

Our contributions are threefold:

- We introduce a local, affordance-centric world model for articulated object manipulation, in which dynamics, interaction location, and task goals share a common geometric representation.
- We demonstrate zero-shot MPC over the learned model on seen and unseen articulated objects, achieving a small seen–unseen performance gap despite training on only eight object instances.
- We provide scaling analyses over model and dataset

size, identifying transfer characteristics and limitations of learning articulated-object dynamics in the proposed representation.

II. Method

A. Local Affordance-Centric Formulation

We represent articulated-object manipulation through a local observation centered on the moving part and its affordance point. Formally, we consider a goal-conditioned discrete-time system with unknown transition dynamics. At each time step t, the agent receives an observation o_t consisting of: (i) a *part point cloud* $\mathcal{P}_t \subseteq \mathbb{R}^3$, a 3D surface reconstruction of the moving part (e.g. a drawer front and its handle, Figure 6); (ii) the current *affordance point* $p_t^{\text{aff}} \in \mathbb{R}^3$, a single point on the part surface marking the contact site (e.g. a handle tip); and (iii) the *proprioceptive state* $q_t = (x_t \in \mathbb{R}^3,\, r_t \in \mathbb{R}^6,\, g_t \in \mathbb{R})$, comprising the end-effector position, orientation in the 6D representation of Zhou et al. [46], and gripper opening. In simulation, the part point cloud (512 points) is sampled from the moving part's mesh using ground-truth part identity, isolating dynamics modeling from perception errors; the affordance point is the end-effector position at first contact in the training demonstrations, while at test time it is sampled randomly from the part surface and propagated through the rollout by the model's predicted transformations. We illustrate this local observation extraction in Figure 2.

The action $a_t \in \mathbb{R}^{10}$ specifies a delta in the same components as q_t: position, orientation, and gripper opening. The agent is given a target affordance location $p^{\text{goal}} \in \mathbb{R}^3$ (e.g. the handle position when a drawer is fully open); an episode succeeds when $\|p_t^{\text{aff}} - p^{\text{goal}}\| \leq \epsilon$.

Given the current observation and a sequence of H actions, the dynamics model $f_\theta(o_t,\, a_{t:t+H-1})$ predicts, for each step $k = 0, \ldots, H-1$, a rigid-body transformation $\Delta T_{t+k} \in SE(3)$ and a proprioceptive update Δq_{t+k}. The transformation is applied to the part point cloud and the affordance point, while Δq_{t+k} updates the proprioceptive state, so each predicted step yields a full observation and a single call produces the predicted trajectory $o_{t+1}, \ldots, o_{t+H}$.

B. Local World Model

The Local World Model (LWM) implements f_θ as a Transformer over $3 + H$ tokens, with prediction horizon $H = 6$. Three observation tokens encode the current observation: the part point cloud is encoded by PointNet [28] into a global feature vector, and the affordance point and proprioceptive state are each embedded by a linear layer; the six action tokens are embedded by a shared linear layer. A prefix-LM attention mask lets the observation tokens attend bidirectionally to each other, while action token k attends to the three observation tokens and causally to action tokens $0, \ldots, k$. After the Transformer blocks, two MLP heads read each action token: a part-motion head outputs ΔT_{t+k}, represented as a 6D continuous rotation [46] concatenated with a 3D translation, and a state head outputs the proprioceptive delta Δq_{t+k}. The architecture is joint-type agnostic: for prismatic joints ΔT_{t+k} reduces to a pure translation, for revolute joints it captures rotation about the joint axis, without explicit joint classification or axis estimation.

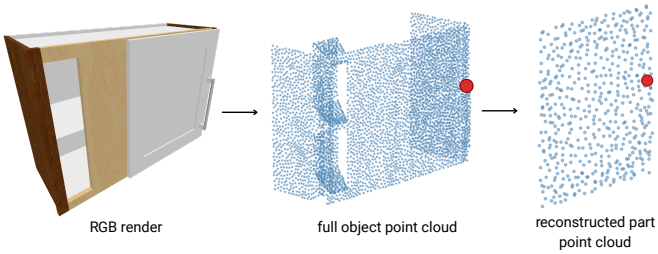

Fig. 2: Local observation extraction. *Left*: RGB render of the object. *Middle*: full object point cloud (4500 points) with affordance point (red). *Right*: reconstructed part point cloud (512 points) with affordance point (red).

