FOREACT3D: POLICY-GROUNDED FUTURE WORLD MODELING FOR VLA POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Robots need to anticipate how their actions will change the world, since manipulation success hinges on the resulting contacts and object motions. However, existing Vision-Language-Action (VLA) policies that predict future observations from shared features leave the forecast decoupled from the actions the policy will actually execute, and impose no physical constraints on how the scene may evolve. We introduce ForeAct3D, a framework for policy-grounded future world modeling within VLA policies. Learnable geometric queries decode depth, semantic segmentation, and camera pose from the policy representation into current and future semantic 3D scene states, and the future queries are conditioned on the policy-generated action chunk to ground the forecast in the planned interaction. A physical-consistency closure relates the two states through background staticity and instance-level rigidity, and anchors the wrist-camera pose to end-effector kinematics. These objectives shape the shared representation used for action generation during training, and no future prediction is required at inference. Without robot pretraining, ForeAct3D achieves 98.3% average success on LIBERO and an average task length of 3.73 on CALVIN, outperforming its base policy on every suite. Ablations show that semantic 3D supervision, physical consistency, and action conditioning each improve manipulation performance, and that action conditioning substantially improves future object localization. Real-world experiments on spatial placement, object insertion, and sequential manipulation further raise average success from 6.7% to 37.8% over the base policy.

1 INTRODUCTION

Future world modeling aims to predict how a physical scene will evolve under robot actions. This capability matters for manipulation because the same intended action can succeed or fail depending on contact: a push may slide an object toward its target or tip it over, and an insertion may seat or jam. Anticipating such outcomes requires reasoning about both the current observation and the planned action, since the resulting transitions involve contact, changing occlusions, and object motion (Zhen et al., 2024; Wang et al., 2026a; Huang et al., 2026). Robots therefore need a mechanism that represents the physical world, models its transitions under robot actions, and predicts the resulting future states subject to physical constraints (LeCun et al., 2022; Huang et al., 2026).

Existing approaches to future world modeling for robotic manipulation broadly follow two directions. The first models future observations with action-conditioned video generation (Zhu et al., 2025a; Cen et al., 2025; Zhu et al., 2025b). Such models capture how appearance evolves under actions, but represent the future in image space, leaving the underlying 3D structure and physical constraints implicit. The second extends Vision-Language-Action (VLA) models (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024) with auxiliary objectives that predict future observations or features from the policy representation (Zhang et al., 2026a; Zheng et al., 2025; Sun et al., 2026; Qian et al., 2026). We adopt the VLA setting because future-state supervision shapes the representation used for action generation while retaining direct action inference without test-time rollouts.

Achieving physically consistent future world modeling within a VLA policy poses three challenges. First, *constructing a task-relevant state* requires recovering 3D geometry from policy features and separating objects, the robot, and the background, so that the predicted state exposes the spatial configuration relevant to manipulation. Second, *constraining state transitions* requires current and

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.

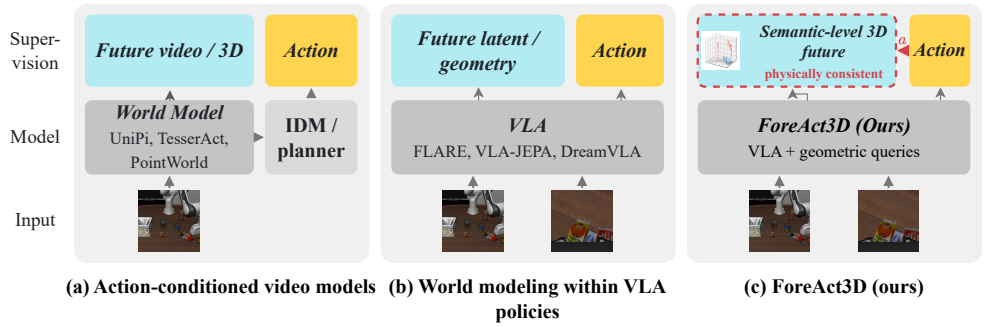


Figure 1: **Future world modeling for robotic manipulation.** (a) Predictive models guide control via inverse dynamics or planning. (b) VLA policies learn future representations jointly with actions. (c) ForeAct3D predicts physically constrained semantic 3D futures conditioned on the policy action. Dashed arrows denote supervision linking the predicted scene states to policy learning.

future predictions to jointly satisfy physical constraints: viewpoint changes must be reconciled with a static environment, and moving objects must preserve rigid geometry. Predicting each state independently does not enforce these relationships. Third, *grounding the future in action* requires predicting the scene changes induced by the robot's chosen action, since the same scene can evolve differently under different actions. Addressing these challenges calls for a structured state, physical constraints across time, and explicit conditioning on the policy's planned interaction.

To close this gap, we introduce **ForeAct3D**, an action-conditioned semantic 3D world modeling framework integrated into VLA policies. As illustrated in Figure 1, ForeAct3D constructs current and future scene states from the representation used for action generation. Learnable geometric queries decode depth, three-class semantic masks, and camera pose, jointly capturing scene geometry and its partition into object, robot, and background regions. During training, a physical-consistency closure relates the two states through background staticity, instance-level rigidity, and end-effector kinematics. Future queries additionally receive the policy-generated action chunk, grounding the predicted transition in the planned interaction. Together, these components train the shared VLA representation to encode a semantic 3D scene and its physically constrained evolution over the action horizon, while retaining direct action inference during deployment.

We evaluate ForeAct3D on LIBERO (Liu et al., 2023) and CALVIN (Mees et al., 2022), achieving 98.3% average success and an average task length of 3.73 without robot pretraining. We further assess the predicted states through geometry, segmentation, and localization metrics, and examine the contributions of structured 3D prediction, physical consistency, and action conditioning through ablations. Real-world experiments on spatial placement, insertion, and sequential manipulation tasks confirm these gains on physical hardware. Our contributions are summarized as follows:

- We introduce ForeAct3D, a framework for policy-grounded future world modeling within VLA policies that reconstructs current semantic 3D states and predicts future states conditioned on policy-generated actions.
- We formulate physical-consistency regularization that relates current and future geometry through background staticity and instance rigidity, while grounding wrist-camera pose in end-effector kinematics.
- We demonstrate improved policy performance on LIBERO, CALVIN, and real-world tasks, with component ablations assessing policy gains and prediction ablations showing the benefit of action conditioning for future object localization.

2 Related Work

Action-Conditioned Video Models. Action-conditioned video models predict future visual observations from the current scene and robot actions (Finn et al., 2016; Finn & Levine, 2017; Yang et al., 2023; Zhu et al., 2025a; Cen et al., 2025). Approaches include autoregressive prediction over visual and action tokens (Wu et al., 2024b; Wang et al., 2025) and diffusion-based generation with

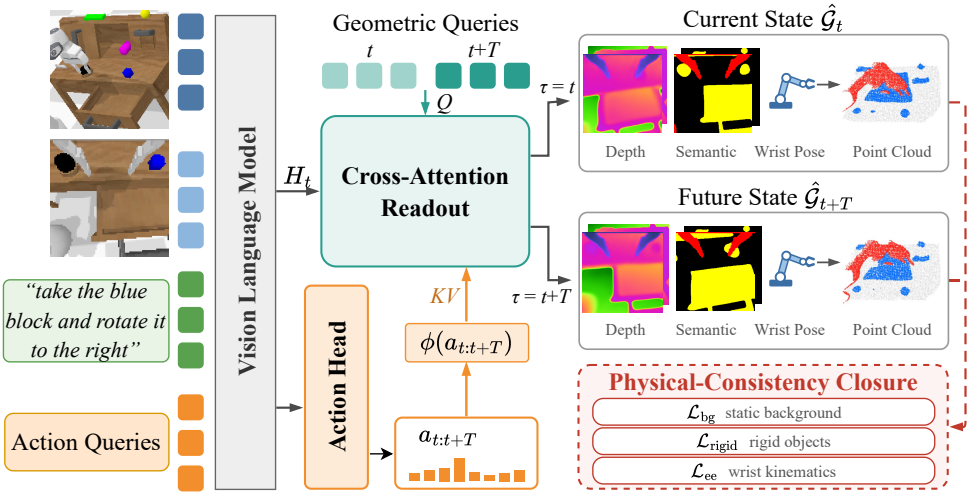


Figure 2: **Overview of the proposed framework.** ForeAct3D integrates semantic 3D prediction, physical-consistency closure, and action-conditioned future forecasting into a unified VLA policy for predicting task-relevant scene evolution under the planned action chunk.

frame-level action conditioning (Rigter et al., 2024; Zhu et al., 2025b). More recently, multi-view prediction and memory mechanisms have improved temporal consistency and enabled policy evaluation within generated rollouts (Guo et al., 2026). These advances strengthen action controllability and visual coherence, but video prediction alone does not guarantee that the underlying 3D state evolves subject to explicit physical constraints. For manipulation, this distinction matters because visually plausible predictions may still be geometrically inconsistent across views or fail to preserve object structure during an interaction. This motivates predicting scene geometry directly and constraining its evolution through physical relationships between current and future states.

