WECO: EQUIVARIANT, GENERALIZABLE ROBOT ACTIONS FROM OBJECT-CENTRIC WORLD CHANGES

Anonymous authors
Paper under double-blind review

ABSTRACT

Generalizing robot manipulation beyond the training distribution remains difficult because demonstrations sparsely cover variations in object appearance, geometry, and workspace configuration. We introduce **WECO**, a world–action framework that separates *what should change in the world* from *how a fixed robot embodiment should realize that change*. Given a predicted visual future, WECO extracts a structured object-centric world change consisting of a coarse rigid transformation $(\mathbf{R}, \mathbf{t})$ and sparse contact or alignment references. A geometry-conditioned inverse dynamics model combines this target change with an object-centered mesh of the current state to predict short action chunks in closed loop. Experiments across multiple manipulation primitives show strong robustness to appearance changes and encouraging transfer across object geometry and workspace positions. However, contact-rich skill composition remains challenging, particularly when geometric and spatial changes must be handled jointly. These preliminary results suggest that structured object-level world changes provide a useful inductive bias for manipulation generalization.

Demonstration videos: Google Drive

1 INTRODUCTION

Robot manipulation data are costly and cover only a small fraction of real-world variation, resulting in policies that generalize poorly and remain sensitive to distribution shifts. **VLAs** map observations and language directly to actions Kim et al.; Huang et al. (2024); Liu et al. (2024); Zhu et al. (2025); Li et al. (2025), but remain sensitive to object, background, viewpoint, spatial, and instruction shifts Wang et al. (2025); Fang et al. (2025); Xiang et al. (2025); Gao et al. (2026). **WAMs** instead expose predicted futures to action generation, either through joint video–action modeling or an explicit IDM Garrido et al. (2024); Hou et al. (2026); Shen et al. (2026); Du et al. (2023); Bharadhwaj et al. (2024); Jang et al. (2025); Cai et al. (2026). However, image-space futures remain sensitive to appearance and viewpoint, while the same visual change may be realized through different contact configurations or robot motions. We argue that an effective world-change interface should satisfy three structural priors:

1. **Rigid-transformation prior.** Over a short manipulation phase, the relevant motion can be approximated by a rigid transform $(\mathbf{R}, \mathbf{t})$ of the manipulated object; for deformables or indirect interactions, we model the motion of a local grasp/contact region or a rigid tool.
2. **Object-centered, scale-normalized geometry prior.** We represent geometry in an object-centered, scale-normalized frame to reduce dependence on absolute workspace position and appearance, improving generalization across color, texture, background, and placement.
3. **Behavior-level spatial consistency prior.** When the object, goal, and geometric references are jointly transformed while preserving their relative configuration, predicted behavior should transform consistently with that spatial change.

We introduce **WECO**, which maps **object-centric world changes** to actions. A high-level world model specifies the intended change of a task-relevant object; WECO lifts it into an object-centered 3D representation and uses a geometry-conditioned inverse dynamics model (IDM) to generate short-horizon actions. WECO executes actions in a receding-horizon loop. At each step, it updates the geometric observation and predicts a short action chunk conditioned on the remaining

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

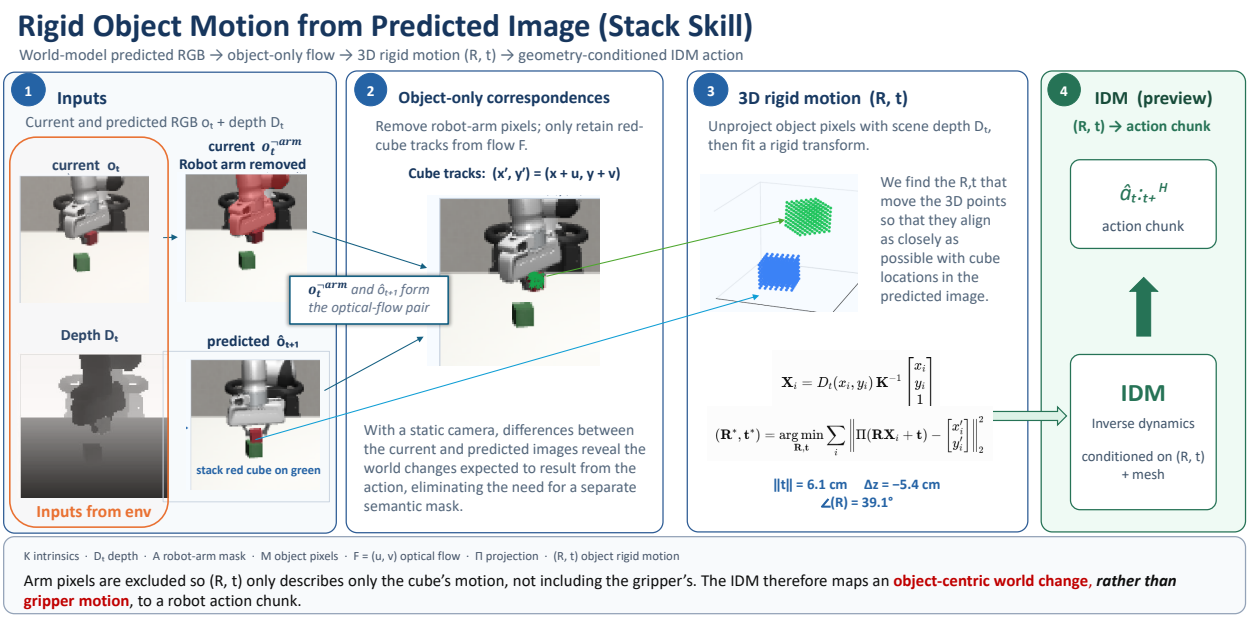

Figure 1: **Object-centric world-change extraction.** Motion between the current observation and predicted visual future is extracted, lifted into 3D, and converted into a coarse rigid transformation and sparse geometric references.

object transformation and the relevant contact or alignment references. This closed-loop formulation enables correction during execution. Experiments show that WECO generalizes to unseen spatial configurations and remains effective under variations in object geometry, scene appearance, and task conditions, supporting object-centric world change as a useful intermediate representation for manipulation under limited interaction data.