Ground-truth rigid deltas are computed between consecutive part point clouds via the Kabsch algorithm, and the model minimizes the mean squared error between predicted and ground-truth deltas, summed over the horizon:

$$\mathcal{L} = \sum_{k=0}^{H-1} \left[\|\Delta \hat{T}_{t+k} - \Delta T_{t+k}\|^2 + \|\Delta \hat{q}_{t+k} - \Delta q_{t+k}\|^2 \right]. \quad (1)$$

The LWM is trained on expert demonstrations from a scripted policy with privileged simulator state [32] ($\sim$56k timesteps), together with diverse non-expert interaction rollouts collected from intermediate checkpoints of a diffusion policy trained on those demonstrations ($\sim$220k timesteps), covering failure modes such as missed grasps and slips. All training data comes from the seen objects only.

C. Model-Predictive Planning

Because goals, state, and dynamics all share the same affordance-centric representation, planning reduces to predicting future affordance locations under candidate action sequences. We therefore use the LWM as the dynamics model in a receding-horizon MPC framework. At each environment step the planner samples K candidate action sequences of length $H_p = 12$, covered by chaining two autoregressive 6-step LWM calls, and evaluates them through the model. Only the first action of the lowest-cost sequence is executed, after which the agent observes the true next state and replans. The cost rewards (i) bringing the end-effector to the affordance point (engagement) and (ii) moving the affordance point toward the goal (task progress):

$$C = \sum_{t=1}^{H_p} \Big[\alpha \, \|p_t^{\text{ee}} - p_t^{\text{aff}}\| + \beta \, \|p_t^{\text{aff}} - p^{\text{goal}}\| \Big], \quad (2)$$

where p_t^{ee} is the (imagined) end-effector position and α, β weight the two objectives. We minimize this cost with iCEM [27].

D. Planning Biases

We inject two known priors into the planning rollout to compensate for the learned model's limitations in these regions.

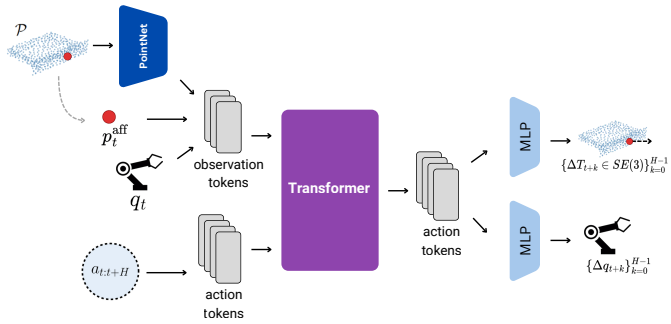

Fig. 3: Local World Model. Three observation tokens (point cloud, proprioceptive state, affordance) form a bidirectional prefix; six action tokens are decoded causally, each attending to the context and all preceding actions. Each action token's output passes through two MLP heads predicting ΔT_{t+k} and Δq_{t+k}.

a) Contact bias.: In out-of-distribution no-contact states, the LWM can predict spurious part motion: it does not reliably distinguish near-contact from actual contact, producing affordance displacement when the end-effector merely approaches the handle without grasping. We therefore gate the predicted transformation on a contact prior:

$$\Delta\tilde{T}_{t+k} = \begin{cases} \Delta T_{t+k} & \text{if contact detected,} \\ I & \text{otherwise,} \end{cases} \tag{3}$$

where I is the identity transformation and contact is detected when the end-effector is within 5 cm of the affordance point with the gripper at least 25% closed.

b) Kinematics bias.: The LWM's agent-state predictions accumulate error over multi-step rollouts, causing the imagined end-effector trajectory to drift from the physically realizable one. Since the robot's kinematic model is known, we replace the predicted proprioceptive delta $\Delta\hat{q}_{t+k}$ with the forward-kinematics result of the commanded action, removing agent-state prediction error from the rollout at the cost of ignoring possible arm–environment collisions.

