OASIS: Observation-Action Space Alignment via SE(3) Trajectory Prediction for Robotic Manipulation

Xinzhe Chen* **Sihua Ren*** **Liqi Huang** **Haowen Sun** **Mingyang Li**
Zeyang Liu **Xingyu Chen†** **Xuguang Lan†**
National Key Laboratory of Human-Machine Hybrid Augmented Intelligence
Institute of Artificial Intelligence and Robotics
Xi'an Jiaotong University, Xi'an, China

Abstract

Recent vision-language-action (VLA) models and world action models (WAMs) advance robotic manipulation by enriching intermediate representations with auxiliary spatial features or future visual-state prediction. However, these representations largely remain within the observation space and do not share the rigid-body geometry of the action space, forcing the action decoder to implicitly recover this geometry. We propose OASIS, a visuomotor policy that aligns the intermediate representation with the action space via $SE(3)$ end-effector trajectory prediction. OASIS couples a 3D-aware feature encoder that fuses vision-language and metric-depth features with an $SE(3)$ trajectory predictor that produces a camera-frame end-effector trajectory. Conditioned on the predictor's pose-supervised hidden states, the action decoder generates action chunks consistent with rigid-body motion. Across simulation and real-world experiments, OASIS outperforms VLA and WAM baselines in success rate and out-of-distribution generalization. Our project page is available at `https://oasis-neurips.github.io`.

1 Introduction

Visuomotor policies for robotic manipulation use neural networks to map image observations, language instructions, and the robot state directly to executable 6-DoF actions and gripper commands [7, 11, 29, 39, 61, 67], and three representative families of methods define the current frontier. Vanilla Vision-Language-Action (VLA) models [3, 26, 27] build multimodal features from image observations and language instructions, typically via a pretrained Vision-Language Model (VLM), and feed these features directly into an action decoder [5, 22]. Feature-enhanced VLAs [19, 40, 41, 52] strengthen spatial understanding by injecting spatial features such as depth maps [28, 64], regions of interest [45], or 2D motion trajectories on the image plane [18, 23]. World Action Models (WAMs) [10, 13, 53] take a different route, learning a world model that predicts future images [17, 60] or latent visual features [33, 48, 51].

Despite their empirical strength, these families decode 6-DoF relative actions from intermediate representations that do not share the rigid-body geometry of the action space. This mismatch matters because rigid-body motions of a robot end-effector live in the Special Euclidean group $SE(3)$, yet a single image observation is consistent with many distinct 6-DoF actions under the same task instruction. Resolving this ambiguity requires an intermediate representation that constrains the target $SE(3)$ pose. Large-scale robotic pretraining or auxiliary spatial supervision can mitigate this ambiguity, but neither injects the geometric structure of $SE(3)$ into the intermediate.

*Equal contribution.
†Corresponding authors: `xglan@mail.xjtu.edu.cn` and `chenxingyu1990@xjtu.edu.cn`.

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

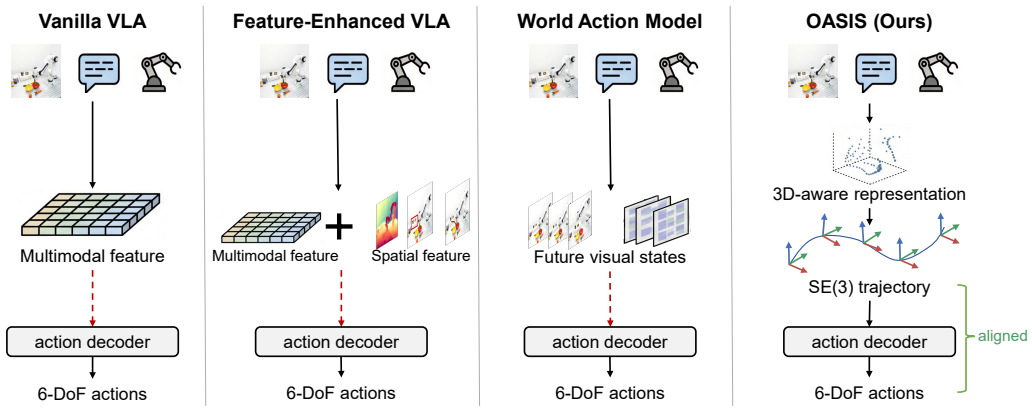


Figure 1: Comparison of existing visuomotor policies and OASIS. VLA models and WAMs construct intermediate representations that are not aligned with the action space. OASIS instead aligns the intermediate representation with the action space via $SE(3)$ trajectory prediction.

Our key insight is that predicting an $SE(3)$ end-effector trajectory aligns the intermediate representation with the action space, providing a geometric inductive bias rather than enriching observation-space features with auxiliary spatial cues or predictive visual capacity. Figure 1 contrasts this design with existing intermediates, and Section 3.2 formalizes it. An intermediate representation is geometrically aligned when it provides an explicit readout of the target $SE(3)$ pose at each future step. Auxiliary supervision via future images, latent visual features, or 2D trajectories leaves pose recovery implicit in the decoder. In contrast, $SE(3)$ end-effector trajectory prediction supervises the intermediate against the rigid-body geometry of the action space, so the decoder receives explicit pose information for action generation.

We instantiate this insight as **OASIS** (**O**bservation-**A**ction **S**pace Al**I**gnment via **SE**(3) Trajectory Prediction), which couples three components trained end-to-end. A 3D-aware feature encoder fuses vision-language and metric-depth features into a 3D-aware representation. On this representation, an $SE(3)$ trajectory predictor forecasts a sequence of end-effector poses in the camera frame. Sharing the frame of the representation spares the predictor from learning the camera-to-world extrinsic. The extrinsic is used only to build camera-frame training labels, so OASIS needs no extrinsic input at inference. The action decoder attends to the predictor's pose-supervised hidden states and the current end-effector state, and generates the action chunk while absorbing residuals such as the frame conversion and the gripper-timing offset. Supervision comes solely from standard expert demonstrations, so OASIS requires neither annotated spatial labels nor pretraining on large-scale robotic demonstrations, and uses only generically pretrained backbones.

Extensive experiments show that OASIS improves performance both on simulation benchmarks and on real robots. OASIS achieves a **97.6%** average success rate on LIBERO [32] and a **4.57** average sequence length on CALVIN ABC→D [36]. On two real robots, it reaches an 89.2% average success rate, outperforming $\pi_{0.5}$ [5], RDT [35], Seer-Large [48], and ACT [63]. Under out-of-distribution (OOD) perturbations in multi-task settings, covering unseen backgrounds, a displaced third-person camera, and human interference, OASIS attains a 90.8% average success rate and loses less than $\pi_{0.5}$ under each perturbation. Adding the $SE(3)$ trajectory predictor to the encoder and decoder raises success from 89.5% to 95.2% on LIBERO-Long, the largest *additional* gain among ablated components. Our contributions are summarized as follows.

- We formulate an aligned-intermediate design principle for visuomotor policies, identifying $SE(3)$ end-effector trajectory prediction as the mechanism that aligns the intermediate representation with the action space, providing a geometric inductive bias for action generation.
- We construct OASIS, a visuomotor policy coupling a 3D-aware feature encoder, an $SE(3)$ trajectory predictor, and an action decoder, trained end-to-end on standard expert demonstrations without spatial annotations or large-scale robotic pretraining.
- Across simulation and real-world settings, OASIS achieves the highest average performance among the compared methods and degrades less than $\pi_{0.5}$ under OOD perturbations, with ablations attributing the largest additional gain to the $SE(3)$ trajectory predictor.

2 Related Work

Vanilla Vision-Language-Action Models. Vanilla VLA models map image observations and language instructions directly to 6-DoF actions through a multimodal encoder and a dedicated action decoder [8, 55]. Early work generates action chunks with autoregressive [47] or denoising action heads [37] such as ACT [63] and Diffusion Policy [12], followed by transformer-based variants [42, 43] including RDT-1B [35]. More recent methods leverage pretrained vision-language models [3, 27] for stronger semantic priors, including OpenVLA [22] on a Llama VLM [49] and π_0 [5, 6] on a PaliGemma VLM [2] with flow matching [31]. The action decoder must recover depth and rotation from multimodal features alone, without a geometrically aligned intermediate.

Feature-Enhanced VLA Models. Feature-enhanced VLA models improve this image-to-action mapping by injecting auxiliary spatial features. One line lifts observations into 3D space [19, 24, 25, 56, 64]. SpatialVLA [40] encodes 3D position information via Ego3D position encoding, and QDepth-VLA [28] introduces quantized depth as auxiliary supervision. A complementary line provides task-relevant visual guidance [23, 52, 65]. ReconVLA [45] reconstructs gaze regions to focus attention on target objects, ThinkAct [18] predicts 2D end-effector trajectories, and UniVLA [9] derives task-centric action representations from videos. These features strengthen spatial reasoning and task understanding, yet the resulting intermediates do not share the rigid-body geometry of the action space, leaving the decoder to bridge them to actions. Structured-pose policies [14, 15, 21, 44] instead treat $SE(3)$ poses as the final action representation, predicted from 3D inputs that require additional sensors or multi-view reconstruction. OASIS, in contrast, predicts an $SE(3)$ trajectory from RGB images as the intermediate that conditions a learned action decoder.

World Action Models. World Action Models predict future images or latent visual features and decode actions from these predicted targets [53, 59]. SuSIE [4] synthesizes subgoal images via image editing as visual guidance. VPP [17] uses video diffusion models to predict future visual states as intermediate representations for action generation, and Seer [48] infers actions via inverse dynamics on predicted latent visual features. Related WAMs explore further couplings between visual prediction and action decoding [58, 60, 62, 66]. WorldVLA [10] formulates an autoregressive action-world model, and Unified-VLA [51] further unifies perception grounding, vision-supervised world modeling, and action generation within a single framework. Although these predictive intermediates make future visual-state structure explicit, they live on the image plane rather than sharing the rigid-body geometry of the action space, leaving the decoder to implicitly recover the rigid-body motion from these visual predictions.

3 Problem Formulation and Analysis

3.1 Problem Formulation

At time step t, a visuomotor policy π receives image observations $\mathbf{o}_t$, a language instruction l, and the current end-effector state $\mathbf{e}_t = [\mathbf{p}_t, \boldsymbol{\theta}_t]^\top$, with position $\mathbf{p}_t \in \mathbb{R}^3$ and rotation parameterization $\boldsymbol{\theta}_t$. Equivalently, $\mathbf{e}_t$ specifies a world-frame homogeneous transformation,

$$\mathbf{T}_t = \begin{bmatrix} \mathbf{R}(\boldsymbol{\theta}_t) & \mathbf{p}_t \\ \mathbf{0}^\top & 1 \end{bmatrix} \in SE(3), \tag{1}$$

where $\mathbf{R}(\boldsymbol{\theta}_t) \in SO(3)$ is the corresponding rotation matrix.