Our contributions are: (1) a structured object-centric interface between visual planning and control; (2) a geometry-conditioned, closed-loop IDM combining rigid motion with sparse geometric references; and (3) a preliminary study of appearance, geometry, and spatial generalization, including comparison with VLA and WAM baselines. We next position WECO relative to prior work and then describe the method in detail.

2 Related Work

VLAs map observations and language directly to actions Kim et al.; Huang et al. (2024); Liu et al. (2024); Zhu et al. (2025); Li et al. (2025), but remain sensitive to object, background, viewpoint, spatial, and instruction shifts Wang et al. (2025); Fang et al. (2025); Xiang et al. (2025); Gao et al. (2026). In contrast, **world-action models (WAMs)** expose predicted futures to action generation, typically through (i) joint video–action modeling or (ii) an explicit inverse dynamics module Garrido et al. (2024); Hou et al. (2026); Shen et al. (2026); Du et al. (2023); Bharadhwaj et al. (2024); Jang et al. (2025); Cai et al. (2026).

Implicit inverse dynamics via joint video–action modeling. Several WAMs learn to generate future observations and actions in a coupled model Garrido et al. (2024); Shen et al. (2026); Jang et al. (2025). While this can leverage rich sequential priors, it also inherits an under-constrained inverse problem: the same apparent image-space change may be achievable by multiple contacts and motions, and small prediction errors in the future can translate into large action errors at test time.

Explicit inverse dynamics from visual transitions. A complementary line predicts actions explicitly from two frames or from a planned visual goal, treating inverse dynamics as a mapping from

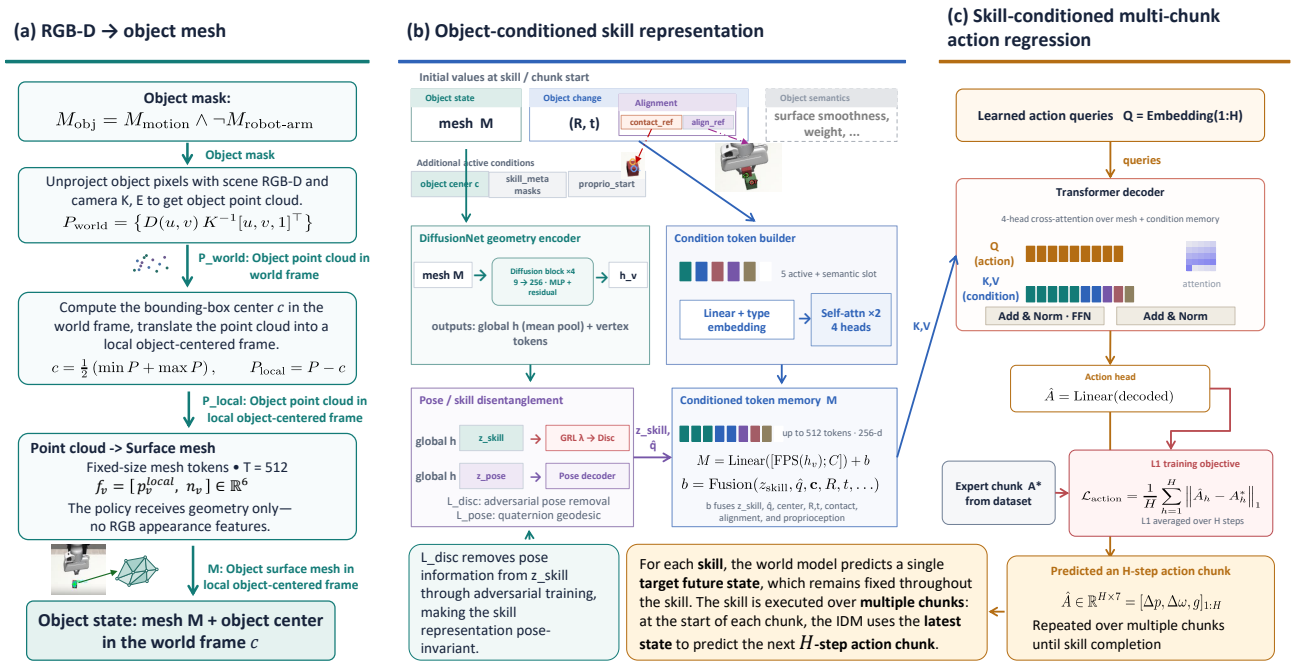

Figure 2: **Geometry-conditioned inverse dynamics model.** Canonicalized object geometry is fused with structured world-change, proprioception, progress, and validity tokens to predict closed-loop action chunks.

observed/predicted transitions to control Du et al. (2023); Bharadhwaj et al. (2024); Cai et al. (2026). These methods simplify planning by delegating control to the IDM, but predicting 3D interaction from 2D appearance can still be brittle under viewpoint and background shifts.

Tool-centric vs. object-centric control interfaces. Video-planning pipelines often extract tool trajectories from predicted videos and execute them (e.g., via tracking and inverse kinematics) Liang et al. (2024). Such tool-centric interfaces can generalize across embodiments when tool motion is the right abstraction, but they primarily model the relationship between *tool motion* and actions rather than the relationship between *object change* and actions. As a result, the same tool motion does not necessarily induce the same object change—e.g., a similar end-effector translation may push an object, slip along its surface, or make no progress depending on contact location, friction, and constraints.