III. Experiments

We evaluate whether the proposed affordance-centric representation is sufficient to support planning for articulated-object manipulation. Specifically, we ask: (i) can a world model learned in this representation support zero-shot MPC, (ii) do the learned dynamics transfer to unseen object instances, and (iii) how do model and dataset scale affect predictive performance?

A. Setup

We evaluate on three articulated manipulation tasks (Figure 6): opening a drawer (prismatic joint), a side-hinged door (revolute joint, vertical axis), and a bottom-hinged door (revolute joint, horizontal axis). Objects are drawn from PartNet-Mobility [35] and simulated in PyBullet [7], with eight training instances and eight held-out instances (16 in total); unseen-object success is thus a pure measure of dynamics transfer.

TABLE I: Zero-shot MPC planning with progressive inductive biases (SR↑). *Contact*: contact-conditioned identity constraint. *Kinematics*: forward kinematics replaces the predicted agent state. SR@0.4 = partial completion; SR@0.8 = full completion. Each split aggregates 8 object instances × 50 episodes (400 episodes); values are mean ± std of the per-instance success rates.

	Bias		SR@0.4		SR@0.8	
Model	**Cont.**	**Kin.**	**Seen**	**Unseen**	**Seen**	**Unseen**
LWM-100M			$3.5_{\pm 3.0}$	$17.5_{\pm 13.7}$	$1.8_{\pm 2.3}$	$9.5_{\pm 7.3}$
LWM-100M	✓		$47.2_{\pm 13.8}$	$52.2_{\pm 16.6}$	$34.0_{\pm 14.7}$	$38.2_{\pm 19.8}$
LWM-100M	✓	✓	$\mathbf{48.8}_{\pm 14.9}$	$\mathbf{54.5}_{\pm 17.5}$	$\mathbf{38.0}_{\pm 14.5}$	$\mathbf{41.0}_{\pm 17.6}$
LWM-200M			$7.2_{\pm 4.8}$	$13.8_{\pm 9.4}$	$3.8_{\pm 4.9}$	$8.5_{\pm 7.5}$
LWM-200M	✓		$48.0_{\pm 10.0}$	$42.2_{\pm 6.5}$	$33.2_{\pm 11.9}$	$29.2_{\pm 11.4}$
LWM-200M	✓	✓	$47.2_{\pm 17.7}$	$47.8_{\pm 13.9}$	$33.2_{\pm 19.1}$	$34.5_{\pm 16.6}$

We report success rates at two thresholds: SR@0.4 (partial task completion) and SR@0.8 (full completion).

We evaluate MPC with progressively added inductive biases (Table I) on two model sizes, LWM-100M and LWM-200M; architecture and training details are given in section C.

B. Zero-Shot Planning Results

A central result is that the learned dynamics transfer well across objects. Across nearly all configurations, success rates on unseen objects remain close to those on seen objects (Table I), despite training on interactions from only eight object instances. This suggests that the local representation captures articulated-object motion in a form that generalizes across geometries and joint configurations. Performance variation is instead dominated by instance-specific factors such as handle geometry, contact opportunities, and initial robot configuration, reflected in the large per-instance standard deviations.

While transfer is strong, successful planning depends critically on addressing a failure mode of the learned dynamics. Without inductive biases, success rates remain near zero. Inspection of planning rollouts reveals that the model occasionally predicts part motion before contact has been established. MPC exploits these hallucinated affordance displacements, repeatedly selecting trajectories that appear successful under the model but fail in the environment (section D).

The contact bias directly addresses this issue and transforms the tasks from effectively unsolved to solvable. Full-completion success rises to roughly 30–40% across configurations, demonstrating that the affordance-centric representation contains sufficient information for zero-shot planning. The planner discovers approach, grasp, and manipulation behaviors without demonstrations at test time or task-specific policy training.

Replacing predicted agent-state updates with forward kinematics yields a smaller gain: rather than fixing a model failure like the contact bias, it merely reduces rollout error in the imagined robot state.