Future World Modeling within VLA Policies. Another line of work integrates future prediction into VLA policies through auxiliary supervision of features shared with action generation. Prediction targets include latent features (Zheng et al., 2025; Sun et al., 2026) and visual or geometric observations (Wu et al., 2024a; Zhen et al., 2024; Zhang et al., 2026a; Qian et al., 2026; Liu et al., 2026). These objectives encourage anticipation but leave two gaps. First, future prediction alone does not enforce a physically coherent transition from the current scene. Second, shared features alone do not specify which planned action a predicted future corresponds to, motivating explicit action conditioning. ForeAct3D addresses both by jointly predicting current and future semantic 3D states, coupling them through physical-consistency constraints, and conditioning future queries on the policy-generated action chunk. Together, these components supervise structured scene evolution under the planned interaction within the representation used for action generation.

3 Method

3.1 Problem Formulation

Overview. To anticipate the consequences of a planned manipulation, we perform future world modeling through semantic 3D scene prediction within a VLA policy. The model estimates the current scene and predicts its future state over the policy's action horizon. As illustrated in Figure 2, structured geometry defines the state, physical-consistency constraints relate the two states across time, and explicit action conditioning grounds the predicted future in the policy's planned action.

At time t, the robot receives a language instruction ℓ and an observation set $o_t = \{o_t^v\}_{v \in \mathcal{V}}$ containing images from an external agent camera and a robot-mounted wrist camera, where $\mathcal{V}$ denotes these two views. The policy π maps (ℓ, o_t) to an action chunk $a_{t:t+T}$ of horizon T, which also defines the future prediction time $t + T$. This shared horizon aligns future-state supervision with the scene

changes induced by the planned action chunk. We predict the scene state at the action horizon's endpoint without generating a sequence of observations for the intermediate action steps.

Semantic 3D scene state. To capture the geometry relevant to manipulation, we represent each scene state using depth, three-class semantic masks, and camera pose. Depth encodes spatial structure, semantic masks separate background, robot, and object regions, and camera pose places each view in a common world frame. All objects share a single object class; the prediction neither assigns instance identities nor establishes object correspondence across the current and future states.

For each view v, depth D_t^v, known intrinsics K^v, and camera pose E_t^v define a point cloud $\mathcal{P}_t^v$ by back-projection Π^{-1}. The semantic mask S_t^v labels these points, so the scene state combines metric geometry with the semantic role of each region:

$$\mathcal{P}_t^v = \Pi^{-1}(D_t^v, K^v, E_t^v), \qquad \mathcal{G}_t = \{D_t^v,\, S_t^v,\, E_t^v\}_{v\in\mathcal{V}}. \tag{1}$$

ForeAct3D estimates the current state from the instruction and observations, and additionally conditions the future estimate on the planned action chunk. Both estimates live in the same semantic 3D representation, which allows physical constraints to relate the observed scene to its configuration after the action chunk:

$$\hat{\mathcal{G}}_t = f_\theta(o_t, \ell), \qquad \hat{\mathcal{G}}_{t+T} = f_\theta(o_t, \ell, a_{t:t+T}). \tag{2}$$

The current estimate recovers the scene the policy observes, whereas the future estimate predicts the consequences of the action it generates. This formulation supervises both endpoints of the interaction in a single representation and makes their geometric relationship part of policy learning.

3.2 Semantic Geometry from Policy Features

Geometric queries. To extract a structured scene state from the policy representation, we introduce six learnable queries $q^{(m,\tau)}$, indexed by geometric quantity $m \in \{\mathrm{depth}, \mathrm{seg}, \mathrm{pose}\}$ and prediction time $\tau \in \{t, t+T\}$. Each query cross-attends to the final backbone features H_t to produce a query feature $z^{(m,\tau)}$; future queries are further conditioned on the action chunk as described in Section 3.4. View embeddings are added before decoding: depth and segmentation queries serve both cameras, while pose queries use the wrist-view embedding to predict only the wrist-camera pose. Depth and segmentation share a decoder that attends to intermediate-layer backbone features of the corresponding view, followed by separate output heads. Because all queries read from the policy backbone, geometric supervision directly shapes the representation used for control.

Ground-truth instance IDs provide supervision distinct from the three predicted semantic classes. The rigidity constraint in Section 3.3 uses these IDs to partition the predicted geometry into individual objects and applies Kabsch alignment to each instance using pairs at shared valid pixels. This supplies instance-level physical supervision without requiring the segmentation head to predict object identities or establish object correspondence between the current and future scene states.

3.3 Physical-Consistency Closure

We impose three complementary constraints on predicted geometry and camera pose: background staticity, instance-level rigidity, and end-effector kinematics. The two geometric constraints operate on agent-view point clouds back-projected from predicted depth with ground-truth intrinsics and extrinsics. Since the agent camera is fixed, its extrinsics are identical at both endpoints, and the resulting points are directly comparable in world coordinates. Labeled calibration isolates depth consistency from pose prediction errors, while kinematics supervises the wrist-camera pose.

Background staticity. To keep the static environment fixed over time, we compare current and future background points back-projected from predicted depth at the same image pixels. A shared mask retains pixels labeled as valid background at both endpoints, thereby excluding regions occluded by the robot or objects at either time. The background loss $\mathcal{L}_{\mathrm{bg}}$ averages a Smooth-L1 distance $d(\cdot,\cdot)$ over these paired points; the same distance is used in the rigidity term below.

Instance-level rigidity. To preserve object shape under motion, we require each instance's current and future geometry to be related by a rigid transform. Ground-truth instance IDs select pixels that belong to the same object k and are valid at both endpoints. For each instance with at least 32

Table 1: Manipulation success rates (%) on LIBERO. Results cover four task suites and their average. Methods are grouped by training protocol and sorted within groups by average success.

Method	Venue, Year	Spatial	Object	Goal	Long	Avg.
One model per suite						
DreamVLA (Zhang et al., 2026a)	NeurIPS'25	97.5	94.0	89.5	89.5	92.6
QDepth-VLA (Li et al., 2025)	arXiv'26	97.6	96.6	95.2	90.0	94.9
UniVLA (Bu et al., 2025)	RSS'25	96.5	96.8	95.6	92.0	95.2
ST4VLA (Ye et al., 2026)	ICLR'26	98.0	99.0	93.8	92.6	95.9
Spatial Forcing (Li et al., 2026)	ICLR'26	**99.4**	**99.6**	**98.8**	**96.0**	**98.5**
One model for all suites						
UD-VLA (Chen et al., 2026)	ICLR'26	94.1	95.7	91.2	89.6	92.7
Evo-1 (Lin et al., 2026)	CVPR'26	92.7	97.7	96.3	92.3	94.8
UniVLA (Wang et al., 2026b)	ICLR'26	95.4	98.8	93.6	94.0	95.5
StarVLA-OFT (Community, 2026)	arXiv'26	97.8	98.6	96.2	93.8	96.6
FASTer (Liu et al., 2025)	ICLR'26	98.0	**99.4**	**98.6**	95.4	97.9
Ours	–	**99.2**	99.2	98.4	**96.2**	**98.3**

such pixels, we form paired world point sets $\hat{P}_{t,k}$ and $\hat{P}_{t+T,k}$. Kabsch alignment estimates the rigid transform $T_k \in \mathrm{SE}(3)$ between them, and the residual after alignment defines the rigidity loss over the retained instance set $\mathcal{K}$:

$$\mathcal{L}_{\text{rigid}} = \frac{1}{|\mathcal{K}|} \sum_{k \in \mathcal{K}} w_k \, d\Big(T_k \hat{P}_{t,k}, \hat{P}_{t+T,k}\Big) \,. \tag{3}$$

Same-pixel pairing approximates temporal correspondence without tracking surface points; it is a short-horizon approximation, since shared instance membership does not guarantee that paired pixels observe the same material point under large motion. The weight w_k down-weights instances likely being grasped: $w_k = 0.1$ when the gripper is closed and the current instance centroid lies within 0.10,m of the end effector, and $w_k = 1$ otherwise. At the batch level, $\mathcal{L}_{\text{rigid}}$ is averaged with equal weight over samples that have valid future targets and at least one retained instance.