Let $\{\mathbf{T}_{t+h}\}_{h=1}^{H}$ denote a world-frame target pose sequence over a prediction horizon of H steps along a successful execution of the manipulation task. The policy must produce an action chunk $\mathbf{A}_t = \{(\mathbf{a}_{t+h-1}, g_{t+h-1})\}_{h=1}^{H}$ of the same length that drives the end-effector through this pose sequence, where $\mathbf{a}_{t+h-1}$ is a 6-DoF relative end-effector action and g_{t+h-1} is the gripper command. For each horizon index h, the relative action is recovered in closed form as

$$\mathbf{a}_{t+h-1} = \rho\big(\mathbf{T}_{t+h-1}^{-1}\mathbf{T}_{t+h}\big), \tag{2}$$

where $\rho : SE(3) \to \mathbb{R}^6$ parameterizes a relative transformation as translation and rotation vectors, such as the axis-angle representation. Equation (2) converts any target pose sequence into executable 6-DoF relative actions. The gripper command is modeled separately because opening and closing the gripper is not a rigid-body motion in $SE(3)$.

3.2 Aligned Intermediate Representations of Visuomotor Policies

We model a visuomotor policy as a sequence of representation and action-decoding stages and adopt the following design principle. A horizon-indexed intermediate $\mathbf{m}_t = \{\mathbf{m}_{t+h}\}_{h=1}^{H}$ is *geometrically aligned* with the action space if it provides an explicit pose readout $r(\mathbf{m}_{t+h}) = \mathbf{g} \cdot \mathbf{T}_{t+h} \in SE(3)$ at each horizon step, where $\mathbf{g} \in SE(3)$ is a fixed rigid transform that may be unknown. Given such a readout, the decoder can approximate the relative-action recovery of Eq. (2) while absorbing $\mathbf{g}$, prediction noise, contact dynamics, and gripper timing as residuals. We treat this as a design principle rather than a closed-form sufficiency theorem, since successful execution still depends on the residual control problem that the decoder must learn.

Vision-Language-Action (VLA) models, including feature-enhanced variants that inject auxiliary spatial features, encode the observation, instruction, and end-effector state into a latent multimodal feature and decode the action chunk directly,

$$\mathbf{z}_t = E_{\mathrm{VLA}}(\mathbf{o}_t, l, \mathbf{e}_t), \qquad \mathbf{A}_t = D_{\mathrm{VLA}}(\mathbf{z}_t). \tag{3}$$

Because $\mathbf{z}_t$ contains no explicit horizon-indexed pose readout, D_{VLA} must implicitly infer both the target poses $\{\mathbf{T}_{t+h}\}_{h=1}^{H}$ and the parameterization ρ to realize Eq. (2). Pose recovery and action parameterization are therefore coupled inside a single learned action decoder.

WAMs insert a predictive intermediate representation before action decoding,

$$\mathbf{u}_t = F_{\mathrm{WAM}}(\mathbf{o}_t, l, \mathbf{e}_t) \in \mathcal{P}^H, \qquad \mathbf{A}_t = D_{\mathrm{WAM}}(\mathbf{u}_t, \mathbf{e}_t), \tag{4}$$

where $\mathcal{P}$ denotes an intermediate space such as future images or latent visual features. Although $\mathbf{u}_t$ is horizon-indexed, $\mathcal{P}$ does not share the rigid-body geometry of the action space. The decoder D_{WAM} must therefore implicitly recover $\mathbf{T}_{t+h}$ from $\mathbf{u}_t$ before applying Eq. (2). WAMs thus expose temporal structure but leave the action-recovery step without geometric guidance.

OASIS instantiates this design principle by predicting a camera-frame $SE(3)$ end-effector trajectory $\{\mathbf{T}^c_{t+h}\}_{h=1}^{H}$. This predicted trajectory satisfies the geometric-alignment property with $\mathbf{g} = \mathbf{T}^{-1}_{c\to w}$, the inverse of the camera-to-world extrinsic, since $\mathbf{T}^c_{t+h} = \mathbf{T}^{-1}_{c\to w} \cdot \mathbf{T}_{t+h}$. The decoder therefore receives explicit pose information for action generation up to this fixed rigid transform, and learns to absorb frame conversion, prediction noise, contact dynamics, and gripper timing as residuals.

4 Methodology

4.1 Overview of OASIS

OASIS instantiates the aligned-intermediate design principle from Section 3.2 with a three-stage policy architecture, illustrated in Figure 2. Since robotic manipulation acts in 3D physical space while image observations alone are 2D projections that lack metric depth, the 3D-aware feature encoder merges vision-language and metric-depth features into a 3D-aware representation $\mathbf{h}_t^{3D}$,

$$\mathbf{h}_t^{3D} = E_{3D}(\mathbf{o}_t, l). \tag{5}$$

Conditioned on $\mathbf{h}_t^{3D}$, the trajectory predictor produces pose-supervised hidden states $\mathbf{h}_{\mathrm{traj}}$, which a linear head projects to a horizon-H camera-frame end-effector $SE(3)$ trajectory $\mathcal{T}_t^c$. This trajectory prediction aligns the predictor's intermediate representation with the action space,

$$\mathbf{h}_{\mathrm{traj}} = P_{SE(3)}(\mathbf{h}_t^{3D}), \qquad \mathcal{T}_t^c = \{\mathbf{T}^c_{t+h} \in SE(3)\}_{h=1}^{H} = \mathrm{Linear}(\mathbf{h}_{\mathrm{traj}}). \tag{6}$$

Predicting in the camera frame keeps the intermediate in the same frame as $\mathbf{h}_t^{3D}$ and spares the predictor from learning the camera-to-world extrinsic. The action decoder, conditioned on these pose-supervised hidden states $\mathbf{h}_{\mathrm{traj}}$ and the current end-effector state $\mathbf{e}_t$, learns to approximate the relative-action recovery of Eq. (2) and emit gripper commands,

$$\mathbf{A}_t = D_{\mathrm{act}}(\mathbf{h}_{\mathrm{traj}}, \mathbf{e}_t). \tag{7}$$

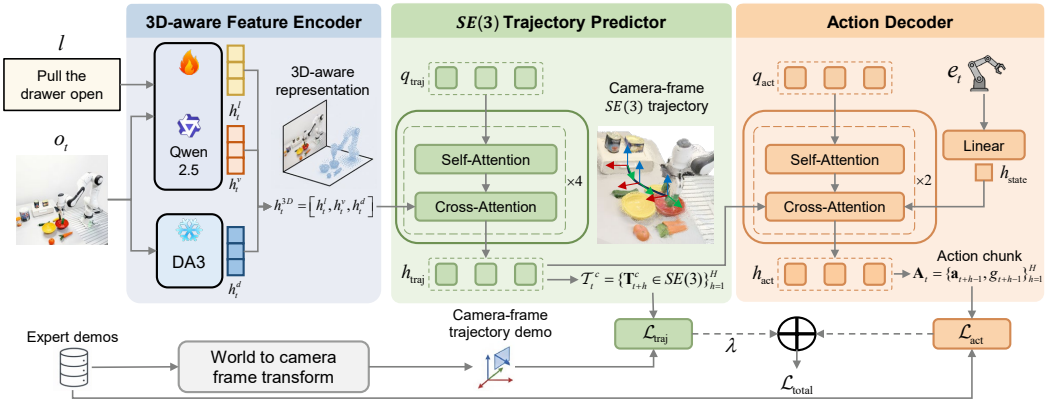


Figure 2: Architecture of OASIS. The 3D-aware feature encoder merges image, language, and metric-depth features into a 3D-aware representation. The $SE(3)$ trajectory predictor aligns the intermediate representation with the action space by predicting end-effector poses. The action decoder generates executable action chunks conditioned on this aligned intermediate representation.

4.2 3D-aware Feature Encoder

The encoder provides the predictor with a 3D-aware representation that preserves the language-conditioned visual information needed to forecast task-relevant target poses.

Metric-depth feature. To supply the calibrated 3D structure that image observations lack, we use Depth Anything 3 [30], specifically the frozen DA3METRIC-LARGE model, which extracts multi-scale metric-depth features from $\mathbf{o}_t$. We project these features into N_d depth hidden states $\mathbf{h}_t^d \in \mathbb{R}^{N_d \times D}$, where D is the hidden dimension shared by the encoder, predictor, and decoder.

Vision-language feature. We use a VLM with a Qwen2.5-0.5B backbone [1, 20] that jointly processes $\mathbf{o}_t$ and l, producing N_v visual hidden states $\mathbf{h}_t^v \in \mathbb{R}^{N_v \times D}$ and N_l language hidden states $\mathbf{h}_t^l \in \mathbb{R}^{N_l \times D}$. We concatenate $\mathbf{h}_t^l$, $\mathbf{h}_t^v$, and $\mathbf{h}_t^d$ into the 3D-aware representation of Eq. (5),

$$\mathbf{h}_t^{3D} = [\mathbf{h}_t^l, \mathbf{h}_t^v, \mathbf{h}_t^d]. \tag{8}$$

4.3 $SE(3)$ Trajectory Predictor

The trajectory predictor aligns its intermediate representation with the action space by predicting the trajectory $\mathcal{T}_t^c$ of Eq. (6), where each $\mathbf{T}_{t+h}^c \in SE(3)$ is the predicted end-effector pose at horizon step h. A linear head projects the predictor's hidden states $\mathbf{h}_{\text{traj}}$ to this trajectory. These pose-supervised hidden states serve as the conditioning signal passed to the action decoder.

Architecture. The trajectory predictor initializes H learnable trajectory queries $\mathbf{q}_{\text{traj}} \in \mathbb{R}^{H \times D}$. Stacked transformer blocks process these queries with self-attention and Rotary Position Embeddings (RoPE) [46] for temporal coherence, while cross-attention conditions each query on $\mathbf{h}_t^{3D}$. The resulting trajectory hidden states $\mathbf{h}_{\text{traj}} \in \mathbb{R}^{H \times D}$ are projected to per-step pose vectors

$$\boldsymbol{\tau}_t^c = \{\mathbf{e}_{t+h}^c = [\mathbf{p}_{t+h}^c, \boldsymbol{\theta}_{t+h}^c]^\top\}_{h=1}^H, \tag{9}$$

where $\mathbf{p}_{t+h}^c \in \mathbb{R}^3$ is the position and $\boldsymbol{\theta}_{t+h}^c \in \mathbb{R}^3$ is an axis-angle vector. Axis-angle is the canonical exponential-coordinate chart of $SO(3)$. Every prediction $\boldsymbol{\theta}_{t+h}^c$ yields a valid rotation matrix $\mathbf{R}(\boldsymbol{\theta}_{t+h}^c) = \exp([\boldsymbol{\theta}_{t+h}^c]_\times) \in SO(3)$ in closed form via the Rodrigues rotation formula, where $[\cdot]_\times$ maps a vector to its skew-symmetric matrix, so each pose vector induces a valid $SE(3)$ element

$$\mathbf{T}_{t+h}^c = \begin{bmatrix} \mathbf{R}(\boldsymbol{\theta}_{t+h}^c) & \mathbf{p}_{t+h}^c \\ \mathbf{0}^\top & 1 \end{bmatrix} \in SE(3) \tag{10}$$

by construction, requiring no projection, Gram-Schmidt orthogonalization, or post-hoc correction. For the bounded range of camera-frame end-effector orientations encountered in tabletop manipulation we further have $\|\boldsymbol{\theta}_{t+h}^c\| < \pi$, keeping all predictions inside the canonical axis-angle chart of $SO(3)$ and the network output on the $SE(3)$ manifold throughout training and inference.