The remaining failures primarily stem from the planning objective. Because task progress is measured solely through affordance displacement, MPC favors actions that reduce affordance-goal distance as quickly as possible. Consequently,

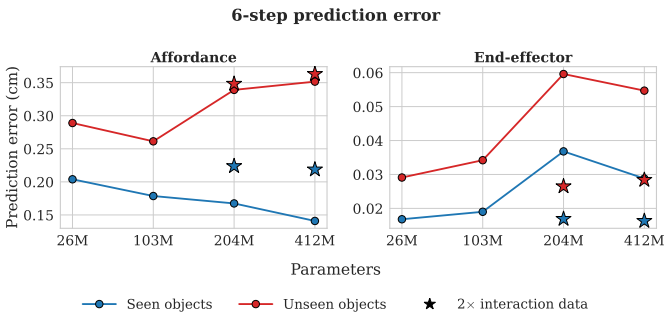

Fig. 4: Model scale: 6-step prediction error across Transformer sizes, averaged over expert and interaction test sets. Stars ($\star$) denote $2\times$ interaction data. Left: affordance error; right: end-effector error.

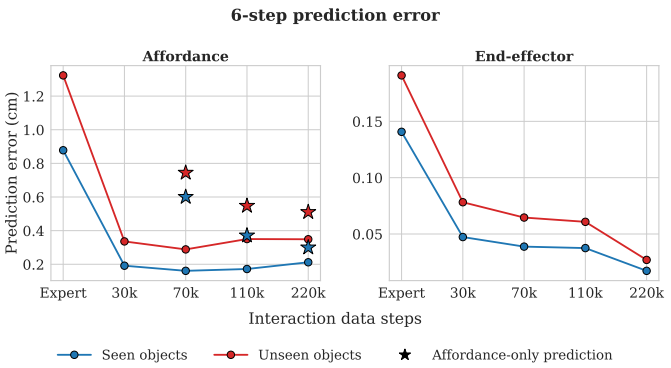

Fig. 5: Dataset scale: effect of interaction-data scale on 6-step prediction error (200M model). "Expert" uses expert demonstrations only; stars ($\star$) show an affordance-only ablation (no end-effector head). Left: affordance error; right: end-effector error.

many successful trajectories rely on incidental contacts or nudging interactions rather than stable grasps. The representation therefore appears sufficient for solving the task, while the current objective remains insufficient for enforcing the desired interaction style.

C. Scaling Ablations

To better understand what the LWM learns, we study how prediction accuracy scales with model size and interaction-data volume. The model jointly predicts two targets over a 6-step horizon: affordance motion and end-effector state.

a) Model scale.: Across four Transformer sizes (26M–412M), the two targets exhibit different scaling behavior (Figure 4). Affordance error on unseen objects is minimized at 100M parameters and increases for larger models, suggesting overfitting to instance-specific contact geometry. End-effector prediction follows a more conventional scaling trend, degrading only for the largest models and recovering when interaction data is doubled ($\star$). Because both targets share a backbone, additional capacity and data appear to shift the representation toward end-effector prediction at the expense of affordance accuracy.

b) Dataset scale.: Fixing the architecture (200M) and varying interaction data from expert-only to 220k steps (Figure 5) shows that expert demonstrations alone are insufficient. Even 30k interaction steps substantially reduce prediction error. Beyond this point, the targets diverge: end-effector error decreases monotonically, whereas affordance error reaches its minimum near 70k interaction steps before increasing. An affordance-only ablation ($\star$) continues improving with more data, indicating that the degradation arises from joint training rather than data scale itself. Nevertheless, jointly predicting end-effector state remains beneficial for affordance prediction across all scales.

IV. Conclusion

We introduced a local, affordance-centric world model for articulated-object manipulation, in which dynamics prediction, interaction location, and task goals share a common geometric representation. As a first study, we showed that this representation supports zero-shot planning with sampling-based MPC, opening doors and drawers from closed states on both seen and unseen objects. The most telling result is not the absolute success rate but the small seen–unseen gap: despite training on interactions from only eight objects, the learned dynamics transfer across geometries and joint configurations, suggesting that articulated-object dynamics may be more naturally modeled in a local, part-centric space than in a full-scene one.

