Lift, See, Act: Hierarchical Robot Policy Pretraining with 3D Foundation Models

Yiyuan Ge **Changxing Ding*** **Ziyu Hao** **Zijie Zheng** **Xiangmin Xu**
School of Electronic and Information Engineering
South China University of Technology
{202510182871,202610183440,202620112031}@mail.scut.edu.cn
{chxding,xmxu}@scut.edu.cn

Abstract

Robot policy pretraining based on human videos is crucial for improving the policy's generalization ability. One main challenge for this task is the lack of explicit action-relevant representations in such unlabeled data. Recent works tend to estimate 3D hand motion trajectories from videos using 3D foundation models (3D FMs). However, they keep solely the sparse trajectories as supervision, ignoring the rich, fine-grained information produced during the trajectory extraction process. To address this issue, we propose **LSA**, a hierarchical robot policy pretraining framework that includes three phases: **L**ift, **S**ee, and **A**ct. Specifically, the Lift phase elevates 2D video observations into dense 3D representations by adaptively aligning features in the policy's shallow layers with the intermediate representations extracted from multiple 3D FMs. Building upon these lifted representations, the See phase equips the policy with explicit geometric and interactive awareness through dual guidance. It introduces depth-map reconstruction to help the model comprehend 3D spatial layouts and utilizes hand region cues to explicitly supervise the encoder's attention toward task-relevant interaction hotspots. Empowered by the dense 3D representation learning and precise spatial guidance, our policy achieves robust and accurate robotic manipulation in the Act phase. Extensive experiments on diverse simulation and real-world tasks demonstrate that LSA significantly outperforms current state-of-the-art approaches. The project page is available at: https://yiyuan00.github.io/Lift-See-Act.

1 Introduction

Large-scale pretraining is crucial for improving the generalization ability of machine learning models [1, 2]. In robotics, pretraining is equally important for policy learning, since high-quality manipulation data are expensive to collect and therefore difficult to scale up. Human videos are a key source of pretraining data, as they cover a wide range of real-world manipulation scenarios and contain rich manipulation knowledge [3, 4]. However, using human videos to pretrain robot policies remains a challenge, as such data inherently lack action-relevant representations [5].

Existing methods for extracting explicit action-relevant representations from human videos can be divided into two categories. The first category of methods extracts 2D visual primitives from video frames, such as contact points [6, 7], contact heatmaps [8], and object bounding boxes [9], to characterize where contact occurs and which regions are potentially involved in manipulation. However, these methods lack temporal representations of the 3D manipulation process. The second category of methods attempts to estimate 3D hand motion trajectories from videos [10–14]. These trajectories reflect both the displacement of the hand in the 3D space and the

*Corresponding author.

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

temporal evolution of that displacement; therefore, they are more in line with the robot's action prediction space. Recently, with the development of 3D foundation models (3D FMs) [15–18], the accuracy of the estimation of 3D hand motion trajectories has improved substantially [12–14]. However, as illustrated in Fig. 1, the latter category of approaches only keeps the estimated hand motion trajectories as supervision for robot policy pretraining. They ignore more fine-grained knowledge, e.g., rich geometric representations and depth maps, produced by the 3D FMs during the hand motion trajectory prediction process. This dense knowledge is fundamental for a robot policy in predicting the 3D motion of the end effector. This insight motivates us to study how to make the best use of multi-tiered knowledge from 3D FMs to promote robot policy pretraining.

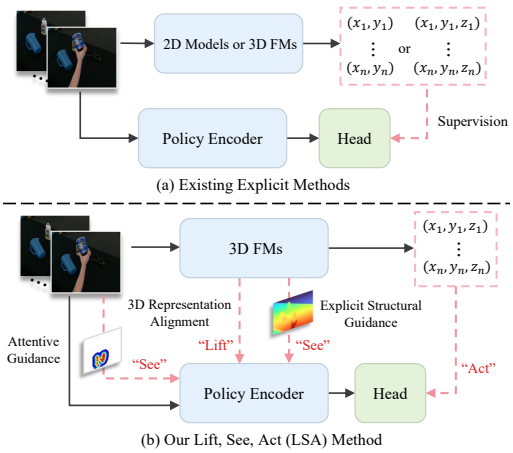


Figure 1: Comparisons between methods that extract explicit action-relevant representations from human videos. (a) Existing methods adopt solely the estimated hand motion trajectories as supervision [8, 12, 10]. (b) Our LSA leverages multi-tiered information from multiple 3D FMs to facilitate robot policy pretraining.

Consequently, we propose **LSA**, a hierarchical framework for robot policy pretraining based on three phases: **Lift**, **See**, and **Act**. Specifically, in the **Lift** phase, our goal is to lift 2D video observations into rich 3D representations. We first construct a 3D hand motion trajectory extraction pipeline utilizing multiple 3D FMs to process unlabeled human videos. This pipeline not only extracts the motion trajectories, but also keeps dense intermediate representations and depth maps produced by these 3D FMs. We then enforce the shallow-layer representations of the policy to align with the intermediate representations of each 3D FM, respectively. We also propose a strategy to adaptively adjust the weights of these alignment losses. This process effectively lifts the policy's 2D representations into 3D, endowing the policy with more fine-grained geometric knowledge.

Building upon these lifted representations, the **See** phase equips the policy with explicit geometric and interactive awareness. While the Lift phase establishes a dense 3D representation space, these representations are implicit. Robot manipulation inherently requires an intuitive and precise understanding of the structure of the physical scene, along with a distinct visual focus on the manipulation objectives. To achieve this, the See phase provides dual guidance: explicit structural guidance and attentive guidance. First, leveraging the extracted depth maps, we introduce depth map reconstruction as an auxiliary task, providing explicit 3D geometric constraints to enable the model to comprehend 3D spatial layouts. Second, we utilize the hand region as a cue to guide the encoder's attention toward task-relevant interaction hotspots. Specifically, we encode 2D hand keypoints into a Gaussian heatmap to represent the hand region. We then select the attention map most similar to this heatmap across all heads in the shallow layers as our target, and finally use it to supervise the mean attention of all heads within a layer. The above dual guidance ensures the policy explicitly **Sees** the 3D scene structure and concentrates on the task-relevant interaction regions.

To the best of our knowledge, LSA is the first approach to leverage multi-tiered 3D information extracted from 3D foundation models for hierarchical robot policy pretraining on human videos. Empowered by the dense 3D representation learning in the Lift phase and the dual guidance in the See phase, our policy predicts 3D hand motion trajectories in the **Act** phase, ultimately achieving robust and accurate robot manipulation. Extensive experiments demonstrate the effectiveness of our design and show that LSA outperforms state-of-the-art methods on various simulation and real-world tasks.

2 Related Works

Existing approaches to pretraining robot policies from human videos often construct action-relevant representations to bridge the gap between passive visual observations and robot control. Based on the space in which these representations are extracted, existing approaches can be broadly categorized into 2D and 3D methods.

2D Methods. The first category of methods constructs action-relevant representations within the 2D image space, which can be further divided into implicit and explicit methods. Implicit methods model inter-frame relationships to extract abstract state representations. One line of work uses contrastive learning [19, 20] to constrain the distance between visual embeddings across different time periods, capturing the temporal dynamics of visual observations (i.e., how the current state transitions to the next). To characterize these state transitions more precisely, other approaches learn discrete latent actions between frames through VQ-VAE objectives [21], subsequently using these latent actions as pseudo-action labels for unannotated videos [3, 22, 23]. Recently, some methods train video prediction models [24–26, 4] to extract predictive visual representations, providing embodied agents with dense information about future states.

In contrast, explicit methods directly extract 2D visual primitives at the pixel level. These approaches typically identify contact points [6, 7], contact heatmaps [8], or bounding boxes [9, 27] to delineate potential interaction regions. Furthermore, extracting human hand poses from video streams serves as a simple but scalable approach to represent the interaction trajectories [7, 28].

3D Methods. The above 2D representations fundamentally lack explicit depth information, limiting their ability to capture inherently 3D physical interactions. To address this limitation, the second category of methods turns to explicit 3D action-relevant representations, such as 3D hand trajectories [12] and parametric hand poses (e.g., MANO [29]) [30, 6]. Due to their 3D nature, these representations align more tightly with the robotic action space.

These 3D methods can be further divided into two groups according to their data requirements. The first group uses videos equipped with annotated 3D hand poses and camera parameters, typically captured through AR/VR devices or RGB-D SLAM [31]. They directly adopt these ground-truth 3D hand poses as supervision for the pretraining of robot policies [30, 10, 32, 33]. However, these videos are recorded in relatively controlled settings, which affects the scale of the video databases. In comparison, the second group constructs 3D representations directly from unlabeled, in-the-wild videos. Early efforts employ structure-from-motion and depth estimation models to recover camera intrinsics and metric depth. They then combine these estimations with continuous 2D hand poses to obtain 3D hand motion trajectories [11, 34]. Recently, the rapid advancement of 3D Foundation Models (FMs) [15, 17, 16] has enabled the extraction of 3D hand trajectories with substantially improved fidelity and efficiency [12, 10].

In this paper, we also use 3D FMs to construct action-relevant representations. Unlike trajectory-based methods that focus exclusively on 3D hand trajectories and Spatial Forcing [35], which distills dense geometric features from a single 3D FM, we extract hierarchical representations (e.g., geometric features and depth maps) from multiple 3D FMs during inference, providing multi-level 3D supervision for robot policy pretraining.

3 Methods

3.1 Problem Formulation and LSA Overview

Following prior work [8, 19, 6, 36, 37], we pretrain the visual encoder part of the robot policy using large-scale human videos. Once trained, the visual encoder F_{ϕ^*}, parameterized by the fixed pretrained parameters ϕ^*, is frozen and serves as the perceptual front-end of the robot policy. Given a current visual observation $\mathbf{I}$, the encoder maps it to a latent embedding $\mathbf{z} = F_{\phi^*}(\mathbf{I})$. This embedding $\mathbf{z}$ is then concatenated with the proprioceptive state of the robot $\mathbf{s}$ and fed into a policy head π_θ, parameterized by θ, to generate an action $\hat{\mathbf{a}} \sim \pi_\theta(\cdot \mid \mathbf{z}, \mathbf{s})$. The policy is trained through behavior cloning [38] using a dataset of expert demonstration trajectories $\mathcal{D}_{\mathrm{IL}} = \{\tau_i\}_{i=1}^{N_{\mathrm{demo}}}$, where each trajectory is defined as $\tau_i = ((\mathbf{I}_{i,t}, \mathbf{s}_{i,t}, \mathbf{a}_{i,t}))_{t=1}^{T}$. Here, N_{demo} is the number of demonstration trajectories and T is the trajectory length. The training objective minimizes the behavior cloning loss between the predicted action $\hat{\mathbf{a}}_{i,t}$ and the ground-truth expert action $\mathbf{a}_{i,t}$:

$$\min_{\theta} \mathbb{E}_{\tau_i \sim \mathcal{D}_{\mathrm{IL}},\, t \sim \mathcal{U}\{1,\ldots,T\}} \left[\mathcal{L}_{\mathrm{BC}} \left(\hat{\mathbf{a}}_{i,t}, \mathbf{a}_{i,t} \right) \right]. \tag{1}$$

Here, $\mathcal{L}_{\mathrm{BC}}$ denotes the behavior cloning loss, instantiated in our implementation as the negative log-likelihood of the expert action under the predicted policy distribution. During evaluation, the policy is deployed in a closed-loop control setting and its performance is quantified by the success rate averaged over multiple deployments.