End-effector kinematics. To anchor the predicted wrist-camera pose to the robot configuration, we supervise both endpoints using a reference pose from forward kinematics and the fixed hand–eye transform. The pose loss $\mathcal{L}_{\text{ee}}$ sums equally weighted translation MSE and geodesic rotation distance in radians over both prediction times. We convert the predicted 6D rotation into a rotation matrix before computing the loss. Pose supervision is separate from the background and rigidity constraints, which reconstruct world points using predicted depth and labeled camera poses.

Closure. The three constraints are combined into a single regularization objective, with coefficients λ_{bg}, λ_{rigid}, and λ_{ee} controlling their relative contributions:

$$\mathcal{L}_{\text{closure}} = \lambda_{\text{bg}} \mathcal{L}_{\text{bg}} + \lambda_{\text{rigid}} \mathcal{L}_{\text{rigid}} + \lambda_{\text{ee}} \mathcal{L}_{\text{ee}}. \tag{4}$$

Together, these constraints encourage a static background, approximately rigid object motion, and wrist-camera poses grounded in robot kinematics. Since geometric predictions are decoded from the policy backbone, these constraints directly regularize its representation for action generation.

3.4 Action-Conditioned State Prediction

The current scene alone does not determine an interaction's outcome: different actions can produce different future states from the same observation. The future branch therefore receives the policy-generated action chunk, explicitly tying its prediction to the planned interaction.

Action-conditioned prediction. To inject the planned action into future prediction, we project the action chunk into the query feature space with a small network ϕ. After all six queries have read scene information from the backbone (Section 3.2), only the three future queries are additionally conditioned on the action. Each future query cross-attends to the concatenation of the projected backbone features $\bar{H}_t$ and the action features $\phi(a_{t:t+T})$:

$$\tilde{z}^{(m,t+T)} = \mathrm{CrossAttn}\Big(z^{(m,t+T)};\, \bar{H}_t,\, \phi(a_{t:t+T})\Big) \,. \tag{5}$$

Table 2: Long-horizon performance on CALVIN ABC→D benchmark. Methods are grouped by training protocol and sorted by average rollout length. * marks results reproduced by us.

Method	Venue, Year	Avg. Len.	Task 1	Task 2	Task 3	Task 4	Task 5
With robot pretraining							
Retrieval-OpenVLA (Zhang et al., 2026b)	CVPR'26	3.53	92.7	79.6	70.5	58.3	51.9
Dita (Hou et al., 2025)	ICCV'25	3.61	94.5	82.5	72.8	61.3	50.0
GHIL-Glue (Hatch et al., 2025)	ICRA'25	3.69	95.2	88.5	72.3	62.9	49.8
ProgressVLA (Yan et al., 2026)	arXiv'26	3.73	95.2	84.8	73.6	67.2	52.0
MoDE (Reuss et al., 2025)	ICLR'25	4.01	96.2	88.9	81.1	71.8	63.5
Seer-Large (Tian et al., 2025)	ICLR'25	<u>4.28</u>	<u>96.3</u>	<u>91.6</u>	<u>86.1</u>	<u>80.3</u>	<u>74.0</u>
DreamVLA (Zhang et al., 2026a)	NeurIPS'25	**4.44**	**98.2**	**94.6**	**89.5**	**83.4**	**78.1**
Without robot pretraining							
Dita	ICCV'25	2.38	89.5	63.3	39.8	27.3	18.5
StarVLA-OFT*	arXiv'26	2.94	86.4	70.2	55.5	45.2	36.2
ProgressVLA	arXiv'26	3.24	89.4	76.8	63.0	52.2	43.1
MoDE	ICLR'25	3.39	91.5	79.2	67.3	55.8	45.3
CLOVER (Bu et al., 2024)	NeurIPS'24	3.53	**96.0**	<u>83.5</u>	70.8	57.5	45.4
Seer (scratch)	ICLR'25	<u>3.64</u>	93.0	82.4	<u>72.3</u>	<u>62.6</u>	<u>53.3</u>
Ours	–	**3.73**	<u>94.2</u>	**85.3**	**74.4**	**64.4**	**54.8**

This step fuses scene context with the planned action into the future query features $\tilde{z}^{(m,t+T)}$. Current queries skip this step and keep their scene-only features $z^{(m,t)}$. View embeddings are then added to both sets of queries before decoding depth, segmentation, and wrist-camera pose, as in Section 3.2.

4 Experiments

We first evaluate manipulation performance on LIBERO and CALVIN in Section 4.1, then assess predicted geometry, semantics, and future object localization in Section 4.2. We analyze the contributions of individual components through ablations in Section 4.3 and evaluate the framework on three real-world tabletop robot manipulation tasks in Section 4.4.

4.1 Manipulation Performance

LIBERO. Table 1 groups methods by training protocol across spatial, object, goal, and long-horizon tasks. Using one model for all suites, ForeAct3D achieves 98.3% average success, the highest among the listed methods under this protocol. It improves over StarVLA-OFT by 1.7 percentage points, with improvements across all four suites, especially Goal and Long, supporting the benefit of structured 3D prediction without suite-specific adaptation.

CALVIN. We evaluate long-horizon language-conditioned manipulation in the ABC→D setting, focusing on models without robot pretraining to limit differences in prior training data. ForeAct3D and StarVLA-OFT use the same training configuration, with the proposed geometric modeling modules as the primary difference. Under this training configuration, ForeAct3D increases the average number of completed subtasks from 2.94 to 3.73 compared with StarVLA-OFT (Table 2), supporting improved sequential task execution under matched training.

4.2 Geometry, Semantics, and Future Localization

We evaluate three aspects of the predictive state separately: geometry reconstructed from predicted depth within GT object regions, three-class semantic segmentation, and future localization using predicted depth and semantic masks. Table 3 summarizes the results; full metrics and protocols appear in Appendices A.4 and A.5. GT instance IDs define the evaluated objects in both localization settings, so predicted-mask localization is not an evaluation of autonomous instance discovery.

Geometry within known object regions. Table 3(a) isolates depth-derived geometry from segmentation quality. On CALVIN, ForeAct3D achieves lower future instance-level errors than CLOVER, while CLOVER has lower scene-level error. Their horizons differ: ForeAct3D uses $T = 8$, CLOVER uses $T = 5$, and DreamVLA uses $H = 3$. Thus, the comparison demonstrates object-

Table 3: **Geometry, semantics, and future localization.** (a) CALVIN geometry; instance metrics use GT masks and horizons differ by method. (b) ForeAct3D semantic mIoU and localization at $t + T$. GT / Pred denotes GT / predicted semantic masks; both retain GT instance IDs. Recall is in percent. Details and protocol qualifications are in Appendix A.5.