The primary limitations arise from the scale and scope of the study. Training is restricted to eight cabinet-like objects, and the local observations rely on ground-truth part segmentation and affordance annotations from the simulator, leaving open how far the observed transfer extends to more diverse objects and realistic perception noise.

Natural next steps are to scale to larger and more diverse datasets such as ArtVIP [17], integrate learned 3D reconstruction and affordance prediction [4, 40, 36] to remove privileged simulator inputs, and leverage high-quality planning rollouts as additional training data for iterative world-model improvement.

References

[1] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

[2] Leonardo Barcellona, Andrii Zadaianchuk, Davide Allegro, Samuele Papa, Stefano Ghidoni, and Efstratios Gavves. Dream to manipulate: Compositional world models empowering robot imitation learning with imagination. In *The Thirteenth International Conference on Learning Representations*, 2025.

[3] Akshay L Chandra, Iman Nematollahi, Chenguang Huang, Tim Welschehold, Wolfram Burgard, and Abhinav Valada. Diwa: Diffusion policy adaptation with world models. *arXiv preprint arXiv:2508.03645*, 2025.

[4] Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, et al. Sam 3d: 3dfy anything in images. *arXiv preprint arXiv:2511.16624*, 2025.

[5] Yue Chen, Muqing Jiang, Kaifeng Zheng, Jiaqi Liang, Chenrui Tie, Haoran Lu, Ruihai Wu, and Hao Dong. Learning part-aware dense 3d feature field for generalizable articulated object manipulation. *arXiv preprint arXiv:2602.14193*, 2026.

[6] 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. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[7] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning, 2016.

[8] Wenbo Cui, Chengyang Zhao, Songlin Wei, Jiazhao Zhang, Haoran Geng, Yaran Chen, Haoran Li, and He Wang. Gapartmanip: A large-scale part-centric dataset for material-agnostic articulated object manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14791–14798. IEEE, 2025.

[9] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv preprint arXiv:2205.04382*, 2022.

[10] Stefano Ferraro, Pietro Mazzaglia, Tim Verbelen, and Bart Dhoedt. Focus: object-centric world models for robotic manipulation. *Frontiers in Neurorobotics*, 19: 1585386, 2025.

[11] Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. In *Conference on Robot Learning*. PMLR, 2018.

[12] Bohan Hou, Gen Li, Jindou Jia, Tuo An, Xinying Guo, Sicong Leng, Haoran Geng, Yanjie Ze, Tatsuya Harada, Philip Torr, Oier Mees, Marc Pollefeys, Zhuang Liu, Jiajun Wu, Pieter Abbeel, Jitendra Malik, Yilun Du, and Jianfei Yang. World model for robot learning: A comprehensive survey. *arXiv preprint arXiv:2605.00080*, 2026.

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

[14] Yuhang Huang, JIazhao Zhang, Shilong Zou, XInwang Liu, Ruizhen Hu, and Kai Xu. Ladi-wm: A latent diffusion-based world model for predictive manipulation. *arXiv preprint arXiv:2505.11528*, 2025.

[15] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[16] Taoran Jiang, Yixuan Guan, Liqian Ma, Jing Xu, Jiaojiao Meng, Weihang Chen, Zecui Zeng, Lusong Li, Dan Wu, and Rui Chen. DexSim2Real^2: Building explicit world model for precise articulated object dexterous manipulation. *IEEE Transactions on Robotics*, 2025.

[17] Zhao Jin, Zhengping Che, Zhen Zhao, Kun Wu, Yuheng Zhang, Yinuo Zhao, Zehui Liu, Qiang Zhang, Xiaozhu Ju, Jing Tian, et al. Artvip: Articulated digital assets of visual realism, modular interaction, and physical fidelity for robot learning. *arXiv preprint arXiv:2506.04941*, 2025.

[18] Dov Katz and Oliver Brock. Manipulating articulated objects with interactive perception. In *2008 IEEE International Conference on Robotics and Automation*, pages 272–277. IEEE, 2008.

[19] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[20] Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020.

[21] Xinqing Li, Xin He, Le Zhang, Min Wu, Xiaoli Li, and Yun Liu. A comprehensive survey on world models for embodied ai. *arXiv preprint arXiv:2510.16732*, 2025.