Training objective. For computational efficiency, we supervise the local pose parameterization with ground-truth pose vectors $\hat{\boldsymbol{\tau}}_t^c = \{\hat{\mathbf{e}}_{t+h}^c\}_{h=1}^H$ obtained by transforming world-frame demonstrations into the camera frame, and adopt the trajectory loss

$$\mathcal{L}_{\text{traj}} = \frac{1}{H}\sum_{h=1}^{H} \left\|\hat{\mathbf{e}}_{t+h}^c - \mathbf{e}_{t+h}^c\right\|_1 . \tag{11}$$

We choose ℓ_1 because it is robust to teleoperation artifacts and yields precise pose modeling [63]. Empirically, axis-angle outperforms the quaternion and Euler parameterizations under matched training on LIBERO-Long, as Figure 6 shows. In our setting, target and predicted absolute future orientations remain within the canonical axis-angle chart, away from the $\|\boldsymbol{\theta}_{t+h}^c\| = \pi$ singular set, so axis-angle coordinates are well-defined throughout training and inference.

4.4 Action Decoder

The action decoder cross-attends to the predictor's pose-supervised hidden states and the current end-effector state, approximating the relative-action recovery of Eq. (2) while absorbing frame conversion, prediction noise, contact dynamics, and gripper timing as residuals. We retain a learned decoder rather than a closed-form pose-to-action converter because these residuals exceed what explicit pose readouts can capture, as detailed in Appendix B.3.

Architecture. The action decoder initializes H learnable action queries $\mathbf{q}_{\text{act}} \in \mathbb{R}^{H\times D}$. These queries attend to two sources of context, namely the trajectory hidden states $\mathbf{h}_{\text{traj}}$ from the predictor and a state embedding $\mathbf{h}_{\text{state}} \in \mathbb{R}^{1\times D}$ obtained by linearly projecting the current end-effector state $\mathbf{e}_t$. The two are concatenated into context hidden states $\mathbf{h}_{\text{ctx}} = [\mathbf{h}_{\text{traj}}, \mathbf{h}_{\text{state}}] \in \mathbb{R}^{(H+1)\times D}$. The decoder adds no encoding of the order in which it receives the H trajectory hidden states, so its cross-attention over them is permutation-invariant. The decoder then outputs the action chunk $\mathbf{A}_t = \{(\mathbf{a}_{t+h-1}, g_{t+h-1})\}_{h=1}^H$, where $\mathbf{a}_{t+h-1} = [\Delta\mathbf{p}_{t+h-1}, \Delta\boldsymbol{\theta}_{t+h-1}]^\top$ is the 6-DoF relative action and g_{t+h-1} is the gripper command.

Training objective. The action loss supervises the output action chunk against expert demonstrations,

$$\mathcal{L}_{\text{act}} = \frac{1}{H}\sum_{h=1}^{H} \left\|[\hat{\mathbf{a}}_{t+h-1}, \hat{g}_{t+h-1}] - [\mathbf{a}_{t+h-1}, g_{t+h-1}]\right\|_1 . \tag{12}$$

The total objective combines trajectory supervision with executable-action supervision,

$$\mathcal{L}_{\text{total}} = \lambda\mathcal{L}_{\text{traj}} + \mathcal{L}_{\text{act}}, \tag{13}$$

where λ balances the two losses and is set to 0.1.

5 Experiments

We evaluate OASIS in simulation and on real robots to answer four questions. First, does OASIS surpass VLA and WAM baselines on standard manipulation suites? Second, is the metric-depth feature of the 3D-aware feature encoder necessary? Third, how do the supervision target of the predictor's hidden states and its reference frame affect policy success? Fourth, does OASIS's lead transfer to real robots across multi-task suites and under OOD perturbations?

5.1 Implementation Details

The trajectory predictor and action decoder share a transformer architecture with four and two blocks respectively. The predictor outputs an eight-step $SE(3)$ end-effector trajectory in the camera frame, and the decoder generates the corresponding action chunk conditioned on the trajectory hidden states. OASIS has 1.73B total parameters, of which 0.18B are trainable. We LoRA-tune [16] the Qwen2.5-0.5B VLM, train the predictor, decoder, and projection layers from scratch without large-scale robotic pretraining, and freeze DA3METRIC-LARGE [30]. Experiments use four NVIDIA A800 GPUs with batch size 64. Appendix A provides further details.

Table 1: Success rates in percent on the four LIBERO suites. The Pretrain column marks pre-training on large-scale robotic datasets, and the Intermediate column lists the representation each method feeds into the action decoder. The highest success rate in each suite is in bold. OASIS reaches the highest average success at 97.6 without any pre-training on large-scale robotic datasets.

Method	Intermediate	Pretrain	Spatial	Object	Goal	Long	Average
SpatialVLA [40]	spatial features	✓	88.2	89.9	78.6	55.5	78.1
WorldVLA [10]	future visual states	×	85.6	89.0	82.6	59.0	79.1
ThinkAct [18]	2D-supervised features	✓	88.3	91.4	87.1	70.9	84.4
π_0 [6]	multimodal features	✓	96.8	**98.8**	95.8	85.2	94.2
QDepth-VLA [28]	spatial features	✓	97.6	96.6	95.2	90.0	94.9
UniVLA [9]	spatial features	✓	96.5	96.8	95.6	92.0	95.2
Unified-VLA [51]	future visual states	✓	95.4	**98.8**	93.6	94.0	95.5
OASIS (Ours)	SE(3)-supervised features	×	**99.0**	**98.8**	**97.4**	**95.2**	**97.6**

Table 2: Results on CALVIN ABC→D. The Pretrain column marks pre-training on large-scale robotic datasets. Tasks completed in a row measures the success rate in percent of sequentially executing 1 to 5 instructions without scene resets. Avg. is the average sequence length, the mean number of instructions completed in a row. † uses multi-view RGB-D. OASIS attains the highest average sequence length at 4.57, with its largest gains at the fourth and fifth instructions.

Method	Intermediate	Pretrain	Tasks Completed in a Row					Avg.
			1	2	3	4	5	
SuSIE [4]	future visual states	×	87.0	69.0	49.0	38.0	26.0	2.69
3D Diffuser Actor† [21]	3D feature	×	93.8	80.3	66.2	53.3	41.2	3.35
ReconVLA [45]	spatial features	✓	95.6	87.6	76.9	69.3	64.1	3.95
Seer-Large [48]	future visual states	✓	96.3	91.6	86.1	80.3	74.0	4.28
VPP [17]	future visual states	✓	96.5	90.9	86.6	82.0	76.9	4.33
Unified-VLA [51]	future visual states	✓	**98.9**	94.8	89.0	82.8	75.1	4.41
DreamVLA [60]	future visual states	✓	98.2	94.6	89.5	83.4	78.1	4.44
OASIS (Ours)	SE(3)-supervised features	×	98.1	**94.9**	**91.7**	**88.9**	**83.3**	**4.57**

5.2 Simulation Experiments

Simulation setup. To answer the first question, we evaluate OASIS on two widely adopted simulation benchmarks, LIBERO [32] and CALVIN [36]. LIBERO has four task suites, namely Spatial, Object, Goal, and Long, which test spatial reasoning, object grounding, goal-directed behavior, and long-horizon planning. Each contains 10 tasks with 50 human-teleoperated demonstrations per task, and we evaluate every task over 50 episodes. For CALVIN, we use the challenging ABC→D setting to test OOD generalization to an unseen environment D after training on environments A, B, and C, reporting the average sequence length over 1,000 distinct instruction chains, each of five consecutive subtasks. Detailed descriptions of the simulation environments appear in Appendix B.1.

Evaluation protocol. The evaluation protocol for LIBERO and CALVIN uses only RGB images, language, and robot state as input. We follow this protocol, train OASIS for 50k steps per suite, and average across three seeds. Baseline numbers come from the original papers. Their tasks and benchmark splits match ours, while evaluation counts, optimization, and model selection follow each source. We primarily compare against RGB-only baselines, and include 3D Diffuser Actor [21] as a structured-pose policy in Table 2, although it consumes calibrated multi-view RGB-D.

LIBERO simulation results. As reported in Table 1, OASIS reaches a 97.6% average success rate on LIBERO, leading the next-best baseline Unified-VLA [51] by 2.1 points. OASIS leads or ties on all four suites, with margins of 1.4 points on Spatial, 1.6 on Goal, and 1.2 on Long over the strongest prior result on each suite, and a tie on Object. Appendix B.5 lists the standard deviations, and even the worst Long seed reaches 94.8%, above Unified-VLA at 94.0%. Against the strongest model of each family, OASIS leads the vanilla VLA π_0 by 3.4 points, the feature-enhanced UniVLA by 2.4, and the WAM Unified-VLA by 2.1. The only other method without large-scale robotic pretraining, WorldVLA, reaches 79.1%. ThinkAct [18], evaluated under the same protocol, predicts a 2D image-plane trajectory, and OASIS leads it by 13.2 points on average and 24.3 on Long.

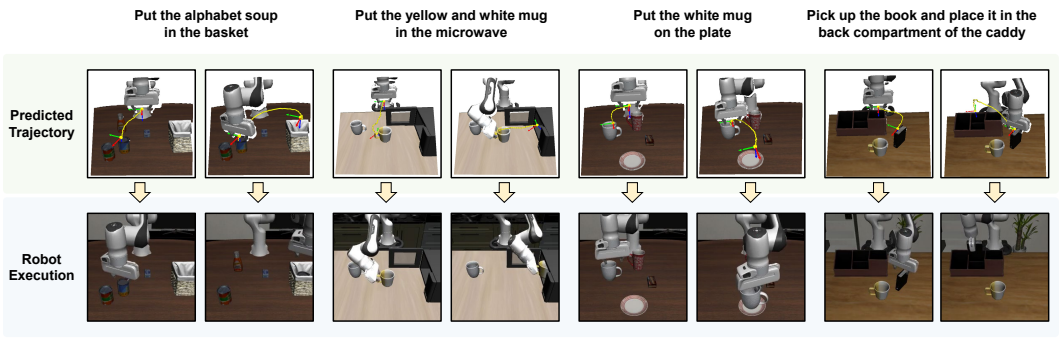


Figure 3: Predicted $SE(3)$ trajectories and robot execution on four LIBERO tasks. The top row overlays the predicted waypoints and rotation axes, and the bottom row shows the executed motion.

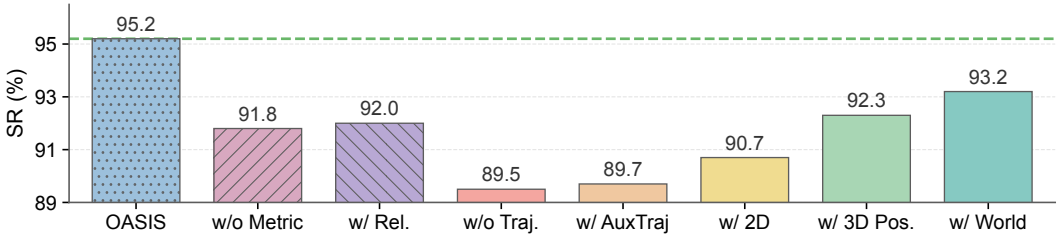


Figure 4: Ablation summary on LIBERO-Long. Success increases across the tested supervision ladder, and the trajectory predictor has the largest effect on success, as removing it costs 5.7 points.

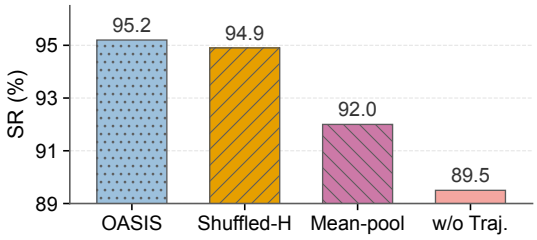


Figure 5: Decoder conditioning controls, averaged over three seeds on LIBERO-Long.