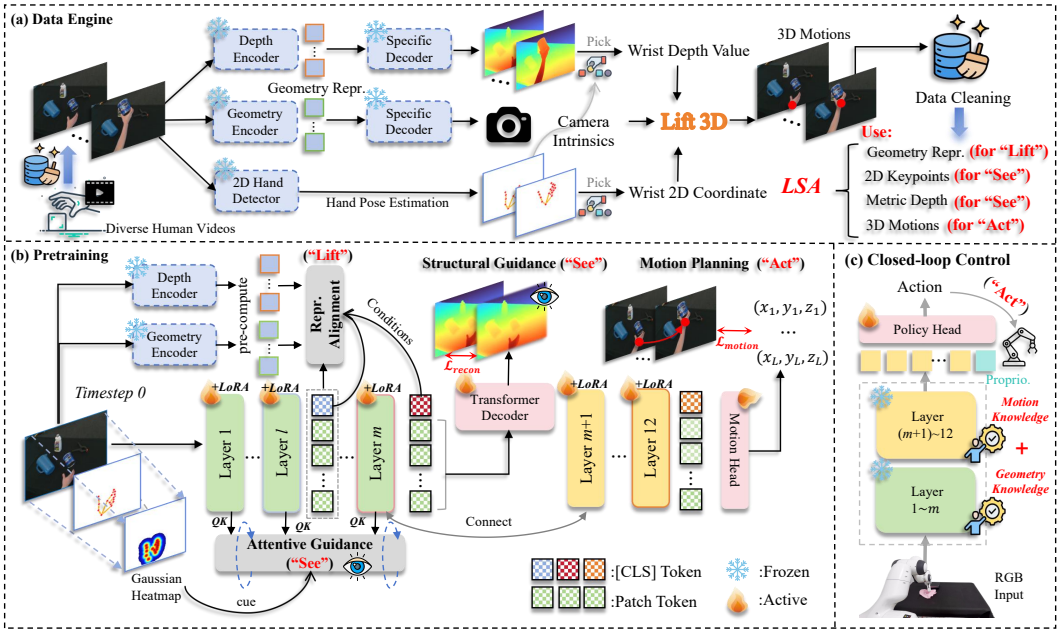


Figure 2: Architecture overview of LSA. (a) In the data engine phase, we utilize two 3D FMs to extract depth maps and camera intrinsics, respectively. Combined with the extracted 2D wrist coordinates, we obtain 3D hand motion trajectories. Furthermore, we also extract dense geometric representations and depth maps. (b) In the pretraining phase, we first align the policy's shallow-layer representations with the intermediate representations of the two 3D FMs pre-computed during the data engine phase. Then, we introduce depth-map reconstruction to equip the model with precise structural knowledge, and utilize hand heatmaps to guide the model's attention toward interaction hotspots. Following this, we predict the 3D hand motion trajectories. (c) Finally, we freeze the pretrained LSA and combine it with a policy head to perform closed-loop robotic control.

As highlighted in Section 1, relying solely on estimated hand trajectories for robot policy pretraining wastes the fine-grained geometric knowledge inherently produced by 3D Foundation Models (3D FMs). To make the best use of this multi-tier knowledge, we propose **LSA**, a hierarchical pretraining framework consisting of three distinct phases: **Lift**, **See**, and **Act**. First, the **Lift** phase elevates 2D video observations into a dense 3D representation space by aligning the policy's shallow-layer features with intermediate representations from 3D FMs. Building upon this, the **See** phase equips the policy with explicit geometric and interactive awareness through structural and attentive dual guidance. Finally, empowered by these representations, the **Act** phase translates visual inputs into 3D hand motion trajectories for robust manipulation.

3.2 The Lift Phase: 3D Representation Elevation

Data Engine. To extract 3D hand motion trajectories, we develop a data engine using multiple 3D FMs. Given an input video frame, we first extract the 2D wrist coordinates $\mathbf{p}_w$. Subsequently, we employ VGGT [15] to estimate camera intrinsics $\mathbf{K}$, and Depth Pro [16] to predict a metric depth map $\mathbf{D}_g$. The intermediate dense features extracted from VGGT and Depth Pro are denoted by $\mathbf{E}_c$ and $\mathbf{E}_d$, respectively. Using $\mathbf{K}$ and $\mathbf{D}_g$, we lift the 2D wrist coordinates $\mathbf{p}_w$ into the 3D space. Finally, we conduct necessary data cleaning to obtain high-quality pretraining data. Details of data cleaning and 3D lifting are provided in **Appendix A.6**.

Dense Representation Alignment. Geometric representations from 3D FMs have been shown to benefit robot policy learning [35]. Given an input image $\mathbf{I}_h \in \mathbb{R}^{H\times W\times 3}$, the vision encoder extracts visual features from it. Let $\mathbf{X}_a \in \mathbb{R}^{N_a\times D_a}$ denote the visual tokens from the l-th layer of the encoder. To distill the fine-grained geometric knowledge of the two 3D FMs into the policy, we align the shallow-layer visual tokens $\mathbf{X}_a$ with their geometric representations $\mathbf{E}_c \in \mathbb{R}^{N_c\times D_c}$ and $\mathbf{E}_d \in \mathbb{R}^{N_d\times D_d}$, respectively.

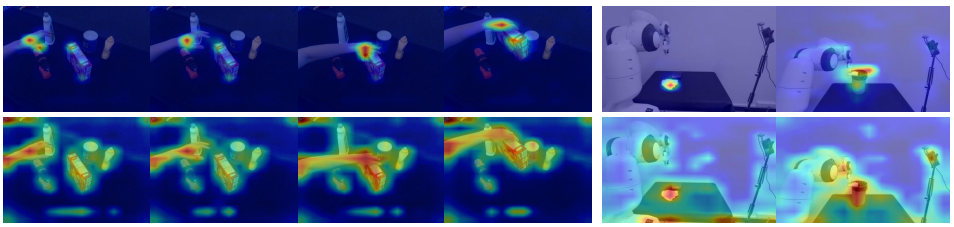


Figure 3: Effects of attentive guidance (AG). Left: attention maps on human videos. Right: attention maps on robotic scenes. In both cases, the top row shows results w/ AG and the bottom row shows results w/o AG.

To ensure spatial resolution compatibility, we first reduce the number of tokens in $\mathbf{E}_c$ and $\mathbf{E}_d$ to match N_a via downsampling and add positional encodings to their respective image patch tokens. We denote the resulting spatially-aligned features as $\mathbf{E}'_c \in \mathbb{R}^{N_a \times D_c}$ and $\mathbf{E}'_d \in \mathbb{R}^{N_a \times D_d}$. Next, to ensure compatibility with the feature dimensions, we map the embedding dimension of $\mathbf{X}_a$ to D_c and D_d using two separate MLPs, respectively. Following the typical token-level alignment paradigm [39], we employ a cosine similarity score to maximize the alignment between the visual tokens and the geometric signals. To balance the contributions of multiple geometric features, we introduce a set of dynamic modulation weights w_c and w_d. The overall alignment loss is formulated as:

$$\mathcal{L}_{\text{align}} = -\frac{1}{N_a} \sum_{i=1}^{N_a} \left[w_c \cdot \mathcal{S}(\text{MLP}_c(\mathbf{x}_{a,i}), \mathbf{e}'_{c,i}) + w_d \cdot \mathcal{S}(\text{MLP}_d(\mathbf{x}_{a,i}), \mathbf{e}'_{d,i}) \right], \tag{2}$$

where $\mathbf{x}_{a,i}$ denotes the i-th visual token in $\mathbf{X}_a$, and $\mathbf{e}'_{c,i}$ and $\mathbf{e}'_{d,i}$ represent the tokens in $\mathbf{E}'_c$ and $\mathbf{E}'_d$ that spatially correspond to $\mathbf{x}_{a,i}$, respectively. $\mathcal{S}(\cdot,\cdot)$ denotes the cosine similarity function. This alignment process effectively lifts the policy's 2D representations into 3D, endowing it with fine-grained geometric awareness.

To ensure that the model captures valuable geometric knowledge from multiple teachers, we propose a dynamic weight assignment mechanism inspired by [40]. We derive a condition query by aggregating the `[CLS]` tokens $\mathbf{c}_l$ and $\mathbf{c}_m$ from the l-th layer and the m-th layer of the vision encoder, where the m-th layer is further used for depth reconstruction. Then, we compute the scaled dot-product similarity between this query and the geometric tokens to adaptively determine the dynamic weights w_c and w_d for $\mathbf{E}'_c$ and $\mathbf{E}'_d$, respectively. More details are provided in **Appendix** A.3.

3.3 The See Phase: Dual Attentive and Structural Guidance

Although the Lift phase establishes a dense 3D representation space, these features remain largely implicit. Robot manipulation inherently requires an intuitive understanding of the structure of the physical scene, alongside a distinct visual focus on the manipulating agent and the target object. To achieve this, the See phase provides dual guidance.

Attentive Guidance. Learning affordance knowledge from human videos benefits policy pretraining [8, 6]. To further improve the policy's perception of hand-centric interaction regions, we design an attentive guidance mechanism. Specifically, we first warm up the LSA visual encoder for several epochs with the motion planning objective, allowing initial task-relevant attention patterns to emerge (see **Appendix** A.5). Given the detected hand keypoints, we generate a Gaussian heatmap and downsample it to the patch resolution, obtaining a spatial hand prior $\mathbf{h} \in \mathbb{R}^{N_a}$, where N_a denotes the number of spatial patches. For each head in the first R layers, we extract the [CLS]-to-patch attention map $\mathbf{A}_{n,k} \in \mathbb{R}^{N_a}$, where n and k denote the layer and head indices. We then compute its alignment score with the hand prior as:

$$s_{n,k} = \frac{1}{N_a} \sum_{i=1}^{N_a} A_{n,k,i} h_i. \tag{3}$$

The attention map with the highest score is selected as the guidance target:

$$(n^*, k^*) = \arg\max_{n,k} s_{n,k}, \qquad \mathbf{A}^* = \mathbf{A}_{n^*,k^*}. \tag{4}$$

We then use $\mathbf{A}^*$ as the self-distillation target to refine the averaged shallow attention. For each layer $n \leq R$, we compute the averaged multi-head attention map $\bar{\mathbf{A}}_n$ and minimize its distance to $\mathbf{A}^*$ after

spatial normalization:

$$\mathcal{L}_{\text{att}} = \sum_{n=1}^{R} \left\| \mathcal{N}(\text{sg}(\mathbf{A}^*)) - \mathcal{N}(\bar{\mathbf{A}}_n) \right\|_2^2, \tag{5}$$

where $\mathcal{N}(\cdot)$ denotes Min-Max normalization and $\text{sg}(\cdot)$ denotes stop-gradient. In this way, attentive guidance transfers the model's intrinsic hand-centric attention pattern to the shallow layers, enabling them to better focus on task-relevant hand-object regions.