[22] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[23] Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk. Temporally consistent object-centric learning by contrasting slots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[24] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kPAM: KeyPoint affordances for category-level

robotic manipulation. In *International Symposium on Robotics Research (ISRR)*, 2019.

[25] Mayank Mittal, David Hoeller, Farbod Farshidian, Marco Hutter, and Animesh Garg. Articulated object interaction in unknown scenes with whole-body mobile manipulation. In *2022 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 1647–1654. IEEE, 2022.

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[27] Cristina Pinneri, Shambhuraj Sawant, Sebastian Blaes, Jan Achterhold, Joerg Stueckler, Michal Rolinek, and Georg Martius. Sample-efficient cross-entropy method for real-time planning. In *Conference on Robot Learning*, pages 1049–1065. PMLR, 2021.

[28] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[29] 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*, 2023.

[30] 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 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[31] Rishi Veerapaneni, John D Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning*. PMLR, 2020.

[32] Yufei Wang, Ziyu Wang, Mino Nakura, Pratik Bhowal, Chia-Liang Kuo, Yi-Ting Chen, Zackory Erickson, and David Held. Articubot: Learning universal articulated object manipulation policy via large scale simulation. *arXiv preprint arXiv:2503.03045*, 2025.

[33] Shijie Wu, Yihang Zhu, Yunao Huang, Kaizhen Zhu, Jiayuan Gu, Jingyi Yu, Ye Shi, and Jingya Wang. Afforddp: Generalizable diffusion policy with transferable affordance. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 6971–6980, 2025.

[34] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. Slotformer: Unsupervised visual dynamics simulation with object-centric models. *arXiv preprint arXiv:2210.05861*, 2022.

[35] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020.

[36] Yifu Yuan, Haiqin Cui, Yaoting Huang, Yibin Chen, Fei Ni, Zibin Dong, Pengyi Li, Yan Zheng, Hongyao Tang, and Jianye Hao. Embodied-r1: Reinforced embodied reasoning for general robotic manipulation. *arXiv preprint arXiv:2508.13998*, 2025.

[37] Andrii Zadaianchuk, Georg Martius, and Fanny Yang. Self-supervised reinforcement learning with independently controllable subgoals. In *Conference on Robot Learning*, pages 384–394. PMLR, 2021.

[38] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Self-supervised visual reinforcement learning with object-centric representations. In *International Conference on Learning Representations*, 2021.

[39] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[40] Andrii Zadaianchuk, Leonardo Barcellona, Lennard Schuenemann, Christian Gumbsch, Zehao Wang, Muhammad Zubair Irshad, Fabien Despinoy, Rahaf Aljundi, Stratis Gavves, and Sergey Zakharov. Reconstruction by generation: 3d multi-object scene reconstruction from sparse observations. *arXiv preprint arXiv:2604.27106*, 2026.

[41] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. *arXiv preprint arXiv:2403.03954*, 2024.

[42] Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Ayzaan Wahid, Vikas Sindhwani, and Johnny Lee. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*. PMLR, 2021.

[43] Weipu Zhang, Adam Jelley, Trevor McInroe, Amos Storkey, and Gang Wang. Object-centric world models from few-shot annotations for sample-efficient reinforcement learning. *arXiv preprint arXiv:2501.16443*, 2025.

[44] Haoyu Zhen, Qiao Sun, Hongxin Zhang, Junyan Li, Siyuan Zhou, Yilun Du, and Chuang Gan. Tesseract: learning 4d embodied world models. *arXiv preprint arXiv:2504.20995*, 2025.