(a) Future geometry: scene and GT-mask object metrics

Method	Horizon	Scene CD (m) ↓	Inst-CD (m) ↓	Trans (cm) ↓	Recall ↑
CLOVER	5	**0.0195**	0.0139	1.741	75.82
DreamVLA	3	0.1350	0.1455	20.61	10.73
ForeAct3D	8	0.0206	**0.01371**	**1.245**	**92.78**

(b) Semantic prediction and localization with predicted masks

Benchmark	mIoU (%), t / $t + T$ ↑	Trans (cm), GT / Pred ↓	Recall, GT / Pred ↑
LIBERO	89.3 / 88.8	1.06 / 0.87	94.0 / 94.9
CALVIN	88.0 / 85.8	1.25 / 3.49	92.8 / 65.9

Table 4: **Policy and prediction ablations.** (a) Cumulative policy ablations on LIBERO and closure ablations on CALVIN. CALVIN variants retain semantic 3D prediction and action conditioning. Scores report LIBERO success (%, averaged over four suites) and mean CALVIN task length. (b) Future geometry with and without action conditioning. Detailed metrics appear in Appendix A.6.

(a) Policy performance

Benchmark	Configuration	Score ↑
LIBERO	StarVLA-OFT	96.6
	+ Semantic 3D prediction	97.3
	+ Physical-consistency closure	97.9
	+ Action conditioning	**98.3**
CALVIN	Full	**3.73**
	w/o kinematics	3.68
	w/o background	3.65
	w/o rigidity	3.55
	w/o closure	3.45

(b) Future geometry

	Metric	Full	w/o action
LIBERO	Inst-CD (m) ↓	**0.0088**	0.0124
	Trans (cm) ↓	**1.06**	2.59
	Recall@2cm ↑	**0.940**	0.884
CALVIN	Inst-CD (m) ↓	**0.0137**	0.0183
	Trans (cm) ↓	**1.25**	1.77
	Recall@2cm ↑	**0.928**	0.799

level geometric accuracy at these horizons, rather than a matched-horizon advantage. The full current/future results in Table 6 show that ForeAct3D's GT-mask Recall@2cm changes modestly from 0.9416 to 0.9278. These conditional scores exclude instances with insufficient valid depth without penalty and do not evaluate predicted segmentation quality or object detection coverage.

Semantic regions across time. Table 3(b) directly evaluates the segmentation output. Mean IoU changes from 89.3% to 88.8% on LIBERO and from 88.0% to 85.8% on CALVIN. The detailed class scores in Table 7 reveal a less uniform picture: LIBERO Spatial has 45.7% future object IoU despite 91.5% background and 95.7% robot IoU. Small changes in aggregate mIoU therefore do not imply uniformly accurate object masks across individual tasks and semantic classes.

Localization with predicted semantic support. Using predicted masks on CALVIN increases translation error from 1.25 to 3.49 cm and reduces Recall@2cm from 92.8% to 65.9%, exposing errors hidden by the GT-mask evaluation. On LIBERO, the corresponding comparison reports 0.87 cm and 94.9% with predicted masks, versus 1.06 cm and 94.0% with GT masks. Lower conditional error under predicted masks does not establish better scene recovery: mask selection changes which points contribute to localization. LIBERO GT-mask results use the protocol in Appendix A.6.

4.3 What Makes the Predictive State Useful?

We evaluate policy success with the same backbone, training data, schedule, and evaluation protocol. The action-conditioning ablation also keeps the geometric queries and physical-consistency closure unchanged to isolate the contribution of the action input to future-state prediction.

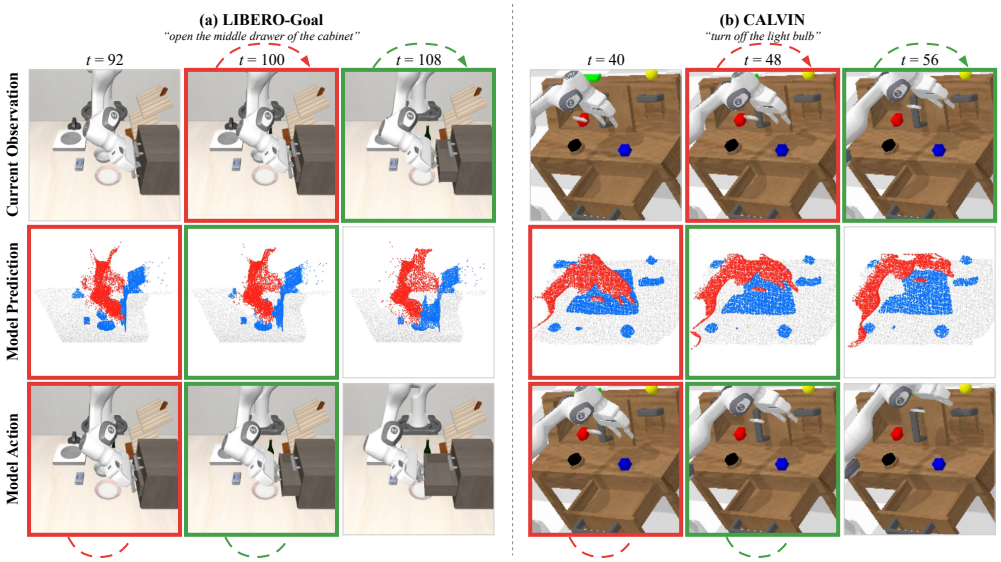


Figure 3: **Scene prediction and policy execution in simulation.** Examples show opening a drawer in LIBERO-Goal and turning off a light in CALVIN. Rows present current observations, predicted semantic 3D states, and execution results. Matching colored borders link corresponding observation and execution frames across successive steps of each manipulation sequence.

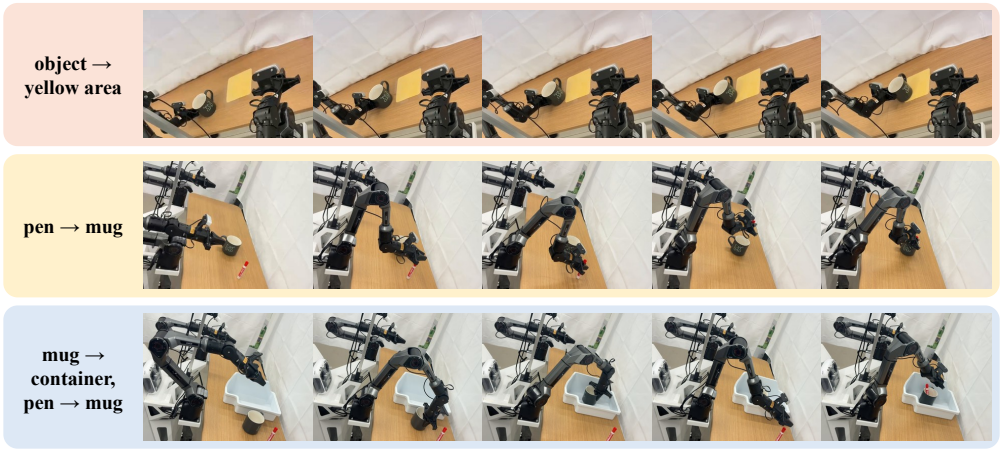


Figure 4: **Real-world task execution.** Rows show moving an object to the yellow area, placing a pen in a mug, and placing the mug in a container before inserting the pen. Frames proceed from left to right and illustrate the spatial goals and sequential structure of the three tasks.

From geometric supervision to the full model. Table 4(a) shows a progressive improvement from StarVLA-OFT (96.6%) to semantic 3D prediction (97.3%), physical-consistency closure (97.9%), and action conditioning (98.3%). These gains support complementary contributions from the three components to downstream manipulation performance across the evaluated LIBERO task suites.

Contributions of physical constraints. Table 4(a) evaluates the closure terms on CALVIN with semantic 3D prediction and action conditioning retained. Removing background consistency, instance rigidity, or kinematic supervision reduces average task length from 3.73 to 3.65, 3.55, and 3.68. Removing all closure terms further reduces it to 3.45. The largest individual decrease occurs without rigidity, supporting its contribution to sequential manipulation.

Table 5: **Real-world cumulative ablations.** Rates (%) are followed by counts in parentheses, with 15 trials per configuration and task. Reach means the object entered the yellow area at least once; success requires it to remain there at termination. First stage records long-task mug placement.