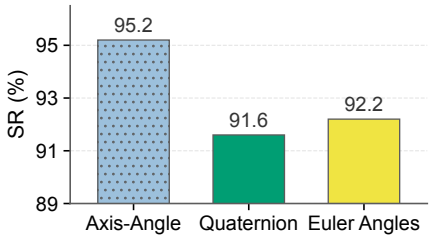


Figure 6: Ablation on choices of $SO(3)$ parameterization on LIBERO-Long.

CALVIN simulation results. As reported in Table 2, OASIS attains an average sequence length of 4.57 on CALVIN ABC→D, ahead of every baseline. It trails Unified-VLA and DreamVLA on the first instruction and leads from the second onward, and it exceeds the multi-view RGB-D policy 3D Diffuser Actor by 1.22 in average sequence length. The gap widens on the five-consecutive-task setting, where OASIS reaches 83.3% and leads the next-best baseline DreamVLA [60] by 5.2 points. This widening lead suggests that small per-step errors accumulate less under an intermediate aligned via $SE(3)$ trajectory prediction than under intermediates that live in the observation space.

Visualization of the $SE(3)$ trajectory. To validate the trajectory predictor qualitatively, we plot in Figure 3 the predicted translation waypoints and rotation axes alongside the executed end-effector path on four complex tasks. The executed path closely follows the predicted positions and orientations, indicating that the predicted trajectory is a consistent rigid-body prior for action generation.

5.3 Ablation Study

To answer questions two and three, we ablate the metric depth, the $SE(3)$ trajectory predictor, and the $SO(3)$ parameterization on LIBERO-Long. The first two are repeated on LIBERO-Spatial in Appendix B.2, where the ranking among variants is consistent.

Effectiveness of the 3D-aware feature encoder. The two hatched bars adjacent to the OASIS bar in Figure 4 ablate the metric-depth feature. The variant *w/o Metric*, which removes this feature from the

3D-aware representation, drops LIBERO-Long from 95.2% to 91.8%, and the variant *w/ Rel.*, which substitutes the relative depth of Depth Anything V2, recovers only marginally to 92.0%. Metric scale, not depth in general, is what helps the 3D-aware representation.

Impact of $SE(3)$ trajectory prediction. Figure 4 sweeps the supervision target on the pose-supervised hidden states under a matched-backbone control, with all variants sharing the encoder, depth model, and training budget. Richer geometric targets yield a monotonic increase on LIBERO-Long from 89.5% to 95.2%. The variant *w/o Traj.* drops the trajectory predictor and reaches 89.5%, marking the floor of the ladder. Since it keeps the same backbone and frozen depth model, *w/o Traj.* is itself a depth-enhanced VLA, and removing the trajectory predictor costs most among the ablated components, 5.7 points against 3.4 for removing metric depth. *w/ AuxTraj* retains the trajectory loss but routes it through a parallel predictor branch, so only $\mathbf{h}_t^{3D}$, not $\mathbf{h}_{\mathrm{traj}}$, conditions the decoder. It climbs only to 89.7%, showing that pose supervision helps only when the supervised hidden states reach the decoder. *w/ 2D* reaches 90.7%, barely above *w/o Traj.*, since image-plane geometry lacks the rigid-body structure of the action space. Adding depth lifts *w/ 3D Pos.* to 92.3%, and adding rotation lifts *w/ World* to 93.2%, 2.0 points below the camera-frame counterpart. The camera-frame $SE(3)$ trajectory attains the highest success, consistent with camera-frame prediction sparing the predictor from recovering the camera-to-world extrinsic.

How the decoder uses the trajectory hidden states. Figure 5 alters the trajectory hidden states of the trained OASIS model at evaluation time and compares it with *w/o Traj.* Since the decoder adds no encoding of their order, permuting them in *Shuffled-H* changes the actions only by summation-order noise, at most 7.8×10^{-3}, and success nearly matches OASIS. *Mean-pool* replaces the H states by their mean, keeping the trajectory-level mean but removing step-wise variation. Its 3.2-point drop shows that the decoder uses step-wise content, and its 2.5-point lead over *w/o Traj.* is consistent with useful trajectory-level conditioning. As Appendix B.4 and Table 11 detail, one attention head of the first decoder block puts 0.98 to 1.00 of each action step's weight on a single trajectory hidden state, and the last two steps read each other's states, so this one-to-one mapping follows content rather than position. From *w/ 2D* to OASIS, every variant carries a trajectory loss while success rises with the geometric richness of the target, so it is the step-wise content, not the trajectory loss itself, that drives success. Appendix B.3 further shows that replacing the learned decoder with a hardcoded pipeline, which converts the predicted poses into actions, collapses LIBERO-Long to 0.0%, confirming the decoder is necessary to absorb prediction noise, contact dynamics, and gripper timing.

Choice of $SO(3)$ parameterization. Figure 6 compares three $SO(3)$ parameterizations under matched training. Axis-angle reaches 95.2%, 3.0 points above Euler angles and 3.6 above quaternions. This ranking is consistent with axis-angle being a minimal, constraint-free chart that covers the bounded range of camera-frame end-effector orientations encountered in tabletop manipulation.

5.4 Real-world Experiments

Real-world setup. To answer the fourth question, we evaluate OASIS on a Franka Research 3 and a Kinova Gen3, each with a third-person and a wrist camera, shown in Figure 7. Three suites probe instruction following, spatial reasoning, and long-horizon manipulation, denoted Goal, Spatial, and Long. OASIS and the compared baselines are trained on 50 teleoperated demonstrations per task and evaluated over three independent runs of 20 trials each, for 60 trials per task, exceeding the trial counts reported in the original baseline papers. All methods are trained for 50k steps and evaluated at the final checkpoint without hyperparameter search, and only $\pi_{0.5}$, RDT, and Seer-Large start from robotic-pretrained weights. OOD perturbations are further tested on the Goal suite under unseen backgrounds, a third-person camera displaced by about 15 cm, and human interference. Full settings and task descriptions appear in Appendix C.1.

Main results analysis. As reported in Table 3, OASIS reaches an 89.2% average success rate, including 98.6% on Goal, 85.8% on Spatial, and 83.3% on Long, leading $\pi_{0.5}$ [5] by 7.6 points, RDT [35] by 19.7, Seer-Large [48] by 30.8, and ACT [63] by 48.7, with the numerical lead holding across all three suites, although Appendix C.3 shows that the 95% Wilson intervals against $\pi_{0.5}$ overlap, widely on Long and slightly on Goal and Spatial. The numerical margin over $\pi_{0.5}$ is largest on Long, 11.7 points against 3.6 on Goal and 7.5 on Spatial.

OOD results analysis. As shown in Figure 8, OASIS holds 94.8%, 91.2%, and 86.5% across unseen backgrounds, displaced camera, and human interference, while $\pi_{0.5}$ reaches 88.6%, 76.4%, and

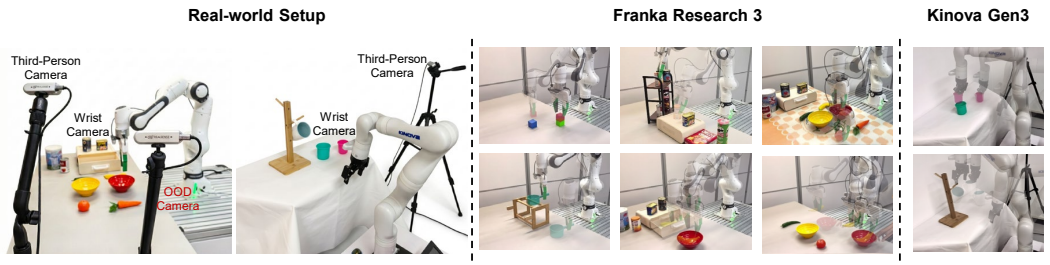


Figure 7: Real-world platforms and task examples. Each robot uses a third-person and a wrist camera, and OOD Camera marks where the third-person camera is moved for the displaced-camera test.

Table 3: Real-world success rates in percent on the Goal, Spatial, and Long suites, with each task evaluated over three independent runs of 20 trials.

Method	Goal	Spatial	Long	Average
ACT [63]	58.3	45.0	18.3	40.5
Seer-Large [48]	73.3	55.2	46.7	58.4
RDT [35]	81.7	66.7	60.0	69.5
$\pi_{0.5}$ [5]	95.0	78.3	71.6	81.6
OASIS (Ours)	**98.6**	**85.8**	**83.3**	**89.2**

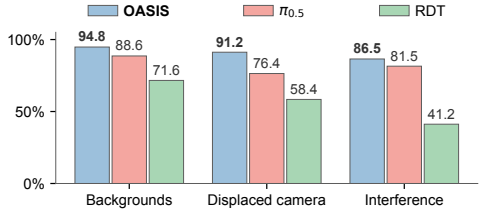


Figure 8: Success rates on the Goal suite in three OOD settings. OASIS loses less than $\pi_{0.5}$ under each of the three perturbations.

81.5%. Under unseen backgrounds, the metric-depth feature keys off scene geometry, so distractor objects and tabletops barely shift the decoder input. Under the displaced camera, only the third-person view moves, and because $\mathbf{h}_t^{3D}$ and $\mathcal{T}_t^c$ both live in the camera frame, the decoder tolerates this displacement somewhat. This probes robustness to a single uncalibrated viewpoint shift and does not establish camera-pose invariance. Under human interference, the predictor regenerates the trajectory for each chunk, so OASIS retargets within one action chunk when the bowl is moved mid-execution. The residual robustness of $\pi_{0.5}$ and RDT likely reflects their large-scale robotic pretraining.

Failures, speed, and data efficiency. A failure analysis in Appendix C.2 traces OASIS's residual errors to sub-centimeter placement and long-horizon compounding, and an error decomposition of five tasks shows that on each platform the lowest-success task among them has the most accurate predicted trajectory, pointing to execution tolerance rather than prediction error. One inference takes 45 ms per action chunk on one RTX 4090, against 120 ms for $\pi_{0.5}$ and 260 ms for RDT. A data-scaling study in the same appendix shows that on the real-world Long task, OASIS reaches 35.0% with only 10 demonstrations, the success rate that $\pi_{0.5}$ reaches with 25, a low-data advantage on this task.

6 Conclusion

Existing visuomotor policies decode 6-DoF actions from intermediate representations that do not share the rigid-body geometry of the action space. We address this gap by aligning the intermediate representation with the action space via camera-frame $SE(3)$ end-effector trajectory prediction, instantiated as OASIS, whose action decoder cross-attends to the predictor's pose-supervised hidden states. Across simulation benchmarks and real-world platforms, OASIS attains superior manipulation precision and generalization, suggesting that aligning the intermediate is a more direct lever for visuomotor policy performance than enriching observation-space features.

Limitations and future work. Our study is limited to tabletop manipulation with a single fixed-base arm, and OASIS has not been validated on bimanual or mobile manipulation tasks. Since the $SE(3)$ intermediate aligns with the action space of a single end-effector, extending this design principle to richer action spaces is a natural next step, including coupled pose manifolds for bimanual manipulation, coupled $SE(2) \times SE(3)$ trajectories for mobile manipulation, and contact-conditioned trajectories for dexterous hands. Future work will also examine the choice between regression and generative trajectory prediction, such as diffusion or flow matching, which the alignment principle equally admits since it concerns what the intermediate represents rather than how it is modeled.