Geometry-grounded and equivariant representations. To improve robustness, prior work encodes correspondences, keypoints, and 3D structure via dense descriptors, transporter-style keypoints, object-centric representations, or 3D-consistent video prediction Florence et al. (2018); Zeng et al. (2021); Chapin et al. (2025); Liu et al. (2026). Separately, spatially equivariant architectures encourage consistent transformation of internal features and outputs under rotations/translations Fuchs et al. (2020); Yang et al. (2025); Gao et al. (2025). However, strict equivariance is not always feasible in manipulation due to reachability and kinematic constraints.

Taken together, these works motivate WECO's design: we represent intended outcomes as structured *object-centric world changes*—more grounded than image-space futures—and condition a closed-loop IDM on object-centered 3D geometry to improve robustness under appearance and spatial shifts.

3 METHOD

WECO couples a *world-change representation* with a geometry-conditioned inverse dynamics model. We first define the object-centric world change $\Delta\mathbf{w}$ that specifies the intended outcome of a skill (Sec. 3.1). We then describe the inverse dynamics model that consumes the current object geometry and $\Delta\mathbf{w}$ to predict closed-loop action chunks (Sec. 3.2), followed by implementation details that make the conditioning pipeline explicit (Sec. 3.3).

3.1 OBJECT-CENTRIC WORLD CHANGE

Motivated by the rigid-transformation prior, WECO represents the outcome of each skill as a structured change to the external world rather than as a full pixel-space future. We define the object-centric world change as

$$\Delta\mathbf{w} = (\mathbf{R}, \mathbf{t}, \mathbf{u}_{\text{contact}}, \mathbf{r}_{\text{align}}), \tag{1}$$

where $(\mathbf{R}, \mathbf{t}) \in \mathrm{SE}(3)$ describes coarse rigid motion, while contact cues and alignment references specify fine-grained interaction and alignment.

The world model can either predict $(\mathbf{R}, \mathbf{t})$ directly or generate a future image from which we recover it geometrically. As shown in Fig. 1, we remove robot-arm pixels, extract object-only optical-flow correspondences, unproject them into 3D using the current depth, and fit $(\mathbf{R}, \mathbf{t})$ by minimizing reprojection error. We exclude the arm and gripper because WECO models how actions change the external world, rather than how actions move the robot itself. The same gripper motion may produce different object outcomes depending on contact and object geometry. Full extraction details are provided in Appendix A.1.

Given this structured target description, the remaining challenge is to map the current object state and $\Delta\mathbf{w}$ into a feasible, closed-loop action chunk.

3.2 GEOMETRY-CONDITIONED INVERSE DYNAMICS MODEL

Our inverse dynamics model (IDM) maps the current object state and object-centric world change $\Delta\mathbf{w}$ to an H-step robot action chunk. As illustrated in Fig. 2, it consists of object-centered mesh encoding, condition fusion, and multi-chunk action regression.

Object-centered mesh encoding. We use the current RGB-D observation and object mask to construct a surface mesh $\mathcal{M}$ with $T = 512$ vertices. The mesh is centered using

$$\mathbf{c} = \frac{1}{2}\left(\min \mathbf{P}^{\text{world}} + \max \mathbf{P}^{\text{world}}\right), \qquad \mathbf{p}_v^{\text{local}} = \mathbf{p}_v^{\text{world}} - \mathbf{c}, \tag{2}$$

and each vertex is represented by geometry-only features $\mathbf{f}_v = [\mathbf{p}_v^{\text{local}}, \mathbf{n}_v] \in \mathbb{R}^6$. Centering gives translated instances of the same object a shared geometric representation. The world-frame center $\mathbf{c}$ is retained separately as a positional bias, providing workspace context while supporting generalization to unseen positions.

Object-conditioned representation. A DiffusionNet encoder Sharp et al. (2022) produces vertex features $\{\mathbf{h}_v\}$ and a global object feature $\mathbf{h}$. The object change $(\mathbf{R}, \mathbf{t})$, center $\mathbf{c}$, contact and alignment references, proprioception, skill metadata, and masks are embedded as typed condition tokens. We additionally disentangle the global feature into a pose-invariant skill representation $\mathbf{z}_{\text{skill}}$ and a pose representation $\mathbf{z}_{\text{pose}}$. The resulting context memory is

$$\mathbf{M}_k = \mathbf{W}_M\,[\mathrm{FPS}(\{\mathbf{h}_v\}); \mathbf{C}_k] + \mathbf{b}_k, \tag{3}$$

where $\mathbf{C}_k$ contains the condition tokens and $\mathbf{b}_k$ provides global skill and workspace conditioning.

Multi-chunk action regression. We use H learned action queries that cross-attend to $\mathbf{M}_k$ through a Transformer decoder Vaswani et al. (2017). A linear head predicts

$$\hat{\mathbf{A}}_k = \mathrm{Linear}\left(\mathrm{Decoder}(\mathbf{Q}, \mathbf{M}_k)\right) \in \mathbb{R}^{H\times 7}, \tag{4}$$

where each action contains Cartesian translation, rotation, and a gripper command. The model is trained with a masked ℓ_1 loss against expert action chunks.

For each skill, the world model predicts one target future state that remains fixed throughout execution. The IDM then operates over multiple chunks: at the beginning of each chunk, it observes the latest state and predicts the next H actions. After execution, the state is updated and the process repeats until skill completion.

3.3 Summary of Implementation Details

World-change extraction. When the world model predicts a future image, we estimate rigid object motion using dense optical flow between the current and predicted images, remove robot-arm flow, back-project surviving pixels with depth, and fit $(\mathbf{R}, \mathbf{t})$ with a RANSAC-PnP objective; we then convert the estimate to the world frame using calibrated extrinsics. Concretely, we keep flow matches in a category-agnostic motion mask,

$$M_t^{\text{obj}} = M_t^{\text{motion}} \wedge \neg M_t^{\text{arm}}, \tag{5}$$

and estimate $(\mathbf{R}, \mathbf{t})$ by minimizing a robust reprojection error over the lifted 3D points (Appendix A.1).