	Yellow area		Pen in mug	Long task	
Configuration	Success	Reach	Success	Full	First stage
StarVLA-OFT	0.0 (0/15)	6.7 (1/15)	20.0 (3/15)	0.0 (0/15)	13.3 (2/15)
+ Semantic 3D	13.3 (2/15)	33.3 (5/15)	46.7 (7/15)	6.7 (1/15)	33.3 (5/15)
+ Physical consistency	13.3 (2/15)	**40.0 (6/15)**	**60.0 (9/15)**	6.7 (1/15)	**40.0 (6/15)**
+ Action conditioning (Ours)	**33.3 (5/15)**	33.3 (5/15)	**60.0 (9/15)**	**20.0 (3/15)**	**40.0 (6/15)**

Predicted states during execution. Figure 3 presents observations, predicted semantic 3D states, and execution results for drawer opening in LIBERO-Goal and light switching in CALVIN. Future predictions provide training supervision and are not used to select or refine actions during execution.

Action input and object motion. Table 4(b) compares future geometry with and without action conditioning. On CALVIN, removing the action input raises Inst-CD from 0.0137 to 0.0183 m and translation error from 1.25 to 1.77 cm, while Recall@2cm falls from 0.928 to 0.799. On LIBERO, macro-averaged translation error rises from 1.06 to 2.59 cm and recall falls from 0.940 to 0.884, with consistent improvements across all four benchmark suites (Appendix A.6).

4.4 Real-World Manipulation

We evaluate ForeAct3D on three tabletop manipulation tasks using the AgileX platform (Figure 4). We collect 50 demonstrations per task and evaluate each configuration over 15 trials per task. Depth camera measurements and SAM3 (Carion et al., 2026) segmentation provide geometric supervision.

Task design. The tasks probe spatial relations, precise placement, and sequential dependencies. *Moving the mug to the yellow area* tests target-region retention: success requires the mug to remain inside at termination, while reach records entry at any point. This distinguishes reaching a target from maintaining the intended final configuration after the interaction has ended. *Placing the pen in the mug* requires accurate relative geometry of the pen, mug opening, and gripper. The long task, *placing the mug in a container and then placing the pen in the mug*, tests sequential dependence: the first action establishes the second-stage target pose. Full success requires both stages; mug placement is also reported as intermediate progress within the same trials.

Result analysis. Table 5 shows average success increasing from 6.7% to 22.2%, 26.7%, and 37.8% as components are added, from 3/45 successes for StarVLA-OFT to 17/45 for ForeAct3D. Semantic 3D prediction provides the largest gain, particularly for pen and mug placement through elevated openings, where depth informs placement and release height. Physical-consistency closure further improves pen placement from 7/15 to 9/15 and first-stage mug placement from 5/15 to 6/15, consistent with benefits from more coherent geometry. With action conditioning, all 5/5 trials reaching the yellow area retain the mug there, compared with 2/6 without it, consistent with improved maintenance of the target spatial relation at termination. Long-task completion also rises from 1/15 to 3/15 while first-stage mug placement remains at 6/15. The gain occurs in the dependent second stage, whose target mug pose is established by preceding robot actions. This supports conditioning future predictions on planned interactions when later actions depend on earlier outcomes.

5 Conclusion

We presented ForeAct3D, a framework for future world modeling in semantic 3D space that combines structured scene prediction, physical-consistency closure, and action-conditioned forecasting within VLA policies. ForeAct3D achieves 98.3% average success on LIBERO and an average task length of 3.73 on CALVIN, with ablations supporting complementary contributions from its three components. Real-world cumulative ablations increase average success from 6.7% to 37.8% across three tabletop tasks, with action conditioning improving target-region retention in the evaluated trials. These results support physically constrained future-state prediction as a useful form of supervision for robot policy learning. Extending the state constraints beyond rigid scenes and using predicted futures to refine actions at inference time remain directions for future work.

AI USE STATEMENT

In this work, we used generative AI tools for drafting and revising portions of the manuscript and assisting with literature discovery. We have not used generative AI tools for conducting experiments or generating synthetic datasets, and proving mathematical claims is not applicable to this work. Additionally, we used generative AI tools for language polishing, grammar correction, and improving clarity and organization. We have reviewed all AI-assisted work. Specifically, we checked technical descriptions against the implementation, verified numerical claims against experimental records, and checked references against their original sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies robot manipulation in simulation and on tabletop tasks, with the aim of improving how policies represent and anticipate physical interactions. Learned policies can still produce inaccurate predictions or unsafe actions, especially outside the evaluated settings. The reported results do not establish safe operation around people or in safety-critical applications. Deployment should therefore include task-specific risk assessment, appropriate human oversight, and independent safety mechanisms. The assumptions and limitations discussed in Appendix A.7 should guide the interpretation of results and any extension to new objects, environments, or robot platforms.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-loop visuomotor control with generative expectation for robotic manipulation. *Advances in Neural Information Processing Systems*, 37:139002–139029, 2024.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Univla: Learning to act anywhere with task-centric latent actions. *arXiv preprint arXiv:2505.06111*, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris Coll-Vinent, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, et al. Sam 3: Segment anything with concepts. In *International conference on learning representations*, volume 2026, pp. 138846–138923, 2026.

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

Jiayi Chen, Wenxuan Song, Pengxiang Ding, Ziyang Zhou, Han Zhao, Barrett Tang, Donglin Wang, and Haoang Li. Unified diffusion vla: Vision-language-action model via joint discrete denosing diffusion process. In *International Conference on Learning Representations*, volume 2026, pp. 139291–139311, 2026.

StarVLA Community. Starvla: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026.

Chelsea Finn and Sergey Levine. Deep visual foresight for planning robot motion. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 2786–2793. IEEE, 2017.

Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. *Advances in neural information processing systems*, 29, 2016.

Yanjiang Guo, Lucy Shi, Jianyu Chen, and Chelsea Finn. Ctrl-world: A controllable generative world model for robot manipulation. In *International Conference on Learning Representations*, volume 2026, pp. 6121–6138, 2026.

Kyle B Hatch, Ashwin Balakrishna, Oier Mees, Suraj Nair, Seohong Park, Blake Wulfe, Masha Itkina, Benjamin Eysenbach, Sergey Levine, Thomas Kollar, et al. Ghil-glue: Hierarchical control with filtered subgoal images. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9516–9524. IEEE, 2025.

Zhi Hou, Tianyi Zhang, Yuwen Xiong, Haonan Duan, Hengjun Pu, Ronglei Tong, Chengyang Zhao, Xizhou Zhu, Yu Qiao, Jifeng Dai, et al. Dita: Scaling diffusion transformer for generalist vision-language-action policy. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7686–7697. IEEE, 2025.

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.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Yann LeCun et al. A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, 62(1):1–62, 2022.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 132324–132345, 2026.

Yixuan Li, Yuhui Chen, Mingcai Zhou, Haoran Li, Zhengtao Zhang, and Dongbin Zhao. Qdepth-vla: Quantized depth prediction as auxiliary supervision for vision-language-action models. *arXiv preprint arXiv:2510.14836*, 2025.

Tao Lin, Yilei Zhong, Yuxin Du, Jingjing Zhang, Jiting Liu, Yinxinyu Chen, Encheng Gu, Ziyan Liu, Hongyi Cai, Yanwen Zou, et al. Evo-1: Lightweight vision-language-action model with preserved semantic alignment. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13397–13406, 2026.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

Jiaming Liu, Qingpo Wuwu, Nuowei Han, Hao Chen, Zhuoyang Liu, Fan Fei, Yueru Jia, Chenyang Gu, Yandong Guo, Boxin Shi, et al. Lift3d-vla: Lifting vla models to 3d geometry and dynamics-aware manipulation. *arXiv preprint arXiv:2607.06564*, 2026.

Yicheng Liu, Shiduo Zhang, Zibin Dong, Baijun Ye, Tianyuan Yuan, Xiaopeng Yu, Linqi Yin, Chenhao Lu, Junhao Shi, Luca Jiang-Tao Yu, et al. Faster: Toward efficient autoregressive vision language action modeling via neural action tokenization. *arXiv preprint arXiv:2512.04952*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

Jingjing Qian, Boyao Han, Chen Shi, Lei Xiao, Long Yang, Shaoshuai Shi, and Li Jiang. Geopredict: Leveraging predictive kinematics and 3d gaussian geometry for precise vla manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13529–13539, 2026.

Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. In *International Conference on Learning Representations*, volume 2025, pp. 17247–17275, 2025.

Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. Avid: Adapting video diffusion models to world models. *arXiv preprint arXiv:2410.12822*, 2024.

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

Yang Tian, Sizhe Yang, Jia Zeng, Ping Wang, Dahua Lin, Hao Dong, and Jiangmiao Pang. Predictive inverse dynamics models are scalable learners for robotic manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 92033–92052, 2025.

Jiaxu Wang, Yicheng Jiang, Tianlun He, Jingkai Sun, Qiang Zhang, Junhao He, Jiahang Cao, Zesen Gan, Mingyuan Sun, Qiming Shao, et al. Mvista-4d: View-consistent 4d world model with test-time action inference for robotic manipulation. *arXiv preprint arXiv:2602.09878*, 2026a.

Lirui Wang, Kevin Zhao, Chaoqi Liu, and Xinlei Chen. Learning real-world action-video dynamics with heterogeneous masked autoregression. *arXiv preprint arXiv:2502.04296*, 2025.

Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. In *International Conference on Learning Representations*, volume 2026, pp. 80929–80944, 2026b.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024a.

Jialong Wu, Shaofeng Yin, Ningya Feng, Xu He, Dong Li, Jianye Hao, and Mingsheng Long. ivideogpt: Interactive videogpts are scalable world models. *Advances in Neural Information Processing Systems*, 37:68082–68119, 2024b.

Hongyu Yan, Qiwei Li, Jiaolong Yang, and Yadong Mu. Progressvla: Progress-guided diffusion policy for vision-language robotic manipulation. *arXiv preprint arXiv:2603.27670*, 2026.

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 2023.

Jinhui Ye, Fangjing Wang, Ning Gao, Junqiu Yu, Yangkun Zhu, Bin Wang, Jinyu Zhang, Weiyang Jin, Yanwei Fu, Feng Zheng, et al. St4vla: Spatially guided training for vision-language-action models. *arXiv preprint arXiv:2602.10109*, 2026.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *Advances in Neural Information Processing Systems*, 38: 24195–24228, 2026a.

Yue Zhang, Rui Wang, Jiehong Lin, Zhongrui Wang, and Xiaojuan Qi. Retrieval-vla: Training-free in-context adaptation for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1358–1367, 2026b.

Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024.

Ruijie Zheng, Jing Wang, Scott Reed, Johan Bjorck, Yu Fang, Fengyuan Hu, Joel Jang, Kaushil Kundalia, Zongyu Lin, Loic Magne, et al. Flare: Robot learning with implicit world modeling. *arXiv preprint arXiv:2505.15659*, 2025.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. *arXiv preprint arXiv:2504.02792*, 2025a.

Fangqi Zhu, Hongtao Wu, Song Guo, Yuxiao Liu, Chilam Cheang, and Tao Kong. Irasim: A fine-grained world model for robot manipulation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9834–9844. IEEE, 2025b.

A Appendix

A.1 Physical-Consistency Formulation

World-coordinate reconstruction. To make the geometric constraints explicit, let $u = (u_x, u_y)$ denote a pixel and $\tilde{u} = (u_x, u_y, 1)^\top$ its homogeneous coordinate. For prediction time $\tau \in \{t, t+T\}$, the agent-view depth $\hat{D}_\tau(u)$ defines a world point through labeled camera calibration:

$$\hat{p}_\tau(u) = R_{\mathrm{wc}}\Big(\hat{D}_\tau(u) K^{-1}\tilde{u}\Big) + t_{\mathrm{wc}}. \tag{6}$$

Here, K is the intrinsic matrix and $(R_{\mathrm{wc}}, t_{\mathrm{wc}})$ maps camera coordinates to world coordinates. The fixed agent camera uses the same calibration at both endpoints. Consequently, background and rigidity residuals compare predicted scene geometry in one common frame; the predicted wrist-camera pose is supervised separately against its labeled pose at each endpoint.

Residual distance. To specify the distance used by both geometric constraints, consider paired point sets $P = \{p_i\}_{i=1}^N$ and $Q = \{q_i\}_{i=1}^N$. We sum coordinatewise Smooth-L1 penalties and then average over points, using the transition threshold $\beta = 0.05\,\mathrm{m}$:

$$\begin{aligned} d(P,Q) &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{3} h_\beta(p_{ij} - q_{ij}), \\ h_\beta(r) &= \begin{cases} r^2/(2\beta), & |r| < \beta, \\ |r| - \beta/2, & |r| \geq \beta. \end{cases} \end{aligned} \tag{7}$$

This definition averages over paired points without dividing the coordinate sum by three. It therefore fixes the reduction convention as well as the quadratic-to-linear transition, which is needed to interpret the scale of the background and rigid-alignment losses consistently.

Background correspondence. To identify comparable static regions, let V_τ be the labeled depth-validity mask and S_τ the ground-truth semantic map. The shared background set contains pixels valid and labeled as background at both endpoints. For sample b, write P_b^M and Q_b^M for reconstructed current and future points selected at the same pixel coordinates in a set M. The background objective is

$$\begin{aligned} M_b^{\mathrm{bg}} &= \{u : V_t(u) \wedge V_{t+T}(u),\ S_t(u) = S_{t+T}(u) = 0\}, \\ \mathcal{L}_{\mathrm{bg}} &= \frac{1}{|\mathcal{B}_{\mathrm{bg}}|}\sum_{b\in\mathcal{B}_{\mathrm{bg}}} d\Big(P_b^{M_b^{\mathrm{bg}}}, Q_b^{M_b^{\mathrm{bg}}}\Big). \end{aligned} \tag{8}$$

The eligible set $\mathcal{B}_{\mathrm{bg}}$ contains samples with valid future targets and at least 32 shared background pixels. A pixel occupied by the robot or an object at either endpoint is excluded. All eligible samples have equal weight after their pointwise residuals are averaged, preventing samples with larger visible backgrounds from dominating the batch loss solely because of their pixel counts.

Rigid alignment. To construct an instance pair, use ground-truth instance IDs I_t and I_{t+T} to retain valid pixels satisfying $I_t(u) = I_{t+T}(u) = k$ for object k. Instances with fewer than 32 shared pixels are excluded. For one retained pair, omit the sample and instance subscripts and denote the resulting points by $P = \{p_i\}$ and $Q = \{q_i\}$. Their unweighted centroids and cross-covariance are

$$\bar{p} = \frac{1}{N}\sum_i p_i, \qquad \bar{q} = \frac{1}{N}\sum_i q_i, \qquad H = \frac{1}{N}\sum_i (p_i - \bar{p})(q_i - \bar{q})^\top. \tag{9}$$

Using the singular value decomposition $H = U\Sigma V^\top$, Kabsch alignment produces a rotation and translation with no scale parameter. The determinant correction excludes reflections, yielding a transform that minimizes the squared distance between the paired point sets:

$$R = V\,\mathrm{diag}\big(1, 1, \det(VU^\top)\big)\,U^\top, \qquad t = \bar{q} - R\bar{p}. \tag{10}$$

We then evaluate $d(RP + t, Q)$ using Equation 7. The transform is thus fitted with a squared-distance criterion and scored with Smooth-L1 residuals. Pairing follows shared pixel coordinates rather than tracked surface points; identical instance membership alone does not establish material-point correspondence when objects move between the two prediction endpoints.

Instance and sample aggregation. To combine object residuals, let $\mathcal{K}_b$ contain the retained instances in sample b, and let $\mathcal{B}_{\mathrm{rigid}}$ contain samples with valid future targets and at least one retained instance. We average instance residuals within each sample and then average eligible samples:

$$\mathcal{L}_{\mathrm{rigid}} = \frac{1}{|\mathcal{B}_{\mathrm{rigid}}|} \sum_{b \in \mathcal{B}_{\mathrm{rigid}}} \frac{1}{|\mathcal{K}_b|} \sum_{k \in \mathcal{K}_b} w_{bk}\, d(R_{bk}P_{bk} + t_{bk}, Q_{bk}). \tag{11}$$

The contact weight is 0.1 when the gripper-closure signal is active and the current instance centroid is within 0.10 m of the end-effector position, and 1 otherwise. The implementation uses the translation of the labeled wrist-camera pose as the position proxy for this test. The denominator remains the number of retained instances, not the sum of contact weights, so downweighting decreases the contribution of contact-associated residuals without renormalizing the remaining objects.