[45] Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

[46] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

APPENDIX A
RELATED WORK

a) Articulated object manipulation.: Recently, imitation learning dominates articulated object manipulation. Diffusion policies [6, 41] have been scaled to a universal policy trained on over 42,000 demonstrations [32], and vision-language-action models add large-scale pretraining and instruction conditioning [19, 15]. By learning appearance, geometry, and strategy jointly, these methods require heavy supervision and degrade on unseen geometries [32, 33, 5]. We instead learn predictive dynamics from action-conditioned transitions and plan with them directly.

b) World models for manipulation.: Recent manipulation world models enable zero-shot goal-conditioned planning via sampling-based MPC [21, 12]. DINO-WM [45] plans by minimizing DINOv2 feature distance to a goal image, V-JEPA 2 [1] scales this recipe with a video-pretrained encoder, and PointWorld [13] predicts per-point scene flow toward a goal point cloud. All define both dynamics and planning over global scene state, so for articulated objects their costs can vary weakly or non-monotonically with task progress, as we show for DINO-WM in Figure 1; we instead learn dynamics in a local, affordance-centric representation. Closest to ours, DexSim2Real^2 [16] plans via MPC over a per-object physics-engine twin built through active interaction, but requires test-time twin-building per object; we learn a single model shared across objects, retaining cross-instance generalization.

c) Object-centric representations for manipulation.: Object-centric learning decomposes a scene into a set of entity representations without supervision, progressing from slots bound to synthetic objects in Slot Attention [22] to real-world images via feature reconstruction in DINOSAUR [29], and to temporally consistent video slots in VideoSAUR [39] and SlotContrast [23]. In control, such structured representations serve as compositional state and dynamics models: SMORL [38] and SRICS [37] treat per-object slots as a goal-conditioned space for self-supervised manipulation, while object-centric and relational world models predict entity-factored dynamics for planning and exploration [20, 31, 34, 10, 43]. A complementary line structures manipulation around object keypoints or dense descriptors rather than learned slots, localizing *where* to act on an object: Dense Object Nets [11], kPAM [24], Transporter Networks [42], and $SE(3)$-equivariant neural descriptor fields [30]. These approaches either maintain a fixed set of latent slots over the whole scene, leaving dynamics to emerge in latent space, or learn descriptors that indicate where to act but not how the contacted object will move. We instead commit to the single task-relevant entity—the moving part and its affordance point—and learn explicit $SE(3)$ rigid-body dynamics for it in a 3D frame shared by representation, dynamics, and goal, so that planning cost reduces to distance to the goal and transfers across articulated instances.

APPENDIX B
TASK OBJECTS

We evaluate on three articulated manipulation tasks drawn from PartNet-Mobility [35]: a drawer (prismatic joint), a side-hinged door (revolute joint, vertical axis), and a bottom-hinged door (revolute joint, horizontal axis), illustrated in Figure 6. Eight seen and eight unseen instances are used per category.

Fig. 6: The three articulated manipulation tasks evaluated in this work. *Left*: drawer (prismatic joint). *Center*: side-hinged door (revolute, vertical axis). *Right*: bottom-hinged door (revolute, horizontal axis).

APPENDIX C
LOCAL WORLD MODEL HYPERPARAMETERS

Table II lists all LWM hyperparameters. The point cloud encoder is a PointNet [28] with a per-point MLP ($3 \to 64 \to 128 \to 256$, ReLU activations, no layer normalization), followed by channel-wise max-pooling and a linear projection to 256 dimensions, then linearly projected to d_{model}. The Transformer backbone uses pre-norm with GELU activation. Each prediction head is a two-layer MLP ($d_{\text{model}} \to d_{\text{model}} \to$ output_dim) with ReLU activation in the hidden layer and no dropout. The two model variants share this architecture and training setup (d_{model}=1024), differing only in depth L (8 vs. 16 layers), learning rate, and dataset size. The deeper LWM-200M requires a lower learning rate for stable training.

TABLE II: LWM hyperparameters.

Parameter	Value
Transformer	
Attention pattern	Prefix-LM[a]
Dimension (d_{model})	1024
Heads (n_{heads})	4
Layers (L)	$\{8, 16\}$[b]
FFN hidden dim	$4 \times d_{\text{model}}$
Dropout	0.1
Input encoders	
Point cloud encoder	PointNet ($256 \to d_{\text{model}}$)
Point cloud points (N)	512
Agent state encoder	Linear ($10 \to d_{\text{model}}$)
Affordance encoder	Linear ($3 \to d_{\text{model}}$)
Action encoder	Linear ($10 \to d_{\text{model}}$)
Training	
Optimizer	AdamW
Learning rate (η)	$\{10^{-4}, 6\text{e-}5\}$[c]
LR schedule	Cosine ($\eta \to 0$)
LR warmup	500 steps (linear)
Weight decay	0.01
Gradient clipping	max norm $= 1.0$
Batch size	1024
Epochs	300
Mixed precision	AMP (float16)
Loss	MSE (z-score norm.)
Train / val split	90% / 10%

[a] Obs. tokens attend bidirectionally; action tokens attend causally to obs. and preceding actions.
[b] L=8 for LWM-100M, L=16 for LWM-200M.
[c] 10^{-4} for LWM-100M, 6×10^{-5} for LWM-200M.