Acknowledgments

This work was supported in part by the NSFC under Grant No. 62125305, the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant No. JYB2025XDXM210, the NSFC under Grant Nos. U23A20339, 62573339, and 62503380, the Natural Science Foundation of Shaanxi Province under Grant No. 2025SYSSYSZD-083, the State Grid Corporation of China Science and Technology Projects 52060025005B-439-ZN, and the Shaanxi Provincial Key R&D Program under Grant No. 2026JC-ZDGG-17.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[2] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

[3] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[4] Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. *arXiv preprint arXiv:2310.10639*, 2023.

[5] Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Robert Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, brian ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 17–40. PMLR, 27–30 Sep 2025.

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

[7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[8] Qingwen Bu, Hongyang Li, Li Chen, Jisong Cai, Jia Zeng, Heming Cui, Maoqing Yao, and Yu Qiao. Towards synergistic, generalized, and efficient dual-system for robotic manipulation. *arXiv preprint arXiv:2410.08001*, 2024.

[9] Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to Act Anywhere with Task-centric Latent Actions. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025.

[10] 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.

[11] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

[12] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[13] 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.

[14] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. *arXiv preprint arXiv:2306.17817*, 2023.

[15] Ankit Goyal, Valts Blukis, Jie Xu, Yijie Guo, Yu-Wei Chao, and Dieter Fox. Rvt-2: Learning precise manipulation from few demonstrations. *arXiv preprint arXiv:2406.08545*, 2024.

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[17] Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video prediction policy: A generalist robot policy with predictive visual representations. In *Forty-second International Conference on Machine Learning*, 2025.

[18] Chi-Pin Huang, Yueh-Hua Wu, Min-Hung Chen, Yu-Chiang Frank Wang, and Fu-En Yang. Thinkact: Vision-language-action reasoning via reinforced visual latent planning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[19] Yueru Jia, Jiaming Liu, Sixiang Chen, Chenyang Gu, Zhilue Wang, Longzan Luo, Lily Lee, Pengwei Wang, Zhongyuan Wang, Renrui Zhang, et al. Lift3d foundation policy: Lifting 2d large-scale pretrained models for robust 3d robotic manipulation. *arXiv preprint arXiv:2411.18623*, 2024.

[20] Siddharth Karamcheti, Suraj Nair, Ashwin Balakrishna, Percy Liang, Thomas Kollar, and Dorsa Sadigh. Prismatic vlms: Investigating the design space of visually-conditioned language models. In *Forty-first International Conference on Machine Learning*, 2024.

[21] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *arXiv preprint arXiv:2402.10885*, 2024.

[22] 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.

[23] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

[24] Chengmeng Li, Junjie Wen, Yan Peng, Yaxin Peng, Feifei Feng, and Yichen Zhu. Pointvla: Injecting the 3d world into vision-language-action models. *arXiv preprint arXiv:2503.07511*, 2025.

[25] Peiyan Li, Yixiang Chen, Hongtao Wu, Xiao Ma, Xiangnan Wu, Yan Huang, Liang Wang, Tao Kong, and Tieniu Tan. BridgeVLA: Input-output alignment for efficient 3d manipulation learning with vision-language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[26] Xinghang Li, Peiyan Li, Minghuan Liu, Dong Wang, Jirong Liu, Bingyi Kang, Xiao Ma, Tao Kong, Hanbo Zhang, and Huaping Liu. Towards generalist robot policies: What matters in building vision-language-action models. *arXiv preprint arXiv:2412.14058*, 2024.

[27] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, Hang Li, and Tao Kong. Vision-language foundation models as effective robot imitators. In *The Twelfth International Conference on Learning Representations*, 2024.

[28] 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.

[29] Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *The Thirteenth International Conference on Learning Representations*, 2024.

[30] Haotong Lin, Sili Chen, Junhao Liew, Donny Y Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth anything 3: Recovering the visual space from any views. *arXiv preprint arXiv:2511.10647*, 2025.

[31] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[32] 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.

[33] Dongxiu Liu, Haoyi Niu, Zhihao Wang, Jinliang Zheng, Yinan Zheng, Zhonghong Ou, Jianming HU, Jianxiong Li, and Xianyuan Zhan. Efficient robotic policy learning via latent space backward planning. In *Forty-second International Conference on Machine Learning*, 2025.

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[35] 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. In *The Thirteenth International Conference on Learning Representations*, 2025.

[36] 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.

[37] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[38] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

[39] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[40] Delin Qu, Haoming Song, Qizhi Chen, Yuanqi Yao, Xinyi Ye, Yan Ding, Zhigang Wang, JiaYuan Gu, Bin Zhao, Dong Wang, et al. Spatialvla: Exploring spatial representations for visual-language-action model. *arXiv preprint arXiv:2501.15830*, 2025.

[41] Kanchana Ranasinghe, Xiang Li, E-Ro Nguyen, Cristina Mata, Jongwoo Park, and Michael S Ryoo. Pixel motion as universal representation for robot control. *arXiv preprint arXiv:2505.07817*, 2025.

[42] Moritz Reuss, Ömer Erdinç Yağmurlu, Fabian Wenzel, and Rudolf Lioutikov. Multimodal Diffusion Transformer: Learning Versatile Behavior from Multimodal Goals. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024.

[43] Moritz Reuss, Jyothish Pari, Pulkit Agrawal, and Rudolf Lioutikov. Efficient diffusion transformer policies with mixture of expert denoisers for multitask learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

[44] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023.

[45] Wenxuan Song, Ziyang Zhou, Han Zhao, Jiayi Chen, Pengxiang Ding, Haodong Yan, Yuxin Huang, Feilong Tang, Donglin Wang, and Haoang Li. Reconvla: Reconstructive vision-language-action model as effective robot perceiver. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 18549–18557, 2026.

[46] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[47] Yanchao Sun, Shuang Ma, Ratnesh Madaan, Rogerio Bonatti, Furong Huang, and Ashish Kapoor. SMART: Self-supervised multi-task pretraining with control transformers. In *International Conference on Learning Representations*, 2023.

[48] 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 *The Thirteenth International Conference on Learning Representations*, 2025.

[49] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[50] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

[51] Yuqi Wang, Xinghang Li, Wenxuan Wang, Junbo Zhang, Yingyan Li, Yuntao Chen, Xinlong Wang, and Zhaoxiang Zhang. Unified vision-language-action model. *arXiv preprint arXiv:2506.19850*, 2025.

[52] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning. *arXiv preprint arXiv:2401.00025*, 2023.

[53] Amber Xie, Oleh Rybkin, Dorsa Sadigh, and Chelsea Finn. Latent diffusion planning for imitation learning. *arXiv preprint arXiv:2504.16925*, 2025.

[54] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[55] Yang Yue, Yulin Wang, Bingyi Kang, Yizeng Han, Shenzhi Wang, Shiji Song, Jiashi Feng, and Gao Huang. Deer-VLA: Dynamic inference of multimodal large language models for efficient robot execution. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[56] Yanjie Ze, Gu Zhang, Kangning Zhang, Chenyuan Hu, Muhan Wang, and Huazhe Xu. 3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[57] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[58] Jianke Zhang, Yanjiang Guo, Yucheng Hu, Xiaoyu Chen, Xiang Zhu, and Jianyu Chen. UP-VLA: A unified understanding and prediction model for embodied agent. In *Forty-second International Conference on Machine Learning*, 2025.

[59] Kaidong Zhang, Pengzhen Ren, Bingqian Lin, Junfan Lin, Shikui Ma, Hang Xu, and Xiaodan Liang. Pivot-r: Primitive-driven waypoint-aware world model for robotic manipulation. *Advances in Neural Information Processing Systems*, 37:54105–54136, 2024.

[60] Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, Fan Lu, He Wang, et al. Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. *arXiv preprint arXiv:2507.04447*, 2025.

[61] Xinyu Zhang and Abdeslam Boularias. One-shot imitation learning with invariance matching for robotic manipulation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[62] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1702–1713, 2025.

[63] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[64] 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.

[65] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *The Thirteenth International Conference on Learning Representations*, 2025.

[66] Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

[67] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

A Implementation details

A.1 OASIS architecture

Given the real-time constraints of visuomotor policies, we prioritize a lightweight pretrained VLM over heavily parameterized alternatives. As Table 4 shows, the trajectory predictor and action decoder together comprise only 70 million parameters, keeping training and inference efficient.

Vision-language model. OASIS uses a VLM architecture derived from Prismatic VLM [20], with Qwen2.5-0.5B [1] as the language backbone and two visual encoders, DINOv2 [38] and SigLIP [57]. The VLM is pretrained on the LLaVA-1.5-Instruct dataset [34] and supplies the visual and language hidden states of the 3D-aware representation in Section 4.2.

Metric-depth module. OASIS uses the DA3METRIC-LARGE model of Depth Anything 3 [30] to supply metric depth. Unlike Depth Anything V2 [54] and VGGT [50], which produce normalized relative depth maps, this model estimates depth in physical units, the absolute scale that precise manipulation needs and that the metric-depth ablation in Section 5.3 isolates.

Trajectory predictor. The trajectory predictor is a 4-block Transformer with $H = 8$ learnable trajectory query tokens. Within each block, these queries perform self-attention with Rotary Position Embeddings, followed by cross-attention whose keys and values come from the 3D-aware representation $\mathbf{h}_t^{3D}$ of language, visual, and metric-depth hidden states. The predictor outputs the trajectory hidden states $\mathbf{h}_{\text{traj}}$, from which a linear head decodes the predicted $SE(3)$ trajectory. The hidden states condition the action decoder, and the decoded trajectory receives the loss of Section 4.3.

Action decoder. The action decoder adopts the same architecture as the trajectory predictor but with 2 blocks. Its cross-attention takes the trajectory hidden states and the embedding of the current end-effector state as keys and values, so the decoder generates actions conditioned on the trajectory hidden states rather than on the decoded trajectory, as Appendix B.4 examines. Table 5 summarizes the configurations of the trajectory predictor and action decoder.

Table 4: Full parameter breakdown of OASIS.

Component	**Total (B)**	**Trainable (B)**
DA3METRIC-LARGE	0.35	0.00 (Frozen)
Qwen2.5-0.5B VLM	1.30	0.10 (LoRA)
Trajectory predictor + action decoder	0.07	0.07
Linear projections + state embedding	0.01	0.01
Total	**1.73**	**0.18**

Table 5: Architecture and parameters of the trajectory predictor and action decoder.

Module	**Trajectory Predictor**	**Action Decoder**
Layers	4	2
Hidden Size	896	896
Attention Heads	8	8
Number of Queries (H)	8	8
Total Parameters	0.05B	0.02B

A.2 Training details

Training dataset. We use datasets in the standard RLDS format. Inputs consist of a language instruction, RGB images from the third-person and wrist views resized to $224 \times 224 \times 3$, and a 7-dimensional robot state comprising the end-effector position, rotation, and gripper state. The policy is supervised with 6-DoF relative actions and gripper commands. Trajectory labels are world-frame end-effector poses transformed into the third-person camera frame with its extrinsic matrix.

Table 6: Detailed training settings.