As shown in Fig. 3, we visualize the averaged attention maps over the first R shallow layers to analyze the effect of attentive guidance. After applying attentive guidance, the model shows a clearer focus on the hand region. Notably, it also exhibits emergent attention to interacting objects, which can be attributed to the motion planning (**Sec.** 3.4) objective that tightly couples future hand trajectories with hand-object spatial relations. Moreover, Fig. 3 shows that these refined attention patterns transfer to robotic manipulation scenes, enabling the model to focus more on task-relevant regions while reducing attention to background.

Explicit Structural Guidance. To provide explicit 3D geometric constraints and enable the model to comprehend 3D spatial layouts more precisely, we introduce depth-map reconstruction as an auxiliary task. We utilize the image patch tokens from the m-th layer of the encoder, denoted as $\mathbf{X}_m$, which are processed by a transformer decoder to regress the depth map. This reconstruction is supervised by the metric depth map $\mathbf{D}_g$ generated during the data engine stage, and the reconstruction loss $\mathcal{L}_{\text{recon}}$ is computed using mean squared error (MSE).

3.4 The Act Phase: Motion Planning

Empowered by the dense 3D representation learning in the Lift phase and the precise dual guidance in the See phase, the Act phase predicts the 3D hand motion trajectory. For a training sample with $L+1$ sequential frames, we first lift the wrist locations into 3D space using the estimated camera intrinsics and the metric depth map, and construct the ground-truth future trajectory $\mathbf{T} \in \mathbb{R}^{L\times 3}$ from the subsequent L frames. We then average-pool the final-layer visual tokens of the initial frame into a global visual embedding, which is fed into a single-layer MLP prediction head to predict the future trajectory $\hat{\mathbf{T}} \in \mathbb{R}^{L\times 3}$. Finally, the motion planning objective $\mathcal{L}_{\text{motion}}$ is optimized by minimizing the ℓ_1 distance between $\hat{\mathbf{T}}$ and $\mathbf{T}$. By grounding action prediction in these rich 3D priors and refined attention, the policy acquires robust, generalized motion planning capabilities, establishing a strong foundation for downstream robot control.

4 Simulation Experiments

4.1 Setup

Evaluation Setup. We evaluate LSA on three simulation benchmarks: MetaWorld [41], DexArt [42], and Adroit [43]. MetaWorld evaluates tabletop manipulation with a Sawyer arm and gripper, while DexArt and Adroit focus on dexterous manipulation with multi-fingered hands. For MetaWorld, we use eight tasks covering different difficulty levels: *hammer*, *dial turn*, *assembly*, *hand insert*, *pick place wall*, *shelf place*, *disassemble*, and *stick pull*. For DexArt, we evaluate on *bucket* and *toilet*; for Adroit, we evaluate on *door*, *hammer*, and *pen*. Following prior methods, the pretrained visual encoder is frozen and combined with a lightweight behavior cloning policy head. We report success rates averaged over three random seeds. Detailed data collection, training, and evaluation protocols are provided in **Appendix B.1**.

Baselines. We compare LSA with three groups of baselines. First, we include general-purpose visual representations, including CLIP [44] and DINOv3 [45]. Second, we compare with representative robot visual representation pretraining methods, including MCR [46], VIP [47], R3M [19], VC-1 [48], HRP [8], MPI [9], and R3M-A [37]. Third, we include baselines that directly take 3D point clouds as inputs, including DP3 [49] and KAN-We-Flow [50]. For VIP, we additionally provide cropped target images following its original protocol. More baseline details are provided in **Appendix B.2**.

Table 1: Success rate (%) reported as mean ± standard deviation over 3 random seeds on MetaWorld, DexArt, and Adroit. * denotes that cropped target images are used as additional inputs. Task abbreviations: HA (Hammer), DT (Dial Turn), AS (Assembly), HI (Hand Insert), PPW (Pick Place Wall), ShP (Shelf Place), DI (Disassemble), StP (Stick Pull), BU (Bucket), TO (Toilet), DR (Door), HA-A (Adroit Hammer), and PE (Pen). Bold and underline denote the best and second-best results, respectively.

Method	**MetaWorld**								**DexArt**		**Adroit**		
	HA	DT	AS	HI	PPW	ShP	DI	StP	BU	TO	DR	HA-A	PE
Direct 3D Policy													
DP3 (*RSS'24*)	91.1 $\pm$2.8	84.9 $\pm$2.0	51.1 $\pm$3.3	33.8 $\pm$4.0	32.9 $\pm$3.0	48.4 $\pm$4.7	40.4 $\pm$3.5	39.1 $\pm$4.0	26.7 $\pm$1.7	71.7 $\pm$2.2	63.3 $\pm$1.7	**100.0** $\pm$0.0	40.6 $\pm$1.0
KAN-We-Flow (*ICRA'26*)	89.3 $\pm$1.4	89.8 $\pm$2.8	61.8 $\pm$4.1	58.7 $\pm$1.4	68.0 $\pm$4.0	70.2 $\pm$2.8	68.9 $\pm$2.0	66.7 $\pm$4.0	31.1 $\pm$2.6	<u>76.7</u> $\pm$1.7	<u>80.5</u> $\pm$2.5	**100.0** $\pm$0.0	<u>56.7</u> $\pm$3.4
2D Pretrain Policy													
CLIP (*ICML'21*)	95.7 $\pm$1.2	78.3 $\pm$4.6	88.7 $\pm$4.0	61.7 $\pm$5.7	<u>92.3</u> $\pm$1.2	72.3 $\pm$5.7	82.7 $\pm$3.2	70.3 $\pm$3.1	58.7 $\pm$3.5	42.0 $\pm$0.0	36.7 $\pm$4.6	75.0 $\pm$3.4	36.7 $\pm$4.3
R3M (*CoRL'22*)	26.7 $\pm$2.3	9.3 $\pm$2.3	32.0 $\pm$4.0	12.0 $\pm$0.0	26.7 $\pm$2.3	22.7 $\pm$2.3	24.0 $\pm$6.9	21.0 $\pm$2.6	51.3 $\pm$6.0	<u>76.7</u> $\pm$2.9	70.0 $\pm$2.1	51.7 $\pm$6.4	56.3 $\pm$4.1
VC-1 (*NeurIPS'23*)	91.7 $\pm$3.5	53.0 $\pm$8.7	73.3 $\pm$3.8	24.7 $\pm$0.6	25.7 $\pm$2.1	24.7 $\pm$2.1	70.3 $\pm$9.6	34.7 $\pm$2.3	10.0 $\pm$8.7	40.7 $\pm$8.1	58.3 $\pm$4.4	83.3 $\pm$5.8	38.3 $\pm$3.3
HRP (*CoRL'24*)	96.3 $\pm$2.9	57.0 $\pm$9.2	63.0 $\pm$7.2	19.0 $\pm$2.0	37.7 $\pm$2.3	45.7 $\pm$2.1	62.3 $\pm$12.7	41.7 $\pm$1.2	62.0 $\pm$5.0	29.7 $\pm$6.4	65.0 $\pm$4.8	<u>98.3</u> $\pm$2.6	38.3 $\pm$3.5
MPI (*RSS'24*)	<u>98.0</u> $\pm$1.7	40.7 $\pm$2.9	47.0 $\pm$13.1	17.3 $\pm$2.6	60.0 $\pm$3.7	<u>81.7</u> $\pm$8.3	68.0 $\pm$9.0	<u>75.0</u> $\pm$1.7	55.5 $\pm$2.0	71.7 $\pm$1.4	66.7 $\pm$3.9	53.7 $\pm$5.8	43.7 $\pm$3.1
MCR (*ICLR'25*)	75.7 $\pm$3.5	50.3 $\pm$9.6	42.7 $\pm$4.6	20.0 $\pm$0.0	28.0 $\pm$3.0	26.3 $\pm$2.1	34.0 $\pm$2.6	24.7 $\pm$4.0	59.0 $\pm$1.7	71.3 $\pm$4.0	60.0 $\pm$4.2	46.7 $\pm$5.1	36.7 $\pm$2.7
R3M-A (*CVPR'25*)	<u>98.0</u> $\pm$0.0	44.0 $\pm$6.9	32.3 $\pm$4.5	21.7 $\pm$2.9	31.0 $\pm$5.3	56.7 $\pm$6.0	50.3 $\pm$2.1	18.3 $\pm$2.1	<u>63.3</u> $\pm$2.9	72.7 $\pm$0.6	68.3 $\pm$3.6	**100.0** $\pm$0.0	45.0 $\pm$2.4
DINOv3 (*arXiv'25*)	**100.0** $\pm$0.0	90.7 $\pm$2.1	<u>96.3</u> $\pm$0.9	<u>75.7</u> $\pm$3.4	80.0 $\pm$5.0	71.7 $\pm$5.7	<u>85.0</u> $\pm$4.3	60.3 $\pm$5.9	26.0 $\pm$8.5	33.3 $\pm$10.3	70.0 $\pm$4.0	**100.0** $\pm$0.0	43.3 $\pm$3.3
VIP* (*ICML'25*)	96.0 $\pm$3.6	<u>91.0</u> $\pm$6.6	63.2 $\pm$5.0	68.0 $\pm$4.6	56.0 $\pm$1.0	73.7 $\pm$3.5	79.7 $\pm$3.5	71.0 $\pm$9.2	30.0 $\pm$1.7	64.4 $\pm$9.2	73.7 $\pm$3.2	96.3 $\pm$2.7	55.0 $\pm$2.6
LSA	**100.0** $\pm$0.0	**93.3** $\pm$2.3	**100.0** $\pm$0.0	**85.3** $\pm$2.3	**98.0** $\pm$2.0	**84.0** $\pm$0.0	**92.3** $\pm$1.5	**82.7** $\pm$1.5	**63.9** $\pm$3.5	**78.3** $\pm$1.6	**85.0** $\pm$1.7	**100.0** $\pm$0.0	**58.3** $\pm$1.3

Table 2: Robustness under visual perturbations.

Method	**Perturb.**	**SR(%)**	**Relative Drop**
DINOv3 [45]	Bright.	82.9 $\rightarrow$ 70.8	$\downarrow$14.6%
	Disturb.	82.9 $\rightarrow$ 55.0	$\downarrow$33.7%
R3M-A [37]	Bright.	36.8 $\rightarrow$ 20.6	$\downarrow$44.0%
	Disturb.	36.8 $\rightarrow$ 18.7	$\downarrow$49.2%
HRP [8]	Bright.	44.0 $\rightarrow$ 13.8	$\downarrow$68.6%
	Disturb.	44.0 $\rightarrow$ 31.0	$\downarrow$29.5%
LSA	Bright.	**92.2** $\rightarrow$ **85.3**	$\downarrow$7.5%
	Disturb.	**92.2** $\rightarrow$ **75.0**	$\downarrow$18.7%

Table 3: Ablation study of LSA's architectural components. **MP**: motion planning; **RA**: representation alignment; **SG**: structural guidance; **AG**: attentive guidance.