Object-centered geometry. We unproject object pixels into a point cloud $\mathbf{P}^{\text{world}}$, center it by its bounding-box midpoint $\mathbf{c}$, and convert the centered cloud to a surface mesh with a fixed number of vertices (positions and normals; no RGB):

$$\mathbf{c} = \tfrac{1}{2}(\min \mathbf{P}^{\text{world}} + \max \mathbf{P}^{\text{world}}), \qquad \mathbf{P}^{\text{local}} = \mathbf{P}^{\text{world}} - \mathbf{c}. \tag{6}$$

The absolute center $\mathbf{c}$ is provided separately as a positional bias.

Pose–skill disentanglement. To encourage pose-invariant skill features, we split the global mesh embedding into $\mathbf{z}_{\text{skill}}$ and $\mathbf{z}_{\text{pose}}$, train a gradient-reversal discriminator to prevent recovering pose from $\mathbf{z}_{\text{skill}}$, and reconstruct object orientation from $\mathbf{z}_{\text{pose}}$ with a quaternion loss (Appendix A.4).

Condition tokens and training. At chunk k, each conditioning variable $\mathbf{x}_{k,j}$ (e.g., remaining motion, center $\mathbf{c}$, contact/alignment references, and optional proprioception) is projected into a typed token

$$\mathbf{c}_{k,j} = f_j(\mathbf{x}_{k,j}) + \mathbf{e}_j^{\text{type}}, \qquad \mathbf{C}_k = \text{SelfAttn}(\{\mathbf{c}_{k,j}\}). \tag{7}$$

We form the context memory by concatenating fused condition tokens with sampled mesh features,

$$\mathbf{M}_k = \mathbf{W}_M\,[\text{FPS}(\{\mathbf{h}_v\});\, \mathbf{C}_k] + \mathbf{b}_k, \qquad \mathbf{b}_k = \text{Fusion}(\mathbf{z}_{\text{skill}}, \mathbf{c}, \mathbf{R}_k, \mathbf{t}_k, \ldots), \tag{8}$$

where $\mathbf{b}_k$ provides global skill and workspace conditioning. We decode H actions by Transformer cross-attention and train with a validity-masked ℓ_1 action loss plus auxiliary pose/disentanglement losses (Appendix A.5–A.6).

Closed-loop remaining-motion conditioning. The world model predicts a fixed skill target $(\mathbf{R}^\star, \mathbf{t}^\star)$. At the beginning of chunk k, the IDM refreshes state and conditions on the remaining transform to the target. For rigid transport, we measure end-effector progress from the start of the skill and compute

$$\mathbf{R}_k^{\text{rem}} = \mathbf{R}^\star(\hat{\mathbf{R}}_k)^\top, \qquad \mathbf{t}_k^{\text{rem}} = \mathbf{t}^\star - \hat{\mathbf{t}}_k, \tag{9}$$

where $(\hat{\mathbf{R}}_k, \hat{\mathbf{t}}_k)$ denotes the accumulated end-effector motion since skill start.

4 Experiments

We evaluate (i) generalization across appearance, geometry, and workspace shifts, (ii) interpolation to unseen continuous targets, and (iii) long-horizon composition of learned primitives.

4.1 Generalization Tests

Setup. We evaluate five phase-level settings in RoboSuite/MimicGen: Stack, Lift, Move, Rotate, and TowelFold. Unless noted otherwise, each condition contains five rollouts. Stack repeats one demonstrated initialization; the other primitives use five cached demonstrations. Move and Rotate isolate empty-hand end-effector translation and rotation, with the table acting as visual clutter. TowelFold uses a hinged-towel proxy and excludes the cylinder condition.

Condition	Stack	Lift	Move	Rotate	TowelFold
Original	1.00	1.00	1.00	1.00	1.00
Object color	1.00	1.00	1.00	1.00	1.00
Background	1.00	1.00	1.00	1.00	1.00
Texture	1.00	1.00	1.00	1.00	1.00
Size ($0.65\times$)	0.00	0.40	1.00	1.00	1.00
Size ($1.55\times$)	0.00	1.00	1.00	0.60	1.00
Shape (cylinder)	0.00	0.60	1.00	1.00	–
Position	0.00	0.60	1.00	1.00	1.00
Overall valid cells	–	0.83	1.00	0.95	1.00

Table 1: Success rates under visual, geometric, and spatial changes ($N = 5$ per non-missing cell). Stack overall is omitted because its appearance tests retain demonstrated positions.