Setting	Value
GPUs	4 × NVIDIA A800
Global Batch Size	64 (16 per device)
Training Steps	50,000
Random Seeds	3 (results averaged)
Optimizer	AdamW
Fine-tuning Method	LoRA
Learning Rate	2×10^{-4}
Warmup Steps	5,000

Table 7: Example language instructions from the four LIBERO suites.

LIBERO	Task Instructions
Spatial	Pick up the black bowl from table center and place it on the plate. Pick up the black bowl on the stove and place it on the plate. ...
Object	Pick up the alphabet soup and place it in the basket. Pick up the chocolate pudding and place it in the basket. ...
Goal	Open the top drawer and put the bowl inside. Put the wine bottle on top of the cabinet. Turn on the stove. ...
Long	Put the white mug on the left plate and put the yellow and white mug on the right plate. Put the black bowl in the bottom drawer of the cabinet and close it. Put the yellow and white mug in the microwave and close it. ...

Training settings. We train OASIS with AdamW at a learning rate of 2×10^{-4} under a cosine annealing schedule with 5,000 warm-up steps. The vision-language model is fine-tuned with LoRA, the metric-depth module stays frozen, and the trajectory predictor and action decoder are trained with full-parameter updates. Table 6 lists the training settings.

B Additional details for simulation experiments

B.1 Simulation benchmarks and settings

LIBERO simulation benchmark. LIBERO comprises four task suites, LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long, which assess spatial reasoning, object grounding, goal-directed behavior, and long-horizon planning, respectively. Each suite consists of 10 tasks with 50 human-teleoperated demonstrations per task. We run 50 rollouts per task and report the average success rate. Figure 9 shows an example scene from each suite, and Table 7 lists example instructions. All ablations in Section 5.3 run on LIBERO-Long, the suite with the longest tasks.

CALVIN simulation benchmark. CALVIN assesses long-horizon, language-conditioned manipulation in four visually distinct environments, A, B, C, and D, each with a Franka Emika Panda robot and a tabletop scene with diverse objects, as Figure 10 shows. Its training data are about 24 hours of teleoperated play, and it defines 34 subtasks, which Table 8 groups into eight categories with example instructions. Evaluation uses 1,000 chains of five consecutive commands executed in order, and we report the success rate at each chain length and the average sequence length. Under the ABC→D setting of Section 5.2, policies are trained on environments A, B, and C and evaluated on environment D, whose scene they never see during training.

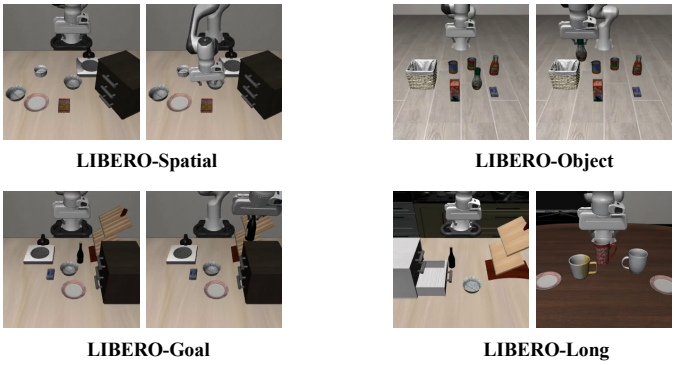


Figure 9: Example scenario from the LIBERO benchmark.

Table 8: Example language instructions for the eight CALVIN task categories.

CALVIN	Task Instructions
Turn On/Off	Turn on/off lightbulb. Turn on/off LED.
Open/Close Drawer	Open drawer. Close drawer.
Move Slider	Move slider left. Move slider right.
Rotate	Rotate red block left/right. Rotate pink block left/right. ...
Stack/Unstack	Stack block. Unstack block.
Lift	Lift red block table. Lift pink block drawer. ...
Place	Place in slider. Place in drawer.
Push	Push red block left/right. Push into drawer. ...

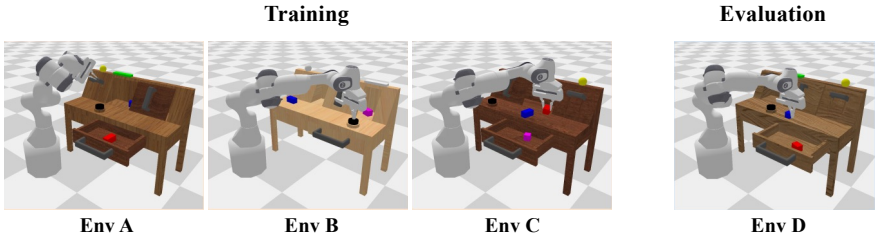


Figure 10: Example scenario from the CALVIN benchmark.

B.2 Additional ablation results on LIBERO-Spatial

To check whether the ablation trends on LIBERO-Long hold on another suite, we repeat the trajectory-predictor, metric-depth, and frame ablations on LIBERO-Spatial, and Table 9 reports both suites. The ranking of supervision targets on LIBERO-Spatial matches that on LIBERO-Long. The monotonic ladder, from *w/o Traj.* to *w/ AuxTraj*, *w/ 2D*, *w/ 3D Pos.*, *w/ World*, and OASIS, holds on both suites, with OASIS reaching 99.0% on Spatial and 95.2% on Long. The variant *w/ 2D* stays close to *w/o Traj.* on Long, at 90.7% versus 89.5%, and adds a modest gain on Spatial, at 93.3% versus 91.6%, indicating that image-plane geometry alone cannot align the intermediate with the action space. Moving to *w/ 3D Pos.*, which adds depth, gains a further 1.6 points on Long and 2.1 on Spatial, and moving to *w/ World*, which adds rotation, gains 0.9 and 1.1 points. OASIS, with camera-frame $SE(3)$, beats *w/ World* by 2.0 points on Long and 2.5 points on Spatial, consistent with deferring the uncalibrated camera-to-world transform to the learned decoder as analyzed in Section 3.2. Removing metric depth also costs more on Spatial, 5.6 points against 3.4 on Long.

Unlike *w/o Traj.*, *w/ AuxTraj* keeps the trajectory predictor's parameters, so the OASIS gain of 5.5 points on LIBERO-Long and 7.1 on LIBERO-Spatial over *w/ AuxTraj* cannot come from trajectory-branch capacity and instead reflects routing the pose-supervised hidden states to the decoder.

Table 9: Intermediate-type screen on LIBERO-Long and LIBERO-Spatial, plus the metric-depth ablation. OASIS, *w/ World*, *w/ 3D Pos.*, and *w/ 2D* differ only in the supervision target. On both suites, success rises monotonically with its geometric richness.

Variant	LIBERO-Long	LIBERO-Spatial
w/o Traj.	89.5	91.6
w/ AuxTraj	89.7	91.9
w/ 2D	90.7	93.3
w/o Metric	91.8	93.4
w/ 3D Pos.	92.3	95.4
w/ World	93.2	96.5
OASIS	**95.2**	**99.0**

Table 10: Closed-form decoder ablation. The hardcoded pipeline reads the same predicted $SE(3)$ trajectory as OASIS but executes it via known camera extrinsics and privileged simulator information for the gripper, a strictly stronger setting than OASIS's learned decoder.

Decoder	LIBERO-Spatial	LIBERO-Long
OASIS (learned decoder)	**99.0**	**95.2**
Hardcoded geometric pipeline	12.4	0.0

B.3 Closed-form decoder ablation

To probe whether the learned action decoder is reducible to closed-form geometry, we replace it with a hardcoded geometric pipeline that converts the same predicted camera-frame $SE(3)$ trajectory into world-frame poses via known camera extrinsics and emits relative actions, with gripper open/close further supplied by privileged simulator information unavailable to OASIS. Despite this strictly stronger setting, the hardcoded pipeline collapses to 12.4% on LIBERO-Spatial and 0.0% on LIBERO-Long against OASIS's 99.0% and 95.2% with the same trajectory predictor, as shown in Table 10. Rollouts confirm that the predicted trajectory does encode meaningful geometry, since the robot consistently moves toward the target object, but small residual prediction noise, contact dynamics, and gripper timing compound into failures that this closed-form pipeline cannot absorb. This result rules out the interpretation that OASIS reduces to a renamed closed-form execution pipeline over predicted poses, and confirms that the learned decoder is necessary.

B.4 Decoder attention analysis

To inspect how the action decoder reads its context, we record the cross-attention weights of both decoder blocks over 3,372 forward passes during LIBERO-Long evaluation rollouts with 10 trials per task. Table 11 averages the first block over its eight heads, with one row per action step and one column for each of the H trajectory hidden states and for the state embedding $\mathbf{h}_{\text{state}}$. Every action step places 0.123 to 0.126 of its attention on a single trajectory hidden state and at most 0.005 on any other. Per-head inspection shows that one head of the first block gives the selected state 0.98 to 1.00 of its weight, so the eight-head average is about $1/8$, while the other seven heads place nearly all their weight on $\mathbf{h}_{\text{state}}$, which therefore receives about $7/8$ of the average. Steps 7 and 8 read each other's states, so the mapping is one-to-one but not index-matched. In the second block, every action step places an average weight of 1.000 on $\mathbf{h}_{\text{state}}$.

Since the decoder adds no encoding of the order of the trajectory hidden states, permuting them leaves its output unchanged up to floating-point summation order. On identical inputs, the *Shuffled-H* permutation changes no action component by more than 7.8×10^{-3} in bf16 or 3.3×10^{-6} in fp32, while *Mean-pool* changes the actions by up to 2.56 on an action scale of about 2.25.

The attention pattern shows that each action step combines one trajectory hidden state with the state embedding. The hardcoded pipeline of Appendix B.3 likewise combines a target pose with the current one, but the two differ in what they consume. The pipeline reads the decoded trajectory, the six numbers per step that the linear head produces from $\mathbf{h}_{\text{traj}}$, converts them with the known camera

Table 11: Cross-attention of the first decoder block on LIBERO-Long, averaged over its eight heads and 3,372 forward passes. Rows are action steps, and columns are the eight trajectory hidden states and the state embedding. The largest weight on a trajectory hidden state in each row is in bold.

	Trajectory hidden state								
Step	1	2	3	4	5	6	7	8	$\mathbf{h}_{\text{state}}$
1	**0.126**	0.003	0.004	0.005	0.003	0.001	0.000	0.001	0.856
2	0.002	**0.123**	0.001	0.000	0.000	0.000	0.001	0.001	0.872
3	0.000	0.000	**0.125**	0.000	0.000	0.000	0.001	0.000	0.873
4	0.000	0.000	0.000	**0.125**	0.000	0.000	0.001	0.000	0.873
5	0.000	0.000	0.000	0.001	**0.124**	0.001	0.000	0.000	0.873
6	0.000	0.000	0.000	0.000	0.000	**0.125**	0.000	0.000	0.873
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.125**	0.874
8	0.000	0.000	0.000	0.000	0.001	0.000	**0.125**	0.000	0.873

Table 12: Mean and standard deviation of OASIS over three random seeds on LIBERO.

LIBERO	**Spatial**	**Object**	**Goal**	**Long**	**Average**
OASIS	99.0 ± 0.2	98.8 ± 0.0	97.4 ± 0.3	95.2 ± 0.4	97.6 ± 0.1

Table 13: Mean and standard deviation of OASIS over three random seeds on CALVIN ABC→D.