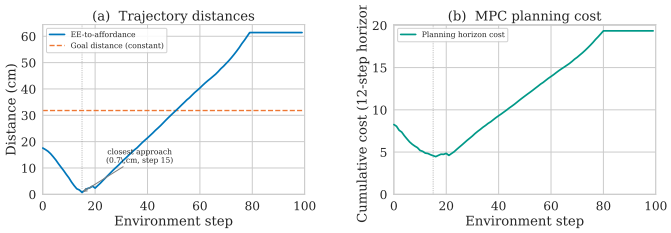

Fig. 7: Failure mode without biases (representative episode, seen drawer, LWM-200M). **(a)** End-effector–to–affordance distance and goal distance: the arm approaches to within 0.7 cm at step 15, then retreats to over 60 cm; the goal distance never changes because contact is never made. **(b)** Cumulative 12-step planning-horizon cost of the selected action sequence over the same episode.

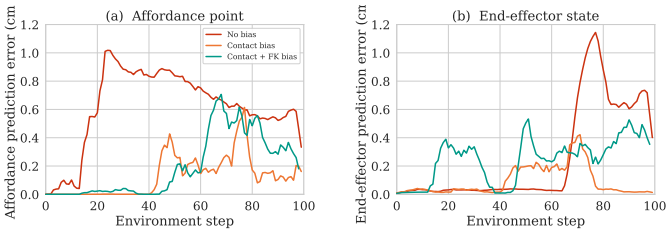

Fig. 8: 1-step prediction error across planner configurations (same episodes as Figure 7; 5-step moving average). **(a)** Affordance point error: without the contact bias, the LWM produces sustained hallucinated displacements; both bias configurations eliminate these during the approach. **(b)** End-effector error: the no-bias planner drives the arm into out-of-distribution states where end-effector prediction also degrades.

Appendix D
Out-of-Distribution Hallucination Analysis

Figure 7 dissects a representative no-bias episode (seen drawer, LWM-200M). The planner first behaves as intended: the arm approaches the handle, reaching within 0.7 cm at step 15. It then retreats steadily to over 60 cm while the goal distance stays constant — no contact ever occurs, so the affordance point never moves. The planning cost (right panel) decreases during the approach and then rises throughout the retreat: the planner keeps selecting these sequences because, in imagination, hallucinated affordance displacements promise cost reductions that never materialize in the environment.

Increasing the engagement weight α by 5–10$\times$ causes the end-effector to stay engaged, as retreating is now heavily penalized. However, this weight is too large: the planner prioritizes actions that stay close to the affordance point rather than actually grasping and retracting to move the part toward the goal.

Figure 8 overlays the 1-step prediction errors for the three planner configurations of Table I. On the affordance target (left),

the no-bias trajectory produces sustained errors of 0.8–1.0 cm from step 15 onward, reflecting the hallucinated displacements that drive the retreat. Both bias configurations stay near zero during the approach and only rise during manipulation, when genuine prediction errors arise from contact dynamics. On the end-effector target (right), the no-bias trajectory shows elevated errors once the arm enters out-of-distribution territory; the contact-bias trajectory keeps errors low throughout by preventing the arm from reaching these states. The contact + kinematics trajectory is the most informative: end-effector errors remain small during free-space motion but rise during active contact, when collision forces cause the realized end-effector state to deviate from the commanded action. The LWM cannot predict these contact-induced deviations, as they depend on collision geometry absent from the training data — the kinematics bias sidesteps this by replacing the learned end-effector predictions with commanded kinematics during planning.