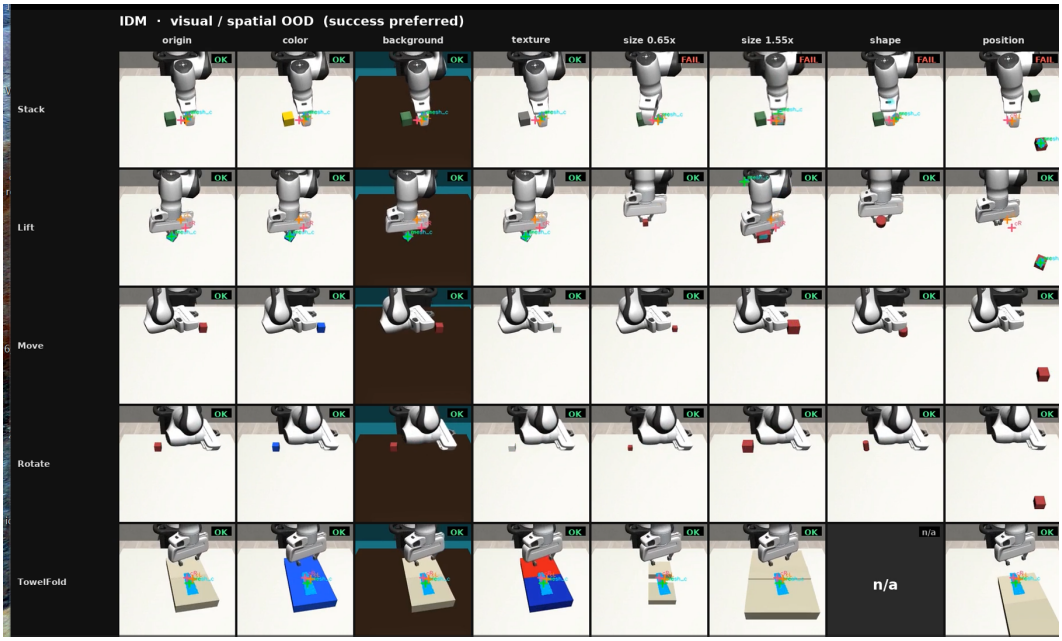

Figure 3: **Representative rollouts across generalization conditions.** Each sequence shows selected frames from one rollout; quantitative success rates are reported in Table 1.

Quantitative shifts. Table 1 reports success under appearance (color/background/texture), geometry (scale/shape), and workspace-position shifts ($N = 5$ per non-missing cell). Move and TowelFold succeed in all tested conditions; Rotate reaches 95.0% and Lift 82.5%. In contrast, Stack is robust to appearance at the demonstrated layout but fails under scale, shape, and position shifts, indicating that contact-rich multi-stage behavior is the current bottleneck. A ± 3 cm XY perturbation also reduces its appearance-condition success to 0%.

Fig. 3 visualizes representative trajectories across appearance, geometry, and spatial shifts.

Novel target generalization. To test whether the learned controller interpolates beyond the discrete demonstrated targets, we evaluate the Move checkpoint on eight novel targets whose minimum distance from the 20 demonstration targets is at least 1.2 cm. The required translations range from 0.070 to 0.095 m, with yaw changes ranging from $-45°$ to $+60°$; the controller succeeds in all eight trials under the relaxed success threshold used for this test.

Long-horizon skill generalization. Finally, we test whether primitives learned for individual tasks can be composed to solve an unseen, longer-horizon task. As shown in Fig. 4, a representative

Method	Original	Color	Background	Shape	Position
WECO ($N = 5$)	1.00	1.00	1.00	0.00	0.00
Seer ($N = 6$)	0.00	0.00	0.00	0.00	0.00
$\pi_{0.5}$ ($N = 6$)	0.17	0.00	0.00	0.17	0.00

Table 2: Stack success rates for WECO and short-finetuned VLA baselines.

Method	Prediction target	Execution	Evaluation	Success
OSC delta (H=96)	7D `delta_actions`	OSC; upsampled to skill length	7 tasks $\times$ 1 rollout	**4/7 (57%)**
EEF start-relative (H=16), Stack overfit	8D `eef_pose`	PD waypoint tracking	Stack $\times$ 1	0/1
EEF start-relative (H=16), low-LR fine-tuning	8D `eef_pose`	PD waypoint tracking	Stack $\times$ 1	0/1
EEF start-relative (H=16), all tasks	8D `eef_pose`	PD waypoint tracking	7 tasks $\times \leq 5$	1/30 (3.3%)
Multi-chunk OSC, Stack $r0$	7D OSC deltas; H=8	closed-loop chunk execution	Stack $\times$ 1	0/1

Table 3: **Action representation and execution ablations.** The OSC delta-action model is currently the best-performing setting. The full-task EEF waypoint setting reaches predicted waypoints but achieves only one successful rollout; its mean normalized progress is 0.726.

trial composes LIFT+MOVE+ROTATE to trace a capital "L," achieving endpoint errors of 0.014 m (foot) and 0.022 m (final). This example is illustrative.

4.2 BASELINES AND DIAGNOSTICS

Having characterized generalization of WECO itself, we next compare against representative VLA/WAM-style baselines and isolate key execution/design choices.

VLA and WAM comparison. Table 2 summarizes Stack outcomes across appearance and geometry shifts for WECO and VLA baselines. We compare Stack with Seer and $\pi_{0.5}$ after 3,000 and 2,000 fine-tuning steps, respectively. Because rollout counts differ, the results are descriptive. WECO handles the demonstrated layout and appearance changes, while geometric and positional changes remain challenging. In diagnostic rollouts, Seer reaches the object in 100%/66.7%/83.3% of original/color/background settings but attains 0% final success; $\pi_{0.5}$ attains 16.7% final success in the original setting and 8.0% final success over a separate 50-rollout evaluation.

Ablations and execution diagnostics. Table 3 compares action parameterizations and execution schemes. Predicting long-horizon operational-space-control (OSC) delta actions and upsampling them to the skill horizon succeeds on $4/7$ tasks (57%) on the designated evaluation instances. In contrast, predicting 16 end-effector waypoints relative to the start pose and tracking them with a PD controller achieves $1/30$ successes (3.3%) across tasks. A single-trial Stack diagnostic suggests the remaining failure is due to learned closed-loop chunk prediction rather than the OSC execution interface.

5 CONCLUSION

WECO proposes object-centric 3D world changes as a structured interface between visual planning and robot control. Across appearance, geometry, and workspace shifts, WECO preserves strong robustness for several isolated primitives, and its closed-loop geometry-conditioned IDM enables continuous-target interpolation and qualitative long-horizon skill composition. At the same time, contact-rich multi-stage behaviors such as Stack remain the dominant failure mode under geometric



Figure 4: **Qualitative task generalization through skill composition.** Selected frames from one trial of an unseen writing task.

and positional shifts. Future work will scale evaluation, strengthen contact-aware conditioning, and study sim-to-real transfer where observation noise and dynamics mismatch further stress the interface.

AI USE DISCLOSURE

We used generative AI tools to help polish the writing (e.g., improving clarity and grammar) and to assist with research ideation (e.g., brainstorming experimental and presentation ideas). All content was reviewed and edited by the authors, who take full responsibility for the paper's claims, results, and conclusions.