Wrist-pose consistency. To compare predicted and reference wrist poses, the pose head outputs a translation $\hat{x}_\tau$ and two three-dimensional rotation vectors. Normalization and orthogonalization of the vectors, followed by their cross product, form $\hat{R}_\tau$. For reference pose (R_τ, x_τ), the per-sample pose error is

$$\ell^\tau_{\mathrm{pose}} = \frac{1}{3}\|\hat{x}_\tau - x_\tau\|_2^2 + \arccos\left(\frac{\mathrm{tr}(\hat{R}_\tau R_\tau^\top) - 1}{2}\right). \tag{12}$$

The inverse-cosine argument is clamped to $[-1+10^{-6}, 1-10^{-6}]$. Current and future pose errors are each averaged over their valid samples and then summed, with future validity handled as described below. Translation MSE and rotation angle have unit weights. The scene-supervision and closure terms use the same pose error; their coefficients add when supervising each predicted wrist pose.

A.2 Dense Supervision and Validity Handling

Depth supervision. For one view and prediction time, let $e = \hat{D} - D$ denote the depth error and let M select valid ground-truth depths satisfying $0 < D < d_{\max}$. We set $d_{\max} = 2.0$ m for the agent camera and 1.0 m for the wrist camera; these training cutoffs are separate from the evaluation protocol in Appendix A.4. Write $\langle x \rangle_M$ for the mean over valid pixels across the batch. The depth objective combines pointwise error and adjacent-pixel differences:

$$\mathcal{L}_D = \langle |e| \rangle_M + 0.5\big(\langle |\Delta_x e| \rangle_{M_x} + \langle |\Delta_y e| \rangle_{M_y}\big). \tag{13}$$

Here, Δ_x and Δ_y are adjacent-pixel differences, and M_x and M_y retain pairs whose two endpoints are valid. Each directional term uses its own valid-pair denominator, so pairs crossing invalid regions are excluded from both the error sum and the corresponding normalization.

Semantic supervision. The segmentation objective combines pixelwise cross-entropy and soft Dice loss over the three classes: background, robot, and object. Let p_{ic} be the softmax probability and y_{ic} the one-hot target for class c at valid pixel i. Dice statistics pool valid pixels across the batch, computing one overlap score per semantic class before averaging the three class losses:

$$\mathcal{L}_{\mathrm{Dice}} = 1 - \frac{1}{3}\sum_{c=0}^{2} \frac{2\sum_{i \in M_S} p_{ic} y_{ic} + 1}{\sum_{i \in M_S} p_{ic} + \sum_{i \in M_S} y_{ic} + 1}. \tag{14}$$

The semantic validity mask M_S excludes invalid labels without imposing the depth cutoff. Cross-entropy is averaged over the same valid pixels, and the additive smoothing constant makes the Dice ratio well defined for empty support. Cross-entropy and Dice have unit weights, with uniform class weights; instance IDs supervise physical constraints without adding segmentation classes.

Missing future targets and empty support. A sample-level validity flag gates all future targets. For depth and segmentation, it is incorporated into pixel masks before reduction, excluding invalid samples from both numerators and denominators. Future pose losses similarly average only over samples with valid future targets. An entirely invalid future batch contributes zero future supervision, while current-state and action supervision remain active. Background and rigidity return zero when no sample has sufficient shared support, using separate denominators for eligible samples.

Training and evaluation support. The 32-pixel threshold above governs physical regularization, while the evaluation rules in Appendix A.4 select objects with at least 30 mask pixels and at least three valid reconstructed points. Training depth cutoffs are not applied to the reported geometric metrics. Geometric evaluation first averages inference windows within an episode and then weights episodes equally, whereas dense supervision pools valid pixels across the training batch. These reductions preserve the distinction between pixel-level supervision and episode-level evaluation.

A.3 Benchmarks and Implementation Details

LIBERO. We evaluate on four LIBERO task suites: Spatial, Object, Goal, and Long. Each suite contains 10 tasks, with 50 evaluation episodes per task and 500 episodes per suite. A single policy is trained across all four suites. We report per-suite success rates and their equally weighted average, covering spatial reasoning, object manipulation, goal-conditioned behavior, and long-horizon tasks.

CALVIN. We adopt the ABC→D setting, training on environments A, B, and C and evaluating zero-shot on the unseen environment D. Evaluation consists of 1,000 rollouts, each containing a sequence of five language-conditioned subtasks. Performance is measured by the average number of successfully completed consecutive subtasks per rollout, with a maximum value of 5. Table 2 additionally reports success rates for completing successive subtasks in the sequence.

Implementation details. We use the same training configuration for LIBERO and CALVIN, including all ablation variants. Models are trained using AdamW (Loshchilov & Hutter, 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.95$, optimizer $\epsilon = 10^{-8}$, and weight decay 10^{-8}. The peak learning rate is 2.5×10^{-5} for the vision-language backbone and 1.0×10^{-4} for both the action module and geometric prediction heads. We apply linear warmup followed by cosine decay, with the scheduler's minimum learning-rate parameter set to 2×10^{-7}, and clip the global gradient norm to 1.0.

Training runs for 30,000 optimization steps, including 5,000 learning-rate warmup steps, on 16 NVIDIA A100 GPUs with 40 GB of memory each. The batch size is 8 per GPU with no gradient accumulation, giving a global batch size of 128. Ablation variants retain these settings while omitting the specified modules and their associated losses; absent prediction heads receive no supervision. Physical constraints and dense supervision are defined in Appendices A.1 and A.2.

The current- and future-head weights and the overall scene-supervision multiplier are all 1.0. The three closure terms have unit relative weights. Their overall coefficient increases linearly from 0 to 0.1 over the first 2,000 training iterations and remains at 0.1 thereafter; this schedule is separate from the learning-rate warmup for the backbone and prediction heads.

A.4 Geometric Evaluation Protocol

Views and prediction horizons. All geometric metrics are evaluated from the agent camera and compared with ground truth at the corresponding timestep. Table 6 reports current and future states separately; only future predictions are evaluated for CLOVER and DreamVLA. ForeAct3D uses a prediction horizon of $T = 8$, CLOVER uses its original checkpoint with $T = 5$, and DreamVLA uses $H = 3$. These comparisons characterize prediction accuracy at each model's evaluation horizon and do not isolate performance differences under a shared horizon.

Depth evaluation. Depth metrics use pixels where both ground-truth and predicted depth exceed 10^{-3} m. All valid regions, including background, contribute without near/far clipping or foreground filtering. Ground-truth depth is resized by nearest-neighbor interpolation to the prediction resolution. Predictions are evaluated in meters without median scaling or scale-and-shift alignment. We report absolute relative error (AbsRel), root mean squared error (RMSE), and depth accuracy δ_1.

Point sampling and Chamfer distance. Scene Chamfer distance (Scene CD) uses 2,048 points sampled uniformly at random from each point cloud. Instance Chamfer distance (Inst-CD) uses all

Table 6: **Complete depth-derived geometry on CALVIN.** Instance metrics use GT masks. Rows separate current and future states; future horizons are 5 (CLOVER), 3 (DreamVLA), and 8 (ForeAct3D). Depth RMSE, CD, and EMD use meters; translation uses centimeters; recall is a fraction.

		Depth			Scene		GT instance regions		
Method	State	AbsRel	RMSE	δ_1	CD	EMD	Inst-CD	Trans	Recall
CLOVER	Future	0.0083	0.1252	0.9955	0.0195	0.0601	0.0139	1.741	0.7582
DreamVLA	Future	0.0867	0.5583	0.9126	0.1350	0.3145	0.1455	20.61	0.1073
ForeAct3D	Current	0.0110	0.1551	0.9942	0.0206	0.0631	0.0129	1.165	0.9416
ForeAct3D	Future	0.0117	0.1633	0.9933	0.0206	0.0623	0.01371	1.245	0.9278

Table 7: **Three-class semantic segmentation quality (%).** Future class IoUs are reported for background, robot, and object. All objects share the same semantic class. LIBERO Average denotes the mean across the Object, Goal, Spatial, and Long task suites.