	MP	**RA**	**SG**	**AG**	**SR**	**Gain**
Exp.1	✗	✗	✗	✗	73.3	+0
Exp.2	✓	✗	✗	✗	81.0	+7.7
Exp.3	✓	✓	✗	✗	85.2	+11.9
Exp.4	✓	✗	✓	✗	85.8	+12.5
Exp.5	✓	✗	✗	✓	86.3	+13.0
Exp.6	✓	✓	✓	✗	89.0	+15.7
Exp.7	✓	✗	✓	✓	88.4	+15.1
Exp.8	✓	✓	✓	✓	**92.2**	**+18.9**

4.2 Results

As shown in Table 1, LSA obtains consistently strong results across MetaWorld, DexArt, and Adroit. On MetaWorld, LSA performs particularly well on challenging tasks such as *pick-place-wall* and *disassemble*, where accurate spatial perception is critical. On DexArt and Adroit, LSA also achieves competitive or best performance on articulated-object and dexterous manipulation tasks. Compared with direct 3D policy baselines such as DP3 [49] and KAN-We-Flow [50], LSA attains these results without using explicit 3D observations during downstream policy learning, suggesting that the proposed pretraining effectively injects useful 3D-aware representations into the visual encoder. We further evaluate robustness by reducing image brightness by 40% and adding random colored circular occlusions during testing. Tab. 2 shows that LSA suffers smaller drops under these visual perturbations, decreasing by only 7.5% under brightness reduction and 18.7% under visual disturbance. In contrast, the compared baselines show much larger degradation, indicating that LSA is more robust to superficial appearance changes.

4.3 Ablation Results

Ablation studies are conducted on four MetaWorld tasks, including *dial turn*, *hand insert*, *pick place wall*, and *disassemble*. We report the average evaluation success rate across these tasks.

Architecture Design. As shown in Table 3, the model without human-video pretraining achieves an average success rate of 73.3%. Adding the motion planning objective improves the success rate to 81.0%, indicating that future 3D hand trajectories provide transferable action-relevant supervision. Built upon this objective, each component in LSA further improves the learned representation: representation alignment, structural guidance, and attentive guidance individually improve the success rate to 85.2%, 85.8%, and 86.3%, respectively. Combining representation alignment with structural

Table 4: Ablation study on key LSA variants.

Configuration	SR(%)
Full Model (Ours)	**92.2**
Motion Supervision	
Only 3D Hand Motion	81.0
Only 2D Hand Motion	77.3
3D → 2D Hand Motion	56.3 (↓ 35.9)
Aligned Representation	
Only VGGT	89.8 (↓ 2.4)
Only Depth Pro	90.3 (↓ 1.9)
Alignment Weight	
None	89.3 (↓ 2.9)
Condition only on $\mathbf{c}_l$	91.2 (↓ 1.0)
Condition only on $\mathbf{c}_m$	90.1 (↓ 2.1)
Attentive Guidance	
Target: Each head	86.3 (↓ 5.9)
Super.: Random head	83.5 (↓ 8.7)
Super.: Worst-matching head	81.8 (↓ 10.4)
Super.: Hand heatmap	84.3 (↓ 7.9)

Table 5: Quality analysis of 3D hand pseudo-labels. ADE denotes the average displacement error in meters.

(a) Wrist 3D lifting accuracy.

Depth	Intrinsics	H2O ADE↓	DexYCB ADE↓
GT	GT	0.0000	0.0000
GT	VGGT	0.0093	0.0079
Depth Pro	GT	0.0306	0.0333
Depth Pro	VGGT	0.0393	0.0364
VGGT	GT	0.0974	0.1147
VGGT	VGGT	0.0966	0.1155

(b) Effect of trajectory supervision quality.

Trajectory Supervision	SR(%)
VGGT-lifted trajectory	88.6 (↓ 3.6)
Depth Pro + VGGT (Ours)	92.2
DexYCB GT trajectory	93.5 (↑ 1.3)

guidance further increases the success rate to 89.0%, suggesting that depth-aware reconstruction further strengthens the dense 3D features learned from representation alignment. Finally, the complete LSA framework achieves the best success rate of 92.2%, improving the non-pretrained baseline by 18.9%. These results show that the three phases of LSA provide complementary supervision, empowering the policy with 3D-aware perception and interaction-aware motion planning abilities learned from human videos.

Analysis of LSA Variants. Tab. 4 analyzes key design choices in LSA. For motion supervision, using only 2D hand motion achieves 77.3%, slightly below the 3D motion planning baseline of 81.0%. However, when the model receives 3D geometric supervision in the Lift and See phases but uses 2D trajectory prediction in the Act phase, the performance drops further. We attribute this to the semantic inconsistency between the lifted 3D representation space and the image-plane 2D motion objective. Using only VGGT or Depth Pro for representation alignment is inferior to using both, indicating that different 3D FMs provide complementary dense geometric cues. Removing dynamic alignment weights or conditioning them on a single layer also degrades performance, showing the need for adaptive multi-teacher weighting with multi-layer geometric context. For attentive guidance, supervising the attention of all shallow heads suppresses attention diversity and thus degrades performance. In addition, using a random head, the worst-matching head, or the hand heatmap as the supervision target also reduces performance, indicating that the internally emerged attention target selected by the hand heatmap more effectively captures interaction regions.

Quality of 3D Pseudo-labels. Tab. 5 evaluates the reliability of the pseudo-3D hand labels used by LSA. To analyze the error, we separately evaluate lifting accuracy on H2O and DexYCB, using 240 valid wrist samples from 12 scenes for each dataset; more details are provided in **Appendix B.3**. Depth Pro depth with VGGT intrinsics substantially reduces wrist lifting errors compared with VGGT depth variants, e.g., 0.0364 m vs. 0.1155 m on DexYCB. When replacing only the Act-phase trajectory labels, noisier VGGT-lifted trajectories reduce SR from 92.2% to 88.6%, while DexYCB GT trajectories only improve SR to 93.5%. The limited improvement from replacing our pseudo-trajectories with GT trajectories suggests that the adopted Depth Pro+VGGT pipeline provides reasonably reliable supervision in our setting. We also validate the effectiveness of our labeling and pretraining strategy on more diverse unlabeled datasets, such as Ego4D (**Appendix C.5**).

More Analyses. Given the limited space, we provide additional analyses in the **Appendix**, including detailed experimental protocols (**Appendix B**), comparisons with direct 3D-FM features and additional pretrained baselines (**Appendix C.1, C.9**), teacher-selection studies (**Appendix C.2**), data scaling and diversity studies (**Appendix C.3, C.5**), parameter update strategies (**Appendix C.4**), hyper-parameter and layer-selection ablations (**Appendix C.6, C.7**), training convergence (**Appendix C.8**), computational overhead (**Appendix C.11**), and more visualizations of attentive guidance and task trajectories (**Appendix E**).

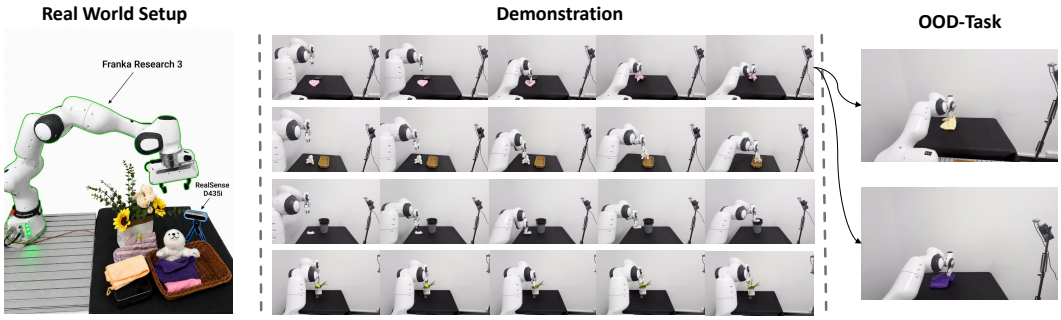


Figure 4: Overview of our real-world experiment setup. The figure shows our robotic platform and sensors (left), the execution of the four tasks under the in-distribution setup (middle), and the modifications applied for the out-of-distribution setup (right).

Table 6: Real-world evaluation with the Franka robot across four tasks. For *Wipe PC*, *U1* and *U2* denote two unseen towel colors.

Method	Pluck Flower	Pick Trash	Place Toy	Wipe PC	Wipe PC U1	Wipe PC U2	Avg.
DINOv3 [45]	20.0	30.0	20.0	30.0	10.0	20.0	21.7
R3M-A [37]	10.0	30.0	30.0	40.0	0.0	10.0	20.0
VIP [47]	**50.0**	40.0	50.0	40.0	20.0	20.0	36.7
HRP [8]	40.0	40.0	60.0	50.0	20.0	40.0	41.7
Ours	**50.0**	**70.0**	**80.0**	**80.0**	**70.0**	**60.0**	**68.3**

5 Real-World Experiments

5.1 Setup

We further evaluate LSA in real-world manipulation using a 7-DoF Franka Research 3 robot equipped with a 1-DoF parallel gripper. RGB observations are captured by a RealSense D435i camera. As shown in Fig. 4, we consider four daily-scenario manipulation tasks: *Wipe PC*, *Pick Trash*, *Place Toy*, and *Pluck Flower*. For each task, we collect 30 demonstrations with variations in the initial poses of the robot and objects. We evaluate each policy over ten trials per task and report the success rate (%).

To evaluate out-of-distribution (OOD) generalization, we further construct two unseen variants of *Wipe PC*. The policy is trained with one towel color and tested with two previously unseen towel colors, denoted as *unseen1* and *unseen2*. This setting tests whether the learned representation can generalize beyond superficial appearance changes. More implementation details are provided in **Appendix B.4**.

5.2 Evaluation Results

Tab. 6 shows that LSA achieves the highest average success rate among all compared methods, with 68.3% vs. 41.7% for the strongest baseline HRP. LSA outperforms the second-best method on *Pick Trash* (70.0% vs. 40.0%), *Place Toy* (80.0% vs. 60.0%), and *Wipe PC* (80.0% vs. 50.0%), while matching the best result on *Pluck Flower* (50.0%). On the two OOD *Wipe PC* settings, LSA obtains 70.0% and 60.0%, whereas the best baselines achieve 20.0% and 40.0%; notably, R3M-A drops to 0.0% on *Wipe PC U1*. These results suggest that LSA transfers better to real-world manipulation. On one hand, its human-video pretraining learns 3D-aware representations from 2D observations, reducing the burden of acquiring spatial relations from limited robot demonstrations. On the other hand, its interaction-focused guidance helps the policy preserve task-relevant hand-object relations under appearance changes.