REFERENCES

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. *arXiv preprint arXiv:2409.16283*, 2024.

Junhao Cai, Zetao Cai, Jiafei Cao, Yilun Chen, Zeyu He, Lei Jiang, Hang Li, Hengjie Li, Yang Li, Yufei Liu, et al. Internvla-a1: Unifying understanding, generation and action for robotic manipulation. *arXiv preprint arXiv:2601.02456*, 2026.

Alexandre Chapin, Bruno Machado, Emmanuel Dellandrea, and Liming Chen. Object-centric representations improve policy generalization in robot manipulation. *arXiv preprint arXiv:2505.11563*, 2025.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

Irving Fang, Juexiao Zhang, Shengbang Tong, and Chen Feng. From intention to execution: Probing the generalization boundaries of vision-language-action models. *arXiv preprint arXiv:2506.09930*, 2025.

Peter R Florence, Lucas Manuelli, and Russ Tedrake. Dense object nets: Learning dense visual object descriptors by and for robotic manipulation. *arXiv preprint arXiv:1806.08756*, 2018.

Fabian B. Fuchs, Daniel E. Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Chongkai Gao, Zhengrong Xue, Shuying Deng, Tianhai Liang, Siqi Yang, Lin Shao, and Huazhe Xu. RiEMann: Near real-time SE(3)-equivariant robot manipulation without point cloud segmentation. In *Proceedings of the Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2164–2182, 2025.

Jensen Gao, Suneel Belkhale, Sudeep Dasari, Ashwin Balakrishna, Dhruv Shah, and Dorsa Sadigh. A taxonomy for evaluating generalist robot manipulation policies. *IEEE Robotics and Automation Letters*, 11(3):3182–3189, 2026. doi: 10.1109/LRA.2026.3656785.

Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. Learning and leveraging world models in visual representation learning, 2024. URL https://arxiv.org/abs/2403.00504.

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.

Huang Huang, Fangchen Liu, Letian Fu, Tingfan Wu, Mustafa Mukadam, Jitendra Malik, Ken Goldberg, and Pieter Abbeel. Early fusion helps vision language action models generalize better. 2024.

Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Johan Bjorck, Yu Fang, Fengyuan Hu, Spencer Huang, Kaushil Kundalia, Yen-Chen Lin, et al. Dreamgen: Unlocking generalization in robot learning through video world models. *arXiv preprint arXiv:2505.12705*, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, et al. Openvla: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*.

Puhao Li, Yingying Wu, Ziheng Xi, Wanlin Li, Yuzhe Huang, Zhiyuan Zhang, Yinghan Chen, Jianan Wang, Song-Chun Zhu, Tengyu Liu, et al. Controlvla: Few-shot object-centric adaptation for pre-trained vision-language-action models. *arXiv preprint arXiv:2506.16211*, 2025.

Junbang Liang, Ruoshi Liu, Ege Ozguroglu, Sruthi Sudhakar, Achal Dave, Pavel Tokmakov, Shuran Song, and Carl Vondrick. Dreamitate: Real-world visuomotor policy learning via video generation. *arXiv preprint arXiv:2406.16862*, 2024.

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

Zeyi Liu, Shuang Li, Eric Cousineau, Siyuan Feng, Benjamin Burchfiel, and Shuran Song. Geometry-aware 4d video generation for robot manipulation, 2026. URL https://arxiv.org/abs/2507.01099.

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov. Diffusionnet: Discretization agnostic learning on surfaces. *ACM Transactions on Graphics*, 41(3):1–16, 2022. doi: 10.1145/3528223.3530178.

Qiuhong Shen, Shihua Zhang, Yue Liao, Qi Li, Zhenxiong Tan, Shizun Wang, Shuicheng Yan, and Xinchao Wang. World action models: A survey, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Zhijie Wang, Zhehua Zhou, Jiayang Song, Yuheng Huang, Zhan Shu, and Lei Ma. VLATest: Testing and evaluating vision-language-action models for robotic manipulation. *Proceedings of the ACM on Software Engineering*, 2(FSE), July 2025. doi: 10.1145/3729343.

Tian-Yu Xiang, Ao-Qun Jin, Xiao-Hu Zhou, Mei-Jiang Gui, Xiao-Liang Xie, Shi-Qi Liu, Shuang-Yi Wang, Sheng-Bin Duan, Fu-Chao Xie, Wen-Kai Wang, Si-Cheng Wang, Ling-Yun Li, Tu Tian, and Zeng-Guang Hou. Parallels between vla model post-training and human motor learning: Progress, challenges, and trends. *arXiv preprint arXiv:2506.20966*, 2025.

Jingyun Yang, Ziang Cao, Congyue Deng, Rika Antonova, Shuran Song, and Jeannette Bohg. EquiBot: SIM(3)-equivariant diffusion policy for generalizable and data efficient learning. In *Proceedings of the Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 1048–1068, 2025.

Andy Zeng, Pete Florence, Jonathan Tompson, Stefan Welker, Jonathan Chien, Maria Attarian, Travis Armstrong, Ivan Krasin, Dan Duong, Vikas Sindhwani, et al. Transporter networks: Rearranging the visual world for robotic manipulation. In *Conference on Robot Learning*, pp. 726–747. PMLR, 2021.

Jianwei Zhu, Xueying Sun, Qiang Zhang, and Mingmin Liu. Vla-grasp: a vision-language-action modeling with cross-modality fusion for task-oriented grasping. *Complex & Intelligent Systems*, 11(6):272, 2025.

A Additional Method Details

A.1 Object-Centric World-Change Extraction

Two prediction routes. For approximately rigid object motion, WECO represents the skill-level target by an $\mathrm{SE}(3)$ transformation $(\mathbf{R}^\star, \mathbf{t}^\star)$. The world model may either predict this transformation directly or generate a future image from which it is recovered geometrically. We describe the latter route below.