Benchmark	Suite	mIoU@t	BG@$t+T$	Robot@$t+T$	Object@$t+T$	mIoU@$t+T$
LIBERO	Object	98.0	99.4	95.8	98.0	97.7
	Goal	87.2	94.8	94.8	70.8	86.8
	Spatial	78.1	91.5	95.7	45.7	77.6
	Long	93.7	97.2	94.5	87.8	93.2
	Average	89.3	95.7	95.2	75.6	88.8
CALVIN	ABC→D	88.0	89.8	90.3	77.3	85.8

valid points within each ground-truth object mask without downsampling. For point sets A and B, we compute the symmetric mean of unsquared Euclidean nearest-neighbor distances:

$$\mathrm{CD}(A,B) = \frac{1}{2}\left(\frac{1}{|A|}\sum_{a\in A}\min_{b\in B}\|a-b\|_2 + \frac{1}{|B|}\sum_{b\in B}\min_{a\in A}\|b-a\|_2\right). \tag{15}$$

Both metrics are reported in meters: Scene CD measures agreement over the sampled scene reconstruction, while Inst-CD measures geometry within ground-truth object regions.

Earth Mover's Distance. Scene Earth Mover's Distance (Scene EMD) uses 1,024 randomly sampled points from each cloud with a fixed random seed of 0. We compute EMD by solving the discrete optimal transport problem between the predicted and ground-truth point clouds. The cost matrix contains unsquared Euclidean distances, and each cloud has uniform point weights summing to one. Mean transport distance is reported in meters without normalizing coordinates or costs.

Instance selection and localization metrics. Ground-truth instance masks select the same valid pixel locations from predicted and ground-truth depth, defining a pair of point clouds for each object. Instance association therefore uses ground-truth IDs rather than predicted instance labels or geometric matching. These metrics assess depth-derived geometry within known object regions; predicted semantic segmentation is evaluated separately as a distinct prediction task.

We exclude object ID 0, which represents background and robot regions, and objects with fewer than 30 mask pixels. Invalid predicted-depth pixels are removed from both point clouds. Instances with fewer than three remaining valid points are excluded from the metric averages and recall denominator without penalty. Translation error is the Euclidean distance between the predicted and ground-truth centroids computed from the retained points. Within each evaluation window, Recall@2cm is the fraction of retained instances with centroid error strictly below 2 cm. It therefore measures localization among evaluable instances rather than coverage of all ground-truth objects.

Metric aggregation. Metrics are first averaged over inference windows within each episode, with approximately one window per action chunk. A window that is invalid for a particular metric is excluded only from that metric's average. Episode means are then averaged with equal weight, regardless of the number of valid windows. Recall@2cm is consequently an average of window-level fractions rather than a pooled ratio over all retained instances. LIBERO summary results use an equally weighted average across all four benchmark task suites.

A.5 Detailed Geometry and Semantic Prediction Results

Geometry and semantic segmentation. Table 6 reports the complete depth and reconstructed-geometry results on CALVIN. Table 7 reports semantic segmentation quality for current and future states, including future IoU for background, robot, and object regions. All objects share one semantic class. These evaluations separately characterize depth-derived geometry within ground-truth instance regions and the accuracy of the predicted semantic masks.

Future localization with predicted masks. Table 8 evaluates localization at $t + T$ using predicted depth with either ground-truth or predicted semantic masks. Both settings use ground-truth instance IDs to identify objects. Predicted-mask results thus assess localization using predicted semantic support, rather than instance detection without ground-truth assistance.

The filtering and aggregation rules in Appendix A.4 describe the GT-mask geometry evaluation. Identical instance filtering across the two mask settings has not been established, and mask-dependent point selection can change both centroid estimates and the evaluated instance population. Differences between the two settings therefore cannot be attributed solely to segmentation accuracy. In particular, lower localization error with predicted masks does not by itself indicate better object coverage. LIBERO GT-mask results follow the action-conditioning protocol in Table 10.

Table 8: **Future localization with predicted depth and semantic masks.** GT instance IDs are retained in both mask settings. Recall@2cm is in percent. These results assess localization with predicted semantic support, not autonomous instance detection without GT IDs.

		Trans (cm) ↓		Recall@2cm ↑	
Benchmark	Suite	GT mask	Pred mask	GT mask	Pred mask
LIBERO	Object	0.56	0.48	96.7	96.9
	Goal	1.58	1.20	91.2	92.8
	Spatial	0.71	0.67	95.7	96.3
	Long	1.38	1.13	92.4	93.8
	Average	1.06	0.87	94.0	94.9
CALVIN	ABC→D	1.25	3.49	92.8	65.9

A.6 Complete Action-Conditioning Results

CALVIN. Table 9 reports future geometry with and without action conditioning. Both variants use the same backbone, geometric queries, physical-consistency closure, training data, evaluation protocol, and prediction horizon $T = 8$. They differ only in whether the future branch receives the policy-generated action chunk. Action conditioning improves all reported metrics, with larger relative reductions in object-level errors than in scene-level errors.

Table 9: Effect of action conditioning on future depth-derived geometry on CALVIN. Both variants use the same evaluation protocol and prediction horizon of $T = 8$.

Metric	ForeAct3D	w/o Action Conditioning
AbsRel ↓	0.0117	0.0165
RMSE (m) ↓	0.1633	0.2144
δ_1 ↑	0.9933	0.9881
Scene CD (m) ↓	0.0206	0.0220
Scene EMD (m) ↓	0.0623	0.0655
Inst-CD (m) ↓	0.0137	0.0183
Trans (cm) ↓	1.25	1.77
Recall@2cm ↑	0.928	0.799

LIBERO. Table 10 reports depth and instance-level metrics for the Goal, Object, Spatial, and Long suites, with the Long suite denoted as LIBERO-10 in the detailed results. Action conditioning improves every reported metric across all four suites. The LIBERO values in Table 4(b) are equally

weighted averages of the suite-level results, computed before rounding. These per-suite results complement the aggregate comparison across manipulation settings.

Table 10: **Action-conditioning results by LIBERO suite.** Values are measured at the future prediction horizon under the same GT-mask protocol as Table 8. Full denotes ForeAct3D with action conditioning; w/o action denotes the variant without it. The main-text instance metrics are equal-weight averages of the four suites, computed before rounding.

Suite	Variant	AbsRel ↓	RMSE ↓ (m)	δ_1 ↑	Inst-CD ↓ (m)	Trans ↓ (cm)	Recall ↑ @2cm
Goal	Full	0.0225	0.214	0.976	0.0101	1.58	0.912
	w/o action	0.0562	0.315	0.945	0.0126	2.65	0.879
Object	Full	0.0078	0.067	0.994	0.0068	0.56	0.967
	w/o action	0.0388	0.216	0.958	0.0115	2.82	0.908
Spatial	Full	0.0229	0.230	0.973	0.0076	0.71	0.957
	w/o action	0.0769	0.378	0.929	0.0123	2.33	0.880
LIBERO-10	Full	0.0173	0.095	0.987	0.0105	1.38	0.924
	w/o action	0.0598	0.224	0.951	0.0133	2.55	0.869

A.7 Limitations and Future Directions

ForeAct3D assumes static backgrounds and rigid instances, limiting applicability to deformable objects, articulated instances, and moving backgrounds. Same-pixel instance membership approximates temporal correspondence without tracking material points and becomes less reliable under large motion. Training requires dense geometric supervision and instance annotations, making predictions sensitive to depth and mask errors. Broader task and robot coverage are needed to assess generalization and variability. Geometric comparisons use different prediction horizons, and predicted-mask localization retains ground-truth instance IDs. Future prediction provides training supervision but is unused at inference. Future work could reduce supervision, extend physical constraints to general scenes, and use predicted futures to evaluate or refine actions.