CALVIN	**Tasks Completed in a Row**					**Avg.**
	1	2	3	4	5	
OASIS	98.1 ± 0.1	94.9 ± 0.2	91.7 ± 0.2	88.9 ± 0.2	83.3 ± 0.3	4.57 ± 0.01

extrinsics, and executes the resulting geometric difference. The decoder instead conditions on $\mathbf{h}_{\text{traj}}$ itself and is trained on demonstrated actions rather than made to execute the plan, so it can act on an imperfect plan instead of executing the prediction error verbatim.

B.5 Seed variance of the simulation results

The OASIS results in Tables 1 and 2 are means over three random seeds, and Tables 12 and 13 add their standard deviations. The largest deviation on LIBERO is 0.4 points on Long, where the three runs reach 95.6%, 95.2%, and 94.8%, so even the lowest run stays above the 94.0% of Unified-VLA. On CALVIN, the deviation is at most 0.3 points at every chain length and 0.01 in average sequence length, where even the lowest run reaches 4.56, above the 4.44 of DreamVLA.

C Additional details for real-world experiments

C.1 Real-world experimental settings

The real-world experiments span three suites, with six tasks each in Goal and Spatial and one in Long. Table 14 lists their instructions, and Figure 11 shows the setup and an execution per suite. Both robots, a Franka Research 3 and a Kinova Gen3, use a third-person and a wrist camera.

Goal suite. The robot must place the object named in the instruction into the named bowl, so the suite tests instruction following across six tasks. The layout of objects and bowls stays fixed while its distance from the robot base varies across trials, which prevents memorizing a single configuration and tests robustness to different initial positions.

Spatial suite. Its six tasks test spatial reasoning under geometric constraints, requiring precise localization and placement of objects in various poses. In *Place pot*, both sides of the pot must rest on the two wooden bars, or the pot tips over, and in *Hang cup*, the handle must engage the cup holder.

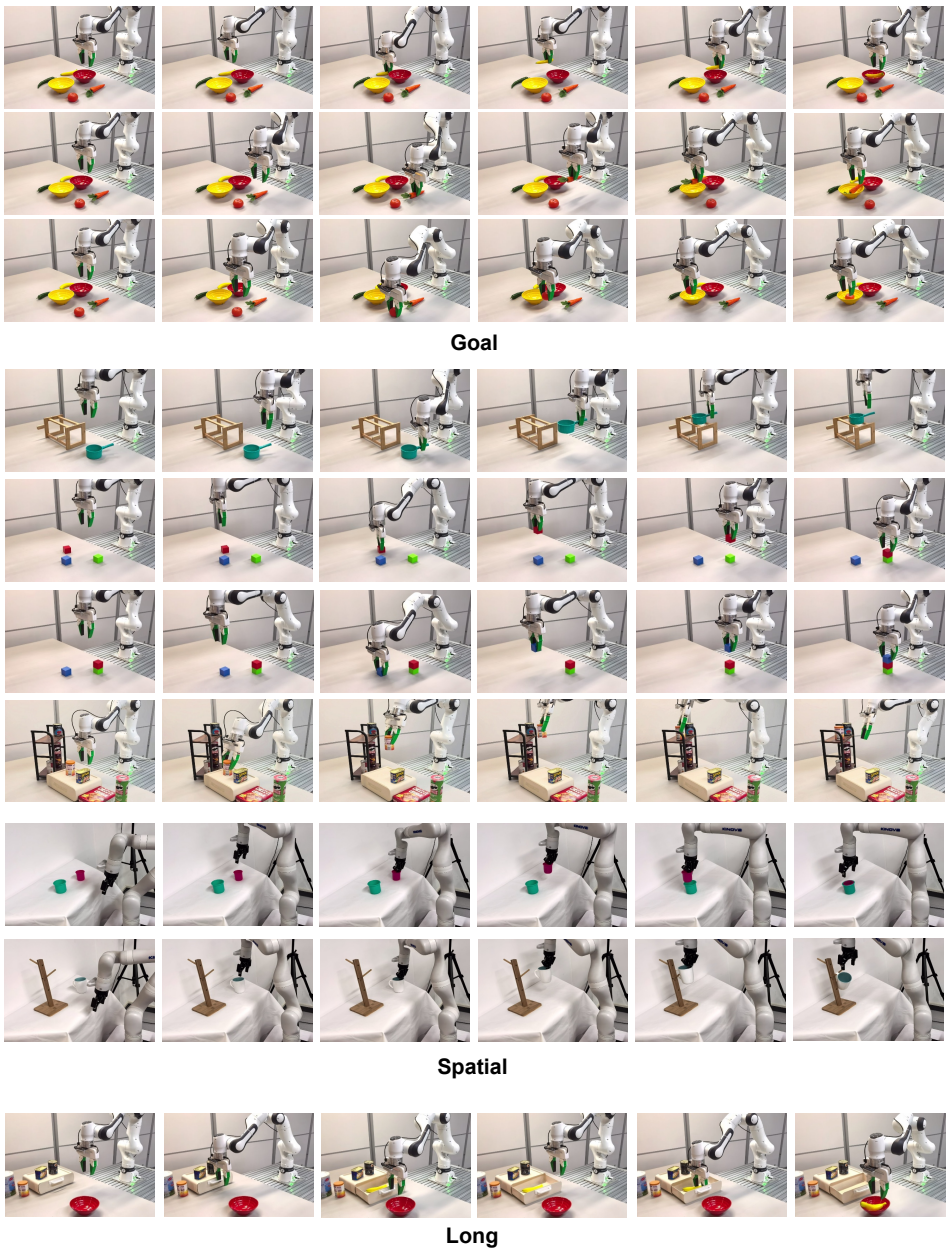


Figure 11: Real-world execution examples of the Goal, Spatial, and Long suites. The Goal suite utilizes a tabletop environment with two bowls and four objects arranged in a structured layout, where the robot must place the correct object into the correct bowl based on the language instruction. The Spatial suite comprises six tasks, namely placing the pot on the wooden bracket, stacking blocks, building towers, putting the orange can on the shelves, placing the pink cup into the cyan cup, and hanging the cup on the cup holder. In the Long task, the robot must execute a multi-stage sequence, first opening a drawer, then retrieving the banana from it, and finally placing it into the red bowl.

Table 14: Examples of language instructions for real-world experiments.

Real-World Experiments	Task Instructions
Goal	Place the banana into the red bowl. Pick up the carrot and place it into the red bowl. Grasp the orange and put it into the yellow bowl. ...
Spatial	Put the pot on the wooden bracket. Stack the red cube on top of the green cube. Place the blue cube on top of the red cube. Put the orange can on the shelves. Place the pink cup into the cyan cup. Hang the cup on the cup holder.
Long	Open the drawer and place the banana into the red bowl.

Table 15: Initialization, learning rate, and batch size of each method in the real-world comparison. Each baseline keeps the default learning rate of its official implementation, and batch sizes are given as the batch per GPU times the number of GPUs.

Method	Initialization	Learning rate	Batch size
$\pi_{0.5}$	Official pretrained	5×10^{-5}	16×4
RDT	Official pretrained	1×10^{-4}	16×4
Seer-Large	Official pretrained	1×10^{-4}	16×4
ACT	From scratch	1×10^{-4}	16×4
OASIS	No robotic pretraining	2×10^{-4}	16×4

Long task. The Long task chains three stages, opening a drawer, retrieving the banana from it, and placing the banana into the red bowl, so it demands both precise low-level control and long-horizon reasoning. Errors compound along the chain, since imprecision while opening the drawer affects the later grasp, so the policy must correct deviations in closed loop. The real-world data-scaling study of Appendix C.2 also uses this task, and Figure 13 shows one of its failures.

Training and evaluation protocol. All four baselines are rerun under one protocol shared with OASIS, and Table 15 lists the settings of each method. Every method is trained for 50k steps on the same 50 demonstrations per task, with both camera views at 224×224, and is evaluated at its final checkpoint over 60 trials per task. No hyperparameter search or checkpoint selection is performed for any method. On the Goal suite, the language-conditioned methods are trained jointly on all tasks and ACT is trained per task, while on the Spatial and Long suites every method is trained per task. Since $\pi_{0.5}$, RDT, and Seer-Large start from robotic-pretrained weights while ACT and OASIS do not, the robotic-pretraining axis favors the baselines. Under this protocol, OASIS still leads $\pi_{0.5}$, the strongest baseline, by 7.6 points on average, 89.2% versus 81.6% in Table 3.

OOD scenario settings. To evaluate zero-shot generalization, we build three out-of-distribution scenarios on the Goal suite, illustrated in Figure 12. The unseen backgrounds scenario adds objects and a tabletop that never appear in training, testing whether a policy attends to task-relevant features and ignores visual distractors. The displaced camera scenario moves only the third-person camera at test time, by a translation of about 15 cm relative to the training viewpoint, while the wrist camera and all other components remain unchanged. No recalibration or test-time adaptation is performed for any policy. This configuration probes robustness to a single uncalibrated third-person viewpoint shift. The human interference scenario relocates the target bowl while the robot is transporting the object, testing whether a policy can retarget in closed loop when its goal moves. Figure 8 in Section 5.4 reports the resulting success rates of OASIS, $\pi_{0.5}$, and RDT, and shows that OASIS loses less than $\pi_{0.5}$ under each of the three scenarios. Against the in-distribution Goal success, the drops are 3.8, 7.4, and 12.1 points for OASIS and 6.4, 18.6, and 13.5 for $\pi_{0.5}$.

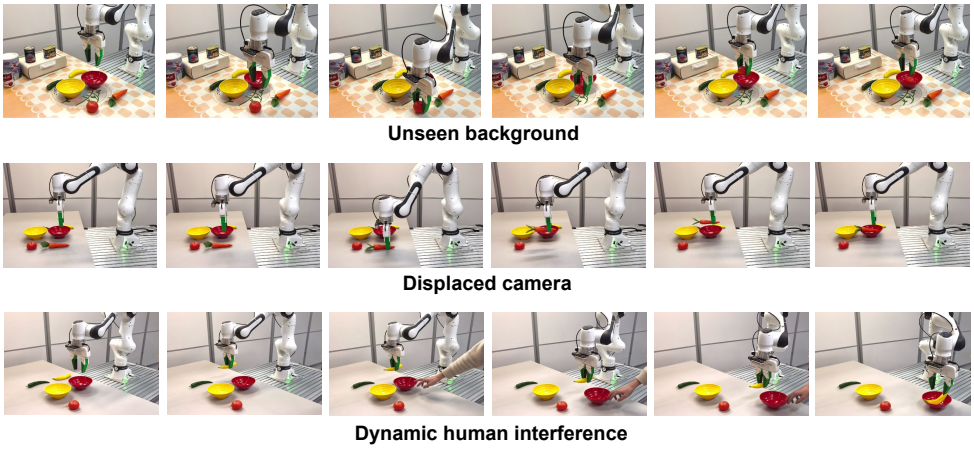


Figure 12: Real-world execution examples under the three OOD scenarios.