Motion isolation and 3D lifting. Given the current image I_t and the predicted future image $\hat{I}_{t'}$, dense optical flow provides

$$\mathbf{F}_t(u, v) = (\Delta u, \Delta v). \tag{10}$$

We remove motion associated with the robot arm:

$$M_t^{\mathrm{obj}} = M_t^{\mathrm{motion}} \wedge \neg M_t^{\mathrm{arm}}, \qquad \mathbf{p}_i' = \mathbf{p}_i + \mathbf{F}_t(\mathbf{p}_i). \tag{11}$$

With a static camera, the remaining image changes are expected to primarily reflect changes caused by the action. Thus, the motion mask provides a category-agnostic cue without requiring a separate semantic object mask.

Using the current depth D_t and camera intrinsics $\mathbf{K}$, each source pixel is back-projected into the camera frame:

$$\mathbf{X}_i^C = D_t(\mathbf{p}_i)\mathbf{K}^{-1} \begin{bmatrix} u_i & v_i & 1 \end{bmatrix}^\top. \tag{12}$$

Rigid-motion fitting. We use PnP with RANSAC to estimate the transformation that moves the 3D points toward their matched locations in the predicted image:

$$(\hat{\mathbf{R}}^C, \hat{\mathbf{t}}^C) = \arg\min_{\mathbf{R},\mathbf{t}} \sum_i \rho\Big(\big\| \Pi\big(\mathbf{K}(\mathbf{R}\mathbf{X}_i^C + \mathbf{t})\big) - \mathbf{p}_i' \big\|_2^2 \Big), \tag{13}$$

where $\Pi(\cdot)$ denotes perspective projection and ρ represents robust outlier rejection. The estimated transformation is converted into the world frame using the calibrated camera extrinsics:

$$\Delta\hat{\mathbf{T}}^W = \mathbf{T}_{WC}\Delta\hat{\mathbf{T}}^C\mathbf{T}_{CW}. \tag{14}$$

We deliberately exclude robot-arm and gripper pixels because WECO models the relationship between actions and changes in the external world, rather than the direct relationship between actions and robot motion. The same apparent gripper movement may produce different object outcomes—or no object movement—depending on contact location, object geometry, grasp stability, and slippage.

Rigid-grasp prior. Once a stable rigid grasp is established, the object-to-gripper transformation $\mathbf{T}_{GO}$ remains approximately fixed:

$$\mathbf{T}_{WO} = \mathbf{T}_{WG}\mathbf{T}_{GO}. \tag{15}$$

Therefore,

$$\begin{aligned} \Delta\mathbf{T}_O &= \mathbf{T}'_{WO}\mathbf{T}_{WO}^{-1} && (16)\\ &= \mathbf{T}'_{WG}\mathbf{T}_{GO}\left(\mathbf{T}_{WG}\mathbf{T}_{GO}\right)^{-1} && (17)\\ &= \mathbf{T}'_{WG}\mathbf{T}_{WG}^{-1} = \Delta\mathbf{T}_G. && (18) \end{aligned}$$

Thus, after a rigid grasp, the object transformation provides a coarse constraint on the required gripper transformation. This equality is not assumed before contact or under grasp slippage.

Contact and alignment references. The coarse transformation $(\mathbf{R}, \mathbf{t})$ may not uniquely determine contact-sensitive actions. Let $\mathbf{r}$ denote the offset from the gripper origin to a task-relevant reference point. Under a small gripper motion,

$$\Delta\mathbf{p}_r \approx \Delta\mathbf{p}_g + \Delta\boldsymbol{\theta} \times \mathbf{r}, \tag{19}$$

or equivalently,

$$\Delta\mathbf{p}_g \approx \Delta\mathbf{p}_r - \Delta\boldsymbol{\theta} \times \mathbf{r}. \tag{20}$$

Therefore, $(\mathbf{R}, \mathbf{t})$ specifies the coarse object-motion trend, while contact cues and alignment references constrain where the gripper should interact and which local object point should reach the target.

Together, these variables define the object-centric world change:

$$\Delta \mathbf{w} = (\mathbf{R}, \mathbf{t}, \mathbf{u}_{\text{contact}}, \mathbf{r}_{\text{source}}, \mathbf{r}_{\text{target}}) . \tag{21}$$

A.2 Closed-Loop Chunk Conditions

The world model predicts one target future state per skill, and this target remains fixed throughout the skill. The IDM executes the skill over multiple action chunks, refreshing its current-state inputs at the beginning of every chunk.

For rigid transport, we measure progress from the end-effector pose at the beginning of the skill:

$$\hat{\mathbf{R}}_t = \mathbf{R}_t^{ee} (\mathbf{R}_0^{ee})^\top, \qquad \hat{\mathbf{t}}_t = \mathbf{p}_t^{ee} - \mathbf{p}_0^{ee}. \tag{22}$$

The remaining motion condition is

$$\mathbf{R}_t^{\text{rem}} = \mathbf{R}^\star \hat{\mathbf{R}}_t^\top, \qquad \mathbf{t}_t^{\text{rem}} = \mathbf{t}^\star - \hat{\mathbf{t}}_t. \tag{23}$$

Here, rotation and translation progress are represented separately, with translation measured in the world frame.

The chunk-level motion condition depends on the skill:

$$(\mathbf{R}_k, \mathbf{t}_k) = \begin{cases} (\mathbf{I}, \mathbf{0}), & \text{approach or grasp,} \\ (\mathbf{R}_k^{\text{rem}}, \mathbf{t}_k^{\text{rem}}), & \text{rigid transport or rotation,} \\ (\mathbf{R}^\star, \mathbf{t}^\star), & \text{placement or refinement.} \end{cases} \tag{24}$$

For placement and refinement, the target alignment reference remains fixed, while the source reference is refreshed from the current state at each chunk.