6 Conclusion

In this paper, we propose LSA, a hierarchical robot policy pretraining framework that systematically decomposes the learning process into three synergistic phases: Lift, See, and Act. Unlike prior works that rely solely on sparse 3D trajectories, LSA fully exploits the multi-tiered knowledge of 3D FMs by aligning policy features with dense, intermediate geometric representations. To bridge

the gap between implicit features and physical execution, we introduce a dual-guidance mechanism in the See phase, combining explicit structural reconstruction with attentive guidance to focus on task-relevant regions. Finally, the Act phase grounds the learned representation in future 3D hand motion, which serves as an action-relevant proxy for downstream robot manipulation. **Limitations.** LSA shares several limitations inherent to human-video-based robot pretraining methods due to the human-robot embodiment gap. **Broader Impacts**. Looking forward, this work opens avenues for scaling multi-tiered 3D knowledge extraction from human videos to broader robot manipulation scenarios, with the potential to further enhance Vision-Language-Action (VLA) model pretraining. To the best of our knowledge, this work has no obvious negative social impacts.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant 62476099 and 62076101, Guangdong Basic and Applied Basic Research Foundation under Grant 2024B1515020082, the 2026 Guangdong Provincial Special Fund for Industrial Foundation Reengineering (Multimodal Adaptive Embodied AI Large Model Project), the Key-Area Research and Development Program of Guangdong Province under Grant 2025B0909020002 and 2024B0101040004, the TCL Young Scholars Program, and the High Performance Computing Platform of South China University of Technology.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[3] Chubin Zhang, Jianan Wang, Zifeng Gao, Yue Su, Tianru Dai, Cai Zhou, Jiwen Lu, and Yansong Tang. Clap: Contrastive latent action pretraining for learning vision-language-action models from human videos. *arXiv preprint arXiv:2601.04061*, 2026.

[4] Shenyuan Gao, William Liang, Kaiyuan Zheng, Ayaan Malik, Seonghyeon Ye, Sihyun Yu, Wei-Cheng Tseng, Yuzhu Dong, Kaichun Mo, Chen-Hsuan Lin, et al. Dreamdojo: A generalist robot world model from large-scale human videos. *arXiv preprint arXiv:2602.06949*, 2026.

[5] Zhiyuan Feng, Qixiu Li, Huizhi Liang, Rushuai Yang, Yichao Shen, Zhiying Du, Zhaowei Zhang, Yu Deng, Li Zhao, Hao Zhao, et al. From human videos to robot manipulation: A survey on scalable vision-language-action learning with human-centric data. *Authorea Preprints*, 2026.

[6] Alexey Gavryushin, Xi Wang, Robert JS Malate, Chenyu Yang, Davide Liconti, René Zurbrügg, Robert K Katzschmann, and Marc Pollefeys. Maple: Encoding dexterous robotic manipulation priors learned from egocentric videos. *arXiv preprint arXiv:2504.06084*, 2025.

[7] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023.

[8] Mohan Kumar Srirama, Sudeep Dasari, Shikhar Bahl, and Abhinav Gupta. Hrp: Human affordances for robotic pre-training. In *Robotics: Science and Systems*, 2024.

[9] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. In *Robotics: Science and Systems*, 2024.

[10] Hao Luo, Yicheng Feng, Wanpeng Zhang, Sipeng Zheng, Ye Wang, Haoqi Yuan, Jiazheng Liu, Chaoyi Xu, Qin Jin, and Zongqing Lu. Being-h0: Vision-language-action pretraining from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

[11] Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. Vidbot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 27661–27672, 2025.

[12] Qixiu Li, Yu Deng, Yaobo Liang, Lin Luo, Lei Zhou, Chengtang Yao, Lingqi Zeng, Zhiyuan Feng, Huizhi Liang, Sicheng Xu, et al. Scalable vision-language-action model pretraining for robotic manipulation with real-life human activity videos. *arXiv preprint arXiv:2510.21571*, 2025.

[13] Juncheng Mu, Sizhe Yang, Yiming Bao, Hojin Bae, Tianming Wei, Linning Xu, Boyi Li, Huazhe Xu, and Jiangmiao Pang. Deximit: Learning bimanual dexterous manipulation from monocular human videos. *arXiv preprint arXiv:2602.10105*, 2026.

[14] Hongyi Chen, Tony Dong, Tiancheng Wu, Liquan Wang, Yash Jangir, Yaru Niu, Yufei Ye, Homanga Bharadhwaj, Zackory Erickson, and Jeffrey Ichnowski. Dexterous manipulation policies from rgb human videos via 3d hand-object trajectory reconstruction. *arXiv preprint arXiv:2602.09013*, 2026.

[15] 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.

[16] Alexey Bochkovskiy, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan Richter, and Vladlen Koltun. Depth pro: Sharp monocular metric depth in less than a second. In *International Conference on Learning Representations*, volume 2025, pages 75602–75637, 2025.

[17] Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. Moge-2: Accurate monocular geometry with metric scale and sharp details. *arXiv preprint arXiv:2507.02546*, 2025.

[18] Jinglei Zhang, Jiankang Deng, Chao Ma, and Rolandos Alexandros Potamias. Hawor: World-space hand motion reconstruction from egocentric videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1805–1815, 2025.

[19] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[20] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation via value-implicit pre-training. *arXiv preprint arXiv:2210.00030*, 2022.

[21] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[22] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. *arXiv preprint arXiv:2410.11758*, 2024.

[23] Jiange Yang, Yansong Shi, Haoyi Zhu, Mingyu Liu, Kaijing Ma, Yating Wang, Gangshan Wu, Tong He, and Limin Wang. Como: Learning continuous latent motion from internet videos for scalable robot learning. *arXiv preprint arXiv:2505.17006*, 2025.

[24] 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 *International Conference on Machine Learning*, pages 24328–24346. PMLR, 2025.

[25] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024.

[26] Sandeep Routray, Hengkai Pan, Unnat Jain, Shikhar Bahl, and Deepak Pathak. Vipra: Video prediction for robot actions. *arXiv preprint arXiv:2511.07732*, 2025.

[27] Gen Li, Nikolaos Tsagkas, Jifei Song, Ruaridh Mon-Williams, Sethu Vijayakumar, Kun Shao, and Laura Sevilla-Lara. Learning precise affordances from egocentric videos for robotic manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10581–10591, 2025.

[28] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Masquerade: Learning from in-the-wild human videos using data-editing. *arXiv preprint arXiv:2508.09976*, 2025.

[29] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: Modeling and capturing hands and bodies together. *arXiv preprint arXiv:2201.02610*, 2022.

[30] Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, et al. Egovla: Learning vision-language-action models from egocentric human videos. *arXiv preprint arXiv:2507.12440*, 2025.

[31] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017.

[32] Hongzhe Bi, Lingxuan Wu, Tianwei Lin, Hengkai Tan, Zhizhong Su, Hang Su, and Jun Zhu. H-rdt: Human manipulation enhanced bimanual robotic manipulation. *arXiv preprint arXiv:2507.23523*, 2025.

[33] Xiongyi Cai, Ri-Zhao Qiu, Geng Chen, Lai Wei, Isabella Liu, Tianshu Huang, Xuxin Cheng, and Xiaolong Wang. In-n-on: Scaling egocentric manipulation with in-the-wild and on-task data. *arXiv preprint arXiv:2511.15704*, 2025.

[34] Tomoya Yoshida, Shuhei Kurita, Taichi Nishimura, and Shinsuke Mori. Developing vision-language-action model from egocentric videos. *arXiv preprint arXiv:2509.21986*, 2025.

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

[36] Younggyo Seo, Danijar Hafner, Hao Liu, Fangchen Liu, Stephen James, Kimin Lee, and Pieter Abbeel. Masked world models for visual control. In *Conference on Robot Learning*, pages 1332–1344. PMLR, 2023.

[37] Jiaming Zhou, Teli Ma, Kun-Yu Lin, Zifan Wang, Ronghe Qiu, and Junwei Liang. Mitigating the human-robot domain discrepancy in visual pre-training for robotic manipulation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 22551–22561, 2025.

[38] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine intelligence 15*, pages 103–129, 1995.

[39] Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

[40] Jiajun Cao, Yuan Zhang, Tao Huang, Ming Lu, Qizhe Zhang, Ruichuan An, Ningning Ma, and Shanghang Zhang. Move-kd: Knowledge distillation for vlms with mixture of visual encoders. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19846–19856, 2025.

[41] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020.

[42] Chen Bao, Helin Xu, Yuzhe Qin, and Xiaolong Wang. Dexart: Benchmarking generalizable dexterous manipulation with articulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21190–21200, 2023.

[43] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[45] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[46] Guangqi Jiang, Yifei Sun, Tao Huang, Huanyu Li, Yongyuan Liang, and Huazhe Xu. Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets. *arXiv preprint arXiv:2410.22325*, 2024.

[47] Zhuoling Li, Liangliang Ren, Jinrong Yang, Yong Zhao, Xiaoyang Wu, Zhenhua Xu, Xiang Bai, and Hengshuang Zhao. Vip: Vision instructed pre-training for robotic manipulation. *arXiv preprint arXiv:2410.07169*, 2024.

[48] Arjun Majumdar, Karmesh Yadav, Sergio Arnaud, Jason Ma, Claire Chen, Sneha Silwal, Aryan Jain, Vincent-Pierre Berges, Tingfan Wu, Jay Vakil, et al. Where are we in the search for an artificial visual cortex for embodied intelligence? *Advances in Neural Information Processing Systems*, 36:655–677, 2023.

[49] 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.

[50] Zhihao Chen, Yiyuan Ge, and Ziyang Wang. Kan we flow? advancing robotic manipulation with 3d flow matching via kan & rwkv. *arXiv preprint arXiv:2602.01115*, 2026.

[51] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.

[52] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[53] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[54] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

[55] Himanshu Gaurav Singh, Antonio Loquercio, Carmelo Sferrazza, Jane Wu, Haozhi Qi, Pieter Abbeel, and Jitendra Malik. Hand-object interaction pretraining from videos. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3352–3360. IEEE, 2025.

[56] Yu-Wei Chao, Wei Yang, Yu Xiang, Pavlo Molchanov, Ankur Handa, Jonathan Tremblay, Yashraj S Narang, Karl Van Wyk, Umar Iqbal, Stan Birchfield, et al. Dexycb: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9044–9053, 2021.

[57] Yingdong Hu, Renhao Wang, Li Erran Li, and Yang Gao. For pre-trained vision models in motor control, not all policy learning methods are created equal. In *International Conference on Machine Learning*, pages 13628–13651. PMLR, 2023.

[58] Nikolaos Tsagkas, Andreas Sochopoulos, Duolikun Danier, Chris Xiaoxuan Lu, and Oisin Mac Aodha. When pre-trained visual representations fall short: Limitations in visuo-motor robot learning. *arXiv e-prints*, pages arXiv–2502, 2025.

[59] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[60] Che Wang, Xufang Luo, Keith Ross, and Dongsheng Li. Vrl3: A data-driven framework for visual deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:32974–32988, 2022.

[61] 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.

[62] Taein Kwon, Bugra Tekin, Jan Stühmer, Federica Bogo, and Marc Pollefeys. H2o: Two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10138–10148, 2021.

[63] Wentao Bao, Lele Chen, Libing Zeng, Zhong Li, Yi Xu, Junsong Yuan, and Yu Kong. Uncertainty-aware state space transformer for egocentric 3d hand trajectory forecasting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13702–13711, 2023.