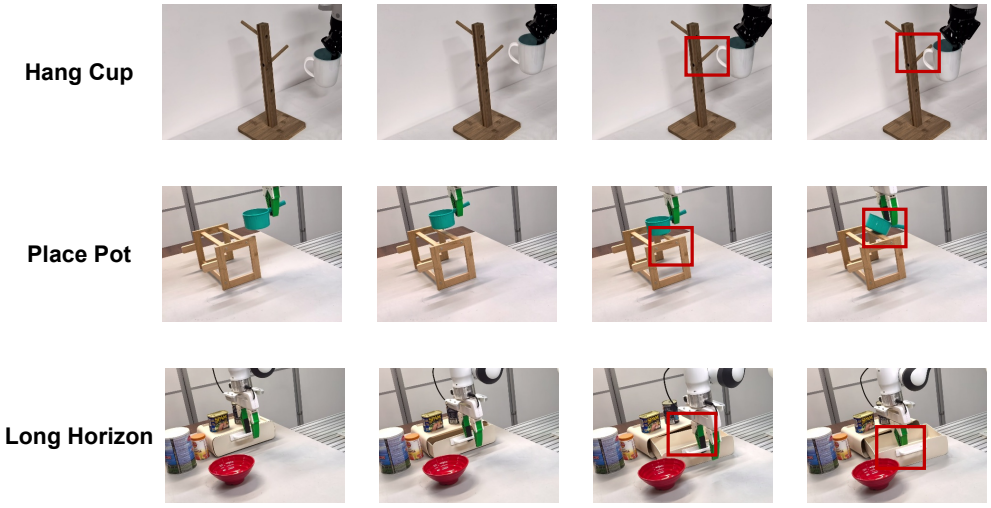


Figure 13: Failure cases of OASIS in real-world execution.

C.2 Additional real-world experimental results

Failure analysis. As shown in Figure 13 and Table 16, three of the four lowest real-world success rates involve tasks that demand sub-centimeter placement or precise orientation. *Hang cup*, at 76.6%, fails when a small orientation deviation keeps the handle from engaging the holder, and *Place pot*, at 83.3%, fails when small x-axis or yaw deviations leave the pot resting on a single supporting bar. The *Long horizon* task, at 83.3%, is dominated by stage compounding: imprecision during drawer opening shifts the banana's reachable region and propagates into grasp failures downstream. Figure 13 shows one failure of each of these three tasks, with the failure marked by a red box.

To test whether the *Hang cup* and *Place pot* failures stem from trajectory prediction, Table 17 measures the position and rotation errors of the predicted trajectory offline on in-distribution demonstrations from both platforms, with separate columns for the last quarter of each trajectory, where fine manipulation happens. Among the tasks in the table, the one with the lowest success rate on each platform has the most accurate predicted trajectory in every column, and the same ordering holds for median errors. Only the ordering within a platform is informative, since rotation errors differ several-fold across platforms. Prediction accuracy therefore does not appear to be the main bottleneck, and the failures point instead to execution tolerance. The pot, for example, must rest on both wooden bars, so deviations that the other tasks absorb are enough to fail it.

Table 16: Per-task success rates of OASIS on the real-world Goal, Spatial, and Long suites.

Category	Task Description	Success Rate (%)
Goal	Place the banana into the red bowl	100.0
	Place the banana into the yellow bowl	100.0
	Place the carrot into the red bowl	100.0
	Place the carrot into the yellow bowl	100.0
	Place the orange into the red bowl	95.0
	Place the orange into the yellow bowl	96.7
Spatial	Stack blocks	90.0
	Build towers	90.0
	Place pot on wooden bracket	83.3
	Put orange can on shelves	81.6
	Hang the cup	76.6
	Place the pink cup	93.3
Long	Open the drawer and place the banana into the red bowl	83.3

Table 17: Trajectory prediction error on in-distribution demonstrations, measured offline. SR is the success rate in percent, and Pos. and Rot. are the mean position error in cm and the mean rotation error in degrees. The most accurate task on each platform is in bold.

			Full trajectory		**Last 25%**	
Platform	**Task**	**SR**	**Pos.**	**Rot.**	**Pos.**	**Rot.**
Kinova	Hang the cup	76.6	**1.02**	**0.08**	**1.33**	**0.11**
	Place the pink cup	93.3	1.27	0.09	1.97	0.16
Franka	Place pot on wooden bracket	83.3	**0.83**	**0.33**	**1.25**	**0.32**
	Stack blocks	90.0	1.02	0.38	1.69	0.47
	Build towers	90.0	0.91	0.68	1.31	0.95

Inference speed analysis. To deploy the trained policy without operating-system or code-version conflicts, we wrap model loading and inference in a WebSocket server on a remote machine. The server communicates in real time with the robot's control computer, receiving live camera images at up to 30 Hz and the end-effector state, and returning actions for execution. On an NVIDIA RTX 4090 GPU without additional computational load, the policy consumes about 4.5 GB of GPU memory. Each inference cycle predicts the $SE(3)$ trajectory and 8 subsequent action steps, executed sequentially. The end-to-end response time of this server loop, including the WebSocket transport, is about 0.05 s, corresponding to roughly 20 Hz. To isolate policy inference, we also time one inference on a single RTX 4090 with the transport removed, at batch size 1 in bf16, taking the mean over 100 runs after 10 warm-up iterations. OASIS takes 45 ms, against 120 ms for $\pi_{0.5}$ and 260 ms for RDT, so it runs 2.7 and 5.8 times faster on the same GPU. Each number is the time to produce one action chunk, so for the flow-matching and diffusion baselines it includes their sampling iterations.

Data-scaling on the real-world Long task. To probe how data efficiency interacts with the aligned-intermediate design, we train OASIS and $\pi_{0.5}$ on the real-world Long task with progressively reduced demonstration budgets and evaluate each variant over three independent runs of 20 trials each, for 60 trials per configuration. Table 18 reports success rate as a function of demonstration count. With only 10 demonstrations, OASIS reaches 35.0% while $\pi_{0.5}$ reaches 15.0%. At 25 demonstrations, OASIS reaches 55.0% while $\pi_{0.5}$ reaches 35.0%, and at the full 50 demonstrations, OASIS reaches 83.3% while $\pi_{0.5}$ reaches 71.6%. OASIS thus matches the observed 25-demonstration success rate of $\pi_{0.5}$ using only 10 demonstrations on this task.

Table 18: Data-scaling results on the real-world Long task. We train OASIS and $\pi_{0.5}$ with 10, 25, and the full 50 demonstrations and evaluate each variant over three independent runs of 20 trials each, for 60 trials per configuration. All numbers are success rates in percent.

Method	10 demos	25 demos	50 demos
$\pi_{0.5}$	15.0	35.0	71.6
OASIS (Ours)	**35.0**	**55.0**	**83.3**

Table 19: Wilson 95% confidence intervals, in percent, for OASIS vs $\pi_{0.5}$ on the real-world results.

Method	Goal	Spatial	Long
$\pi_{0.5}$	95.0 [92.2, 96.8]	78.3 [73.8, 82.3]	71.6 [59.2, 81.5]
OASIS (Ours)	98.6 [96.8, 99.4]	85.8 [81.9, 89.1]	83.3 [72.0, 90.7]

C.3 Wilson 95% confidence intervals for real-world results

To quantify finite-trial uncertainty, we report Wilson score 95% confidence intervals in Table 19 for the head-to-head OASIS versus $\pi_{0.5}$ comparison on the real-world main results in Table 3. Goal and Spatial each comprise six tasks evaluated over 60 trials per task, so we report a pooled descriptive Wilson interval over n=360 trials per suite, while Long is a single task with n=60. The intervals overlap by under one point on Goal and Spatial and widely on Long.

D Broader impacts

OASIS proposes a new design principle for visuomotor policies, aligning the intermediate representation with the action space via $SE(3)$ end-effector trajectory prediction. Instead of decoding actions from observation-space features alone, our framework first predicts a camera-frame $SE(3)$ trajectory whose pose-supervised hidden states condition the action decoder. This aligned intermediate improves manipulation precision and generalization across simulation benchmarks and real-world platforms.

A key strength of OASIS lies in its training efficiency and accessibility. It requires neither large-scale robotic pretraining nor annotated spatial labels, with supervision derived solely from standard expert demonstrations. With only 0.18B trainable parameters, trained on four GPUs and run on a single RTX 4090, the framework lowers the cost barrier for research groups and small organizations developing manipulation policies. Separating geometric prediction from action execution also makes the intermediate representation inspectable, which can support debugging and safer deployment.

Practically, this design can benefit assistive household robots, industrial manipulators, and tabletop laboratory automation, where precise $SE(3)$ control under modest data budgets is essential. We hope OASIS inspires further work on geometrically aligned intermediates and low-cost robot learning.

OASIS also carries risks. A learned policy can still fail under conditions outside its training data, and our OOD tests show success dropping by up to 12.1 points. A robot acting on a wrong prediction near people could break objects or cause injury, so deployment outside the lab calls for speed and force limits, a monitored workspace, and a human who can stop the robot at any time.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction claim that OASIS aligns the intermediate representation with the action space via $SE(3)$ end-effector trajectory prediction, and the experiments in Sections 5.2 and 5.4 verify the claimed gains over VLA and WAM baselines on LIBERO and CALVIN, with consistent improvements on real-world Franka and Kinova platforms including OOD perturbations on the Goal suite.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations are discussed in Section 6.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: The paper uses a design principle in Section 3.2 to motivate the architecture, but it does not present formal theorems or proof-based claims.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 5.1 reports the architecture, parameter counts, and compute setup. Appendix A gives the parameter breakdown in Table 4, the optimizer, and the training settings in Table 6. Section 5.2 and Appendix B.1 describe the standard LIBERO and CALVIN ABC→D evaluation protocols, and Appendix C.1 details the real-world platforms, task suites, OOD scenario settings, and the training protocol shared by all compared methods, with the per-method settings in Table 15.

Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: Code, pretrained checkpoints, and the processed training data will be released on the project page `https://oasis-neurips.github.io`, and the original data of the LIBERO and CALVIN benchmarks are publicly available.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Section 5.1 specifies the architecture, trajectory horizon $H{=}8$, batch size 64 across 4 NVIDIA A800 GPUs, and the LoRA-adapted versus frozen parameter split, while Appendix A provides the AdamW optimizer, learning rate $2{\times}10^{-4}$, cosine schedule with warm-up, and 50,000 training steps, with results averaged over three seeds.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: Wilson 95% confidence intervals are reported in Appendix C.3 for the head-to-head OASIS versus $\pi_{0.5}$ real-world comparison, while simulation results follow the standard LIBERO and CALVIN protocols averaged over three random seeds, with their standard deviations for both benchmarks in Appendix B.5.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Section 5.1 states that experiments use 4 NVIDIA A800 GPUs with batch size 64, Appendix A provides 50,000 training steps, and Appendix C.2 reports approximately 4.5 GB of inference-time GPU memory on a single NVIDIA RTX 4090.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics in the research.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix D discusses positive impacts such as lower training cost and assistive applications, and the negative impact that a policy failing near people could break objects or cause injury, together with mitigations.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This work releases a manipulation policy trained on benchmark and lab-collected demonstrations, not pretrained generative models, scraped datasets, or other high-risk assets requiring safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We follow the licenses of open-source code, data, and models.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: The paper does not release new assets.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects, and real-world demonstrations were collected via teleoperation by the authors themselves.

Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects beyond the authors are involved in any experiment, and under our institutional guidelines IRB approval is not required for self-conducted teleoperation of a robot platform.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: The 3D-aware feature encoder uses a VLM with a Qwen2.5-0.5B language backbone, a standard pretrained component described in Section 4.2 and Appendix A.1. Beyond it, LLMs were used only for grammar and phrasing suggestions.

    Guidelines:

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