A.3 Object-Centered Mesh Representation

From the current RGB-D observation, we unproject the object pixels into a world-frame point cloud $\mathbf{P}^{\text{world}}$. Its bounding-box center is

$$\mathbf{c} = \frac{1}{2} \left(\min \mathbf{P}^{\text{world}} + \max \mathbf{P}^{\text{world}} \right) , \tag{25}$$

and the local object-centered coordinates are

$$\mathbf{P}^{\text{local}} = \mathbf{P}^{\text{world}} - \mathbf{c}. \tag{26}$$

The centered point cloud is converted into a surface mesh $\mathcal{M}$ and normalized to $T = 512$ vertices using farthest-point or surface sampling. Each vertex is represented by

$$\mathbf{f}_v = \left[\mathbf{p}_v^{\text{local}}, \mathbf{n}_v \right] \in \mathbb{R}^6. \tag{27}$$

We omit RGB features to reduce dependence on object appearance.

Centering removes absolute position from the geometry, so translated instances of the same object share the same local representation. The world-frame center $\mathbf{c}$ is retained separately as a positional bias, allowing the model to preserve workspace context without encoding absolute position directly into the mesh.

A DiffusionNet encoder produces vertex-level and global features:

$$\{\mathbf{h}_v\}, \mathbf{h} = \text{DiffusionNet} \left(\mathcal{M}, \{\mathbf{f}_v\} \right) , \qquad \mathbf{h} = \text{MeanPool}_v(\mathbf{h}_v). \tag{28}$$

A.4 POSE–SKILL DISENTANGLEMENT

We project the global mesh feature into separate skill and pose representations:

$$\mathbf{z}_{\text{skill}} = f_{\text{skill}}(\mathbf{h}), \qquad \mathbf{z}_{\text{pose}} = f_{\text{pose}}(\mathbf{h}). \tag{29}$$

A gradient-reversal discriminator attempts to infer pose from $\mathbf{z}_{\text{skill}}$. Through gradient reversal, the mesh encoder is trained to remove pose information from the skill representation. In parallel, a pose decoder reconstructs the object quaternion from $\mathbf{z}_{\text{pose}}$ using

$$\mathcal{L}_{\text{pose}} = 2 \arccos\left(|\langle \hat{\mathbf{q}}, \mathbf{q}^{\star} \rangle|\right). \tag{30}$$

This encourages $\mathbf{z}_{\text{skill}}$ to encode pose-invariant skill information while $\mathbf{z}_{\text{pose}}$ preserves object orientation.

A.5 CONDITION TOKENS AND CONTEXT MEMORY

At chunk k, each conditioning variable is independently projected and augmented with a type embedding:

$$\mathbf{c}_{k,j} = f_j(\mathbf{x}_{k,j}) + \mathbf{e}_j^{\text{type}}, \tag{31}$$

where

$$\mathbf{x}_{k,j} \in \left\{ \mathbf{R}_k, \mathbf{t}_k, \mathbf{c}, \mathbf{u}_{\text{contact}}, \mathbf{r}_{\text{source},k}, \mathbf{r}_{\text{target}}, \mathbf{s}_k^{\text{proprio}}, \mathbf{s}^{\text{skill}}, \mathbf{m}_k \right\}. \tag{32}$$

The condition tokens are fused using lightweight self-attention:

$$\mathbf{C}_k = \text{SelfAttn}\left(\{\mathbf{c}_{k,j}\}\right). \tag{33}$$

A semantic token is reserved but unused in the current implementation.

We combine the typed condition tokens with sampled mesh features:

$$\mathbf{M}_k = \mathbf{W}_M \left[\text{FPS}(\{\mathbf{h}_v\}); \mathbf{C}_k\right] + \mathbf{b}_k, \tag{34}$$

where the global conditioning bias is

$$\mathbf{b}_k = \text{Fusion}\left(\mathbf{z}_{\text{skill}}, \hat{\mathbf{q}}, \mathbf{c}, \mathbf{R}_k, \mathbf{t}_k, \mathbf{u}_{\text{contact}}, \mathbf{r}_{\text{source},k}, \mathbf{r}_{\text{target}}, \mathbf{s}_k^{\text{proprio}}\right). \tag{35}$$

A.6 ACTION DECODING AND TRAINING

We introduce one learned query for each step in the action horizon:

$$\mathbf{Q} = \text{Embed}(1{:}H). \tag{36}$$

The learned queries cross-attend to the mesh and condition memory:

$$\mathbf{D}_k = \text{Decoder}\left(\mathbf{Q}, \mathbf{M}_k\right). \tag{37}$$

A linear action head predicts

$$\hat{\mathbf{A}}_k = \text{Linear}(\mathbf{D}_k) \in \mathbb{R}^{H \times 7}, \tag{38}$$

with

$$\hat{\mathbf{a}}_{k,h} = \left[\Delta\mathbf{p}, \Delta\boldsymbol{\omega}, g\right]_{k,h}. \tag{39}$$

The model is trained against expert action chunks using a validity-masked ℓ_1 objective:

$$\mathcal{L}_{\text{action}} = \frac{\sum_{h=1}^{H} m_{k,h} \left\| \hat{\mathbf{a}}_{k,h} - \mathbf{a}_{k,h}^{\star} \right\|_1}{\sum_{h=1}^{H} m_{k,h}}, \tag{40}$$

where $m_{k,h}$ excludes padded steps in the final chunk. The full training objective is

$$\mathcal{L} = \mathcal{L}_{\text{action}} + \lambda_{\text{pose}} \mathcal{L}_{\text{pose}} + \lambda_{\text{disc}} \mathcal{L}_{\text{disc}}. \tag{41}$$

At execution time, the IDM predicts one H-step action chunk at a time. After the chunk is executed, the current observation, proprioception, source reference, and remaining motion are refreshed before predicting the next chunk. This closed-loop process repeats until the skill is completed.