[64] 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.

[65] Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. Unidepthv2: Universal monocular metric depth estimation made simpler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[66] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022.

[67] Yi Chen, Yuying Ge, Weiliang Tang, Yizhuo Li, Yixiao Ge, Mingyu Ding, Ying Shan, and Xihui Liu. Moto: Latent motion token as the bridging language for learning robot manipulation from videos. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19752–19763. IEEE, 2025.

Appendix Contents

A. Additional Methodology Details 15
A.1 Overall Pretraining Objective 15
A.2 Motion Planning Loss 15
A.3 Dynamic Weight Allocation 16
A.4 More Details of Depth Reconstruction 16
A.5 Attentive Guidance Details 16
A.6 Data Cleaning and 3D Lifting 17
A.7 Hand Prior Heatmap 18
A.8 Pretraining Implementation Details 18

B. Experimental Protocols 18
B.1 Simulation Evaluation Setup 18
B.2 Baseline Details 19
B.3 3D Wrist Lifting Evaluation Set Construction 19
B.4 Real-World Implementation Details 20

C. Additional Experimental Results 20
C.1 Comparison with Direct 3D-FM Features 20
C.2 Selection of 3D FMs 21
C.3 Data Scaling and Sample Efficiency 21
C.4 Parameter Update Strategies 21
C.5 Pretraining Data Diversity 22
C.6 Hyperparameter Selection 22
C.7 Layer Selection 22
C.8 Training Loss Curves 23
C.9 Additional Baseline Comparisons 23
C.10 Additional Real-World Generalization Experiments 23
C.11 Computational Overhead 24

D. Further Discussion on Limitations 24

E. Additional Visualizations 25
E.1 Simulation Trajectory Visualizations 25
E.2 Real-World Trajectory Visualizations 25
E.3 More Attentive Guidance Visualizations 25

A Additional Methodology Details

A.1 Overall Pretraining Objective

The overall supervision loss consists of four terms: the motion planning loss $\mathcal{L}_{\text{motion}}$, the representation alignment loss $\mathcal{L}_{\text{align}}$, the depth reconstruction loss $\mathcal{L}_{\text{recon}}$, and the attentive guidance loss $\mathcal{L}_{\text{att}}$. The total loss in the pretraining stage is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{motion}} + \lambda_g \cdot (\mathcal{L}_{\text{align}} + \mathcal{L}_{\text{recon}} + \lambda_a \cdot \mathcal{L}_{\text{att}}), \tag{6}$$

where λ_g balances the geometry-related supervision terms and λ_a controls the strength of attentive guidance. In our experiments, we set $\lambda_g = 0.3$ and $\lambda_a = 0.5$.

A.2 Motion Planning Loss

We provide the explicit formulation of the motion planning objective and define all symbols below:

$$\mathcal{L}_{\text{motion}} = \frac{1}{3L} \sum_{t=1}^{L} \left\| \hat{\mathbf{T}}_t - \mathbf{T}_t \right\|_1 = \frac{1}{3L} \sum_{t=1}^{L} \left(|\hat{x}_t - x_t| + |\hat{y}_t - y_t| + |\hat{z}_t - z_t| \right), \tag{7}$$

where L is the number of predicted future timesteps; t indexes these timesteps; $\hat{\mathbf{T}}_t = (\hat{x}_t, \hat{y}_t, \hat{z}_t) \in \mathbb{R}^3$ and $\mathbf{T}_t = (x_t, y_t, z_t) \in \mathbb{R}^3$ denote the predicted and lifted target 3D wrist positions at timestep t, respectively; and $\| \cdot \|_1$ denotes the sum of absolute errors over the three spatial coordinates. The objective then averages the absolute errors over the L future timesteps and the three spatial coordinates.

A.3 Dynamic Weight Allocation

We design the dynamic weight assignment mechanism to efficiently harness representations from multiple 3D FMs, as the optimal contribution of each FM fundamentally varies across diverse manipulation scenarios. To adaptively learn the most valuable knowledge, we formulate a condition query leveraging the `[CLS]` tokens $\mathbf{c}_l$ and $\mathbf{c}_m$ from the representation alignment layer (l-th layer) and the depth reconstruction layer (m-th layer), respectively. As summarized in **Algorithm** 1, this condition query is used to compute the scaled dot-product similarity with geometric tokens and produce the dynamic teacher weights. First, the `[CLS]` token naturally serves as a compact global representation, effectively encapsulating the spatial semantics of the entire scene. Second, the two selected `[CLS]` tokens capture geometry-related semantics. Specifically, the l-th layer is directly aligned with the implicit intermediate representations of the 3D FMs, thus capturing **implicit geometric information**. Conversely, the m-th layer is explicitly optimized for the depth reconstruction auxiliary task, thereby encoding the **explicit 3D spatial geometry** of the physical world. By aggregating them, we formulate a condition query that achieves a comprehensive geometric awareness, thereby better modulating the teacher weights. Here, $\mathrm{MeanPool}$ denotes channel-group mean pooling: it reshapes the last feature dimension into groups of $D = 512$ and averages over the group axis while preserving the batch and spatial-token dimensions.

Algorithm 1 Dynamic Weight Allocation

Input: $\mathbf{c}_l, \mathbf{c}_m$ from l-th and m-th layers; geometric tokens $\mathbf{E}'_c, \mathbf{E}'_d$.
Output: Global modulation weights w_c, w_d.

$\mathbf{Q} \leftarrow \mathrm{Unsqueeze}(\mathrm{MeanPool}(\mathrm{Concat}(\mathbf{c}_l, \mathbf{c}_m)), 1) \in \mathbb{R}^{B\times 1\times D}$ {Query}
for all $j \in \{c, d\}$ **do**
  $\hat{\mathbf{E}}'_j \leftarrow \mathrm{MeanPool}(\mathbf{E}'_j) \in \mathbb{R}^{B\times N_a\times D}$ {Channel-group pooling}
  $\mathbf{S}_j \leftarrow \mathrm{Squeeze}\Big((\mathbf{Q}\hat{\mathbf{E}}'^{\top}_j)/\sqrt{D}, 1\Big) \in \mathbb{R}^{B\times N_a}$
  $\mathbf{s}_j \leftarrow \frac{1}{N_a}\sum_{i=1}^{N_a} \mathbf{S}_{j,:,i} \in \mathbb{R}^B$ {Similarity}
end for
$\mathbf{G} \leftarrow \mathrm{Softmax}([\mathbf{s}_c, \mathbf{s}_d]) \in \mathbb{R}^{B\times 2}$ {Gating}
$w_c \leftarrow \mathrm{avg}(\mathbf{G}_{:,0})$; $w_d \leftarrow \mathrm{avg}(\mathbf{G}_{:,1})$ {Aggregation}
return w_c, w_d

A.4 More Details of Depth Reconstruction

The decoder takes the 196 patch tokens from encoder layer $m = 6$, with shape $B \times 196 \times 768$, corresponding to a 14×14 spatial grid. Each token is projected from 768 to 512 dimensions and augmented with a fixed 2D sine–cosine positional embedding. The resulting sequence is processed by eight self-attention Transformer blocks with 8 heads, preserving its shape of $B \times 196 \times 512$. A linear head maps each token to 256 depth values representing one 16×16 patch, producing $B \times 196 \times 256$. A non-overlapping Fold operation with kernel size and stride 16 assembles these patches into a $B \times 1 \times 224 \times 224$ depth map. Thus, depth is predicted patchwise and supervised pixelwise over the full 224×224 resolution.

We resize the metric-depth target to 224×224, map its pixel values to $[0, 1]$, and compute the mean squared error over all spatial locations:

$$\mathcal{L}_{\mathrm{recon}} = \frac{1}{BHW}\sum_{b=1}^{B}\sum_{p=1}^{HW}\Big(\hat{D}_{(b,p)} - D_{g(b,p)}\Big)^2, \tag{8}$$

where B is the batch size; H and W are the height and width of the depth map; b and p index the samples and spatial pixels, respectively; $\hat{\mathbf{D}}$ denotes the predicted depth map; and $\mathbf{D}_g$ denotes the metric-depth target. The standard setting uses neither an invalid-pixel mask nor confidence filtering.

A.5 Attentive Guidance Details

We further provide details and visual evidence for attentive guidance. We treat the `[CLS]` tokens from different ViT layers as distinct entities, since they encode hierarchical global semantics. As

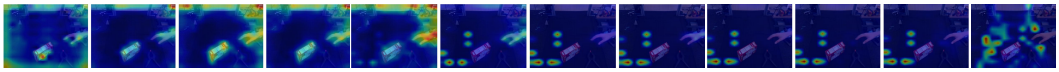


Figure 5: Attention maps of the [CLS] token from layers 1 to 12 in LSA without attentive guidance. Shallow layers tend to focus on local regions such as hands and objects, while deeper layers become more dispersed and abstract.

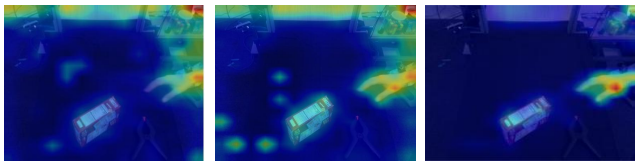


Figure 6: Comparison of attention maps. **Left:** averaged attention map over the first $R = 6$ layers of the original ViT. **Middle:** averaged attention map after motion-planning pretraining. **Right:** target attention map selected by the hand prior. The selected target attends to both the hand and the interacting object.

shown in Fig. 5, the attention maps associated with these layer-specific [CLS] tokens also evolve across layers. Shallow layers tend to focus on local regions such as hands and objects, whereas deeper layers become more abstract and dispersed. Motivated by this layer-wise pattern, we apply attentive guidance to the shallow layers. As shown in Fig. 6, the original ViT produces relatively dispersed shallow attention, while motion planning encourages clearer attention to the hand region and further induces emergent attention to the interacting object. This suggests that future hand trajectories provide supervision signals that implicitly capture hand-object spatial relations. We therefore use the hand prior to select the internal attention map most aligned with the hand region; as shown in Fig. 6, the selected target also extends to the interacting object. Compared with supervising each individual head, our method guides only the head-averaged attention map of each shallow layer, aligning the overall spatial focus while preserving inter-head diversity.

A.6 Data Cleaning and 3D Lifting

We use DexYCB as the main pretraining data source. Our data cleaning process consists of three steps.

First, we detect the hand pose in each frame. Due to occlusions and detection failures, we remove a video if the hand is not detected for ≥ 3 consecutive frames.

Second, given a per-frame metric depth sequence $\{z_t\}_{t=1}^{T}$ in a video, we compute its temporal standard deviation $\sigma_z = \mathrm{std}(\{z_t\})$ and frame-to-frame depth differences $\Delta z_t = |z_t - z_{t-1}|$ for $t = 2, \ldots, T$. A video is discarded as low quality if it is globally unstable, i.e., $\sigma_z > 0.09$ and $\frac{1}{T-1}\sum_{t=2}^{T} \Delta z_t > 0.02$, or if it contains an abrupt jump, i.e., $\max_t \Delta z_t > 0.15$. These thresholds are set empirically to remove unstable lifted trajectories.

Third, we apply Gaussian smoothing to the metric depth sequence $\{z_t\}$ with $\sigma_{\mathrm{t}} = 3.0$ and obtain the smoothed depth sequence $\{\tilde{z}_t\}$. The first and last depth values are kept unchanged to avoid boundary artifacts.

After data cleaning, we lift each 2D wrist keypoint (u_t, v_t) into 3D space using the smoothed metric depth $\tilde{z}_t$ and camera intrinsics. Let (f_x, f_y) denote the focal lengths and (c_x, c_y) denote the principal point. The lifted 3D wrist point $\mathbf{p}_t$ in the camera coordinate system is computed by the pinhole camera model:

$$\mathbf{p}_t = \begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} = \begin{bmatrix} (u_t - c_x)\tilde{z}_t / f_x \\ (v_t - c_y)\tilde{z}_t / f_y \\ \tilde{z}_t \end{bmatrix}, \quad t = 1, \ldots, T. \tag{9}$$

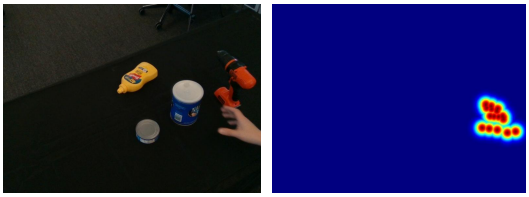


Figure 7: **Left:** Original image. **Right:** Generated hand prior heatmap.

A.7 Hand Prior Heatmap

To construct the spatial hand prior used in attentive guidance, we employ MediaPipe [51] to extract the 2D image coordinates of 21 hand joints. Let $\mathbf{p}_k = (x_k, y_k)$ denote the 2D coordinate of the k-th hand joint. We place a Gaussian kernel at each detected keypoint and aggregate the responses using the maximum value at each pixel location:

$$H(x, y) = \max_{k \in \{1,\dots,21\}} \exp\left(-\frac{(x - x_k)^2 + (y - y_k)^2}{2\sigma_{\mathrm{h}}^2}\right). \tag{10}$$

Here, σ_{h} controls the spatial spread of the local response. We further convert H into a softened spatial prior using a temperature-scaled Softmax:

$$h(x, y) = \frac{\exp(H(x, y)/\tau_{\mathrm{h}})}{\sum_{i,j} \exp(H(i, j)/\tau_{\mathrm{h}})}. \tag{11}$$

Here, τ_{h} denotes the Softmax temperature. This operation smooths the distribution and allows the hand prior to cover not only exact keypoint locations but also nearby hand regions and interaction context. We then min–max normalize h to $[0, 1]$ and resize it to the 14×14 patch grid before computing the attention-alignment scores. In our experiments, we set $\sigma_{\mathrm{h}} = 12$ and $\tau_{\mathrm{h}} = 5$. Fig. 7 shows the results.

A.8 Pretraining Implementation Details

We adopt a pretrained ViT-B [52] initialized on ImageNet [53] as the visual encoder. To enable parameter-efficient adaptation, we keep the original ViT-B parameters frozen and apply LoRA [54] with rank $r = 8$ and scaling factor $\alpha = 16$. Following our hierarchical design, we perform representation alignment at the $l = 3$ layer of the visual encoder, while the visual tokens from the $m = 6$ layer are routed to an 8-layer Transformer decoder for metric depth reconstruction. Attentive guidance is applied to the first $R = 6$ layers. Attentive guidance is enabled after a 10-epoch warm-up with the motion planning objective, so that the model can first develop initial task-relevant attention patterns before the target attention map is selected. For the Act phase, the model predicts the future 3D wrist trajectory for the next $L = 3$ frames.

Following prior work [55], we use DexYCB [56] to pretrain LSA. After data cleaning, we retain 20K video clips for pretraining. Each clip consists of consecutive frames that depict human hands performing tasks and moving within the scene. Optimization is performed using Adam with a cosine learning rate schedule and an initial learning rate of 5×10^{-4}. The model is trained for 50 epochs on NVIDIA RTX 4090 GPUs with 24GB memory each, using a batch size of 16 per GPU.

B Experimental Protocols

This section provides detailed experimental protocols for simulation benchmarks, baseline implementations, 3D wrist lifting evaluation, and real-world experiments.

B.1 Simulation Evaluation Setup

We provide detailed simulation settings for the three benchmarks used in the main paper: Meta-World [41], DexArt [42], and Adroit [43].

MetaWorld. MetaWorld provides tabletop manipulation tasks with a Sawyer arm and gripper. We select eight tasks spanning different difficulty levels [36]: regular tasks including *hammer* and *dial turn*, hard tasks including *assembly* and *hand insert*, and very hard tasks including *pick place wall*, *shelf place*, *disassemble*, and *stick pull*. For each task, we use scripted policies to collect 25 expert demonstrations. Following prior works [57, 58], the maximum rollout length is strictly limited to 175 steps, which makes the setting more challenging by requiring faster and more accurate task completion. The behavior cloning policy is trained for 20,000 steps with a batch size of 32 and evaluated every 2,000 steps.

DexArt. DexArt focuses on dexterous manipulation of articulated objects using an XArm6 arm and an Allegro hand. We evaluate on two tasks: *bucket* and *toilet*. For each task, we collect 100 expert demonstrations using pretrained reinforcement learning policies. Each rollout is capped at 100 steps. The behavior cloning policy is trained for 3,000 epochs with a batch size of 64 and evaluated every 200 epochs. The pretrained visual encoder is frozen during downstream policy learning, and only the policy head is trained.

Adroit. Adroit simulates a multi-fingered Shadow Hand in MuJoCo [59] and contains contact-rich dexterous manipulation tasks involving rigid and articulated objects. We evaluate on three tasks: *door*, *hammer*, and *pen*. For each task, we collect 10 expert demonstrations using a pretrained reinforcement learning policy [60]. Each demonstration contains 100 steps. The training and evaluation protocols follow the DexArt setting. We include this benchmark to further evaluate whether LSA can transfer to complex dexterous manipulation with fine-grained hand-object interactions.

Reporting Protocol. For each evaluation checkpoint, we compute the success rate over all rollouts. For each random seed, we average the success rates of the top-3 performing checkpoints. We then report the mean and sample standard deviation over three random seeds, rounded to one decimal place.

B.2 Baseline Details

We compare LSA with three categories of baselines.

General-purpose Visual Representations. We include CLIP [44] and DINOv3 [45] as strong general visual representation baselines. CLIP is a vision-language model trained with contrastive image-text supervision, which learns transferable semantic visual features from large-scale image-caption pairs. DINOv3 is a self-supervised vision foundation model that learns strong image-level and dense visual features without manual labels. Although both models provide powerful general-purpose representations, their pretraining objectives are not explicitly designed for action-relevant 3D reasoning or robot manipulation.

Robot Visual Pretraining Methods. We compare with MCR [46], VIP [47], R3M [19], VC-1 [48], HRP [8], MPI [9], and R3M-A [37]. These methods represent strong robot-oriented visual pretraining baselines, covering different ways of learning action-relevant representations for downstream robotic control.

Direct 3D Policies. We further compare with DP3 [49] and KAN-We-Flow [50], both of which directly take 3D point clouds as policy inputs and learn action prediction through diffusion- or flow-matching-based objectives. These methods provide direct comparisons to LSA from the perspective of 3D-aware policy learning. Following Lift3D [61], we compare the 2D and 3D policies under the same evaluation protocol, using 224×224 RGB images and single-view point clouds with 1,024 points as inputs, respectively, while retaining the original method-specific settings of the 3D policies.

For all 2D baselines, we use the same downstream demonstrations. The pretrained visual encoder is frozen and combined with the same lightweight behavior cloning policy head. The policy head is implemented as a 3-layer MLP with hidden size 256, following prior works [46, 8]. We apply the reporting protocol used for LSA to all 2D baselines.

B.3 3D Wrist Lifting Evaluation Set Construction

To quantify the noise of the lifted 3D wrist trajectories, we construct two small evaluation sets from H2O [62] and DexYCB [56], respectively. For each dataset, we randomly sample 240 valid wrist samples from 12 scenes, covering either left or right hands. Both datasets provide calibrated cameras

and ground-truth 3D hand poses. For each sample, we use the ground-truth 2D wrist projection as the wrist location, estimate depth and camera intrinsics from different sources, and lift the wrist point to the camera coordinate system. Following [63], we report the 3D Average Displacement Error (ADE) between the lifted wrist and the ground-truth 3D wrist.

B.4 Real-World Implementation Details

Hardware Setup. We conduct real-world experiments using a 7-DoF Franka Research 3 robot equipped with a 1-DoF parallel gripper. A RealSense D435i camera is used to capture RGB observations from a fixed third-person viewpoint. During data collection, we record robot states and RGB observations at 16 Hz.

Demonstration Collection. For each task, we collect 30 teleoperated demonstrations. The demonstrations include variations in the initial poses of the robot and objects to avoid learning a fixed execution trajectory. The four real-world tasks are defined as follows:

- *Wipe PC*: the robot grasps a pink towel and wipes the surface of a laptop.
- *Pick Trash*: the robot picks up a piece of trash and drops it into a trash bin.
- *Place Toy*: the robot grasps a toy and places it into a basket.
- *Pluck Flower*: the robot grasps a flower and lifts it upward.

Policy Training. Following prior real-world policy evaluation protocols, we freeze the pretrained visual encoder and train only the downstream policy head. The policy head is a 3-layer MLP with two hidden layers of size 512. We train the policy using a log-likelihood objective with a learning rate of 1×10^{-4} and a batch size of 128 for 50k steps. The predicted actions are linearly interpolated to 1000 Hz to satisfy the Franka control frequency.

Evaluation Protocol. For each task, we evaluate each policy over 10 independent trials and report the success rate. A trial is considered successful if the robot completes the target manipulation goal without human intervention. For the OOD variants of *Wipe PC*, the policy is trained with a pink towel and evaluated with two unseen towel colors, purple and yellow. This setting changes the object appearance while preserving the same task goal and interaction structure.

C Additional Experimental Results

This section provides additional experimental analyses that are not included in the main paper due to space limitations. Unless otherwise specified, ablation studies are conducted on four MetaWorld tasks, including *dial turn*, *hand insert*, *pick place wall*, and *disassemble*. We report the average evaluation success rate across these tasks.

C.1 Comparison with Direct 3D-FM Features

We compare Full LSA with variants that directly use 3D-FM features. We construct direct Depth Pro, VGGT, and dual-FM policies by uniformly projecting their frozen features to 768 dimensions and feeding them into the same action head. We also combine these direct features with LSA without Lift. All variants use the same downstream training and evaluation settings as Full LSA. The results are shown in Table 7.

Table 7: Comparison between Full LSA and the direct use of 3D-FM features.

Method	SR (%) ↑	Params (M) ↓	GFLOPs ↓	FPS ↑	Slowdown ↓
Full LSA	**92.2**	**86.1**	**35.1**	**230.6**	**1.0×**
Direct Depth Pro	40.7	953.0	382.3	59.6	3.9×
Direct VGGT	29.8	911.0	493.6	18.7	12.3×
Direct Dual-FM	26.9	1,863.7	875.9	14.5	15.9×
LSA w/o Lift + Depth Pro	81.4	1,039.4	417.5	47.0	4.9×
LSA w/o Lift + VGGT	61.6	997.3	528.8	17.5	13.2×
LSA w/o Lift + Dual-FM	66.4	1,950.1	911.1	13.6	17.0×

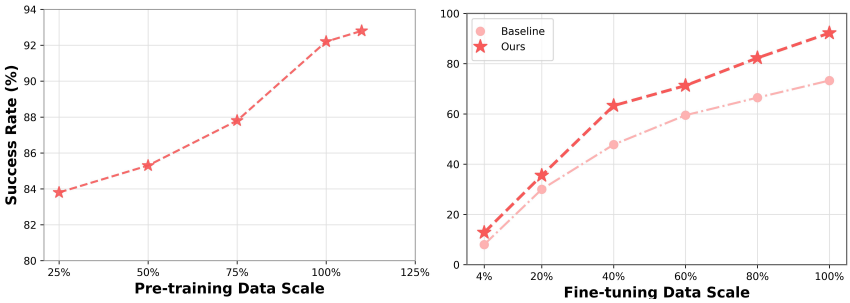


Figure 8: Influence of pretraining and fine-tuning data scale.

Full LSA achieves 92.2%, whereas the three direct policies achieve only 40.7%, 29.8%, and 26.9%. Adding direct features to LSA without Lift improves the results but remains below Full LSA, with the strongest variant reaching 81.4%. Full LSA also deploys only 86.1M parameters and runs at 230.6 FPS, while the direct variants require 911.0M–1,950.1M parameters and are 3.9×–17.0× slower. These results show that directly deploying generic 3D-FM features provides neither effective action representations nor practical inference efficiency. Lift instead transfers their dense 3D knowledge into a compact visual encoder during pretraining, yielding substantially stronger performance without running either 3D FM. All inference costs are measured on a single RTX 4090 with batch size 1 and one 224×224 RGB frame. Slowdown is measured relative to Full LSA.

C.2 Selection of 3D FMs

We use multiple 3D FMs to obtain complementary 3D supervision. VGGT provides camera parameters and geometric representations, while Depth Pro provides absolute-depth information. Their combination provides richer and complementary representations than either teacher alone. This design is not restricted to specific 3D FMs. We further replace VGGT and Depth Pro individually or jointly with Depth Anything 3 (DA3) [64] and UniDepthV2 [65]. As shown in Table 8, replacing the teachers individually or jointly achieves comparable performance, demonstrating that LSA is compatible with alternative 3D FMs.

Table 8: Teacher combinations in the Lift phase.

Teacher Combination	SR (%)
VGGT + Depth Pro	92.2
DA3 + Depth Pro	92.0
VGGT + UniDepthV2	**92.5**
DA3 + UniDepthV2	91.8

C.3 Data Scaling and Sample Efficiency

We first investigate the effect of data scale during both human-video pretraining and downstream policy learning. As shown in Fig. 8, increasing the amount of pretraining video data consistently improves downstream success rates, indicating that LSA can benefit from larger-scale human-video data. We also evaluate the sample efficiency of downstream policy learning by varying the number of expert demonstrations. LSA nearly matches the peak performance of baseline methods using only 60% of the downstream expert demonstrations. This suggests that the multi-tiered 3D knowledge learned during pretraining reduces the burden of learning spatial and interaction cues from limited robot demonstrations.

C.4 Parameter Update Strategies

We compare different parameter update strategies during pretraining. As shown in Tab. 9, updating all parameters leads to a large performance drop. This may be caused by overfitting to the relatively limited pretraining data and forgetting the general visual priors of the initialized ViT. Fine-tuning only LayerNorm parameters performs much better, but still underperforms LoRA. Therefore, we adopt LoRA as the default parameter-efficient adaptation strategy, as it adapts the encoder to our 3D supervision while preserving the general visual representation ability of the pretrained backbone.

Table 9: Comparison of different parameter update strategies.

Method	SR (%)
Full Parameters	67.2 (↓ 25.0)
LN Fine-tuning	90.1 (↓ 2.1)
LoRA Fine-tuning	**92.2**

Table 10: Effect of pretraining data diversity. All settings use 20K sampled clips.

Data Source	#Videos	SR (%)
H2O	∼100	90.2
DexYCB	∼400	92.2
Ego4D	∼1000	**93.1**

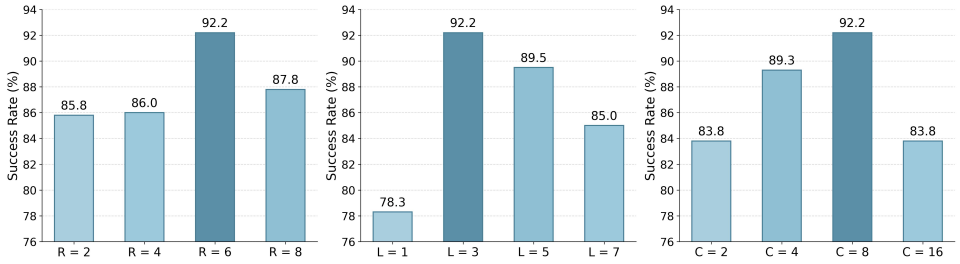


Figure 9: Impact of attentive guidance depth R, motion planning steps L, and the number of depth reconstruction decoder layers C.

C.5 Pretraining Data Diversity

We further evaluate whether LSA benefits from more diverse human-video sources. As shown in Tab. 10, all settings use the same number of sampled clips, while the number of source videos varies. Pretraining on H2O [62] with around 100 source videos achieves 90.2%, while DexYCB with around 400 source videos improves the success rate to 92.2%. Using Ego4D [66] with around 1000 source videos further improves the result to 93.1%. These results suggest that LSA benefits from broader source-video diversity, indicating its potential to scale with more diverse human-video pretraining data.

C.6 Hyperparameter Selection

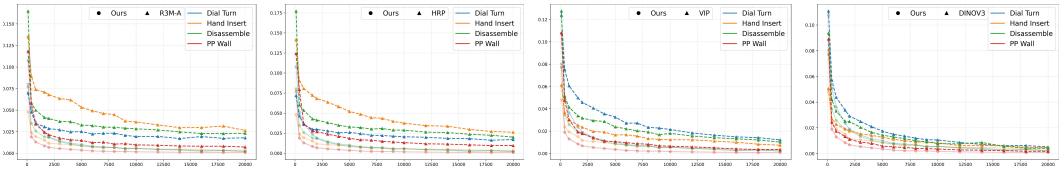
As shown in Fig. 9, we investigate the optimal configuration of three critical hyperparameters: the depth of attentive guidance R, the number of motion planning steps L, and the number of depth reconstruction decoder layers C. For attentive guidance, applying it to the first $R = 6$ layers achieves the highest success rate of 92.2%. Restricting guidance to only the first two layers ($R = 2$) lowers the success rate to 85.8%, because this limited supervision makes it difficult to identify a reliable target attention map and leaves other optimizable shallow layers unguided. In contrast, extending it too deeply ($R = 8$) reduces performance to 87.8%, as it limits the ability of deeper layers to learn abstract task semantics. For motion planning, $L = 3$ yields the best performance. A shorter horizon ($L = 1$) severely degrades the success rate to 78.3%, because such a short horizon lacks sufficient temporal context to represent meaningful hand motion. In contrast, a longer horizon ($L = 7$) reduces performance to 85.0%, since predicting farther-future trajectories becomes overly difficult and inaccurate. Finally, an 8-layer depth decoder ($C = 8$) best balances model capacity and training stability. A shallower decoder ($C = 2$) leads to a lower success rate of 83.8%, because its limited capacity is insufficient to reconstruct dense and accurate depth maps. In contrast, an overly deep decoder ($C = 16$) also reduces performance to 83.8%, as it introduces optimization difficulties that hinder overall training. Therefore, we adopt $R = 6$, $L = 3$, and $C = 8$ as our default setup.

C.7 Layer Selection

Tab. 11 presents an ablation on the layer configurations for representation alignment and depth reconstruction. While keeping the depth reconstruction fixed at the 6th layer, setting the alignment layer too shallow—such as the (1, 6) configuration—leads to a 2.7% drop in success rate, as the network has not yet extracted sufficient semantic features to effectively align with the high-dimensional implicit representations of the 3D FMs. Conversely, setting the alignment layer too

Table 11: Ablation study on the selection of alignment and reconstruction layers.

Layer (Align, Recon)	Ours	Varying Alignment		Varying Reconstruction		Reversed Order	Shared Layer	
	(3, 6)	(1, 6)	(5, 6)	(3, 4)	(3, 8)	(6, 3)	(3, 3)	(6, 6)
SR (%)	**92.2**	89.5 (↓ 2.7)	88.6 (↓ 3.6)	88.3 (↓ 3.9)	84.6 (↓ 7.6)	89.3 (↓ 2.9)	85.8 (↓ 6.4)	82.3 (↓ 9.9)



(a) vs. R3M-A (b) vs. HRP (c) vs. VIP (d) vs. DINOv3

Figure 10: Comparison of training convergence between LSA and several baselines: (a) R3M-A, (b) HRP, (c) VIP, and (d) DINOv3.

deep, as seen in the (5, 6) setup, causes a 3.6% performance decrease by excessively compressing the subsequent feature space required for explicit structural perception.

Similarly, fixing the feature alignment at the 3rd layer and altering the depth reconstruction layer also significantly impacts performance. Introducing reconstruction too early, as in the (3, 4) configuration, results in a 3.9% decline in success rate. At this stage, the network has not fully assimilated the implicit 3D representations from the Lift phase, preventing it from effectively comprehending complex spatial layouts. Furthermore, delaying the reconstruction task to an excessively deep layer like (3, 8) triggers a severe performance drop of 7.6%. This is because representations become highly abstract as they approach the Act phase, focusing primarily on global motion trajectory planning. Forcing these deep layers to retain fine-grained, pixel-level geometric information inherently interferes with the action planning task.

In addition, reversing the order to (6, 3) results in a 2.9% drop. The model performs better when it first acquires a high-level 3D understanding (implicit) before attempting detailed geometric reconstruction (explicit). This suggests that the aligned 3D representations provide a necessary foundation for learning accurate 3D spatial layouts. Finally, forcing the model to perform both feature alignment and depth reconstruction at the exact same layer—such as in the (3, 3) or (6, 6) configurations—results in substantial performance drops of 6.4% and 9.9%, respectively. Since feature alignment and depth reconstruction are tasks of completely different granularities, forcing them to share the same feature space triggers severe feature interference.

C.8 Training Loss Curves

Fig. 10 presents the fine-tuning loss curves of different pretraining methods on MetaWorld. Our method exhibits the fastest and most stable convergence and reaches the lowest loss among all compared approaches. We attribute this advantage to the multi-tiered 3D supervision introduced during pretraining, which equips the policy with more generalizable spatial and motion-planning knowledge and thus enables faster adaptation to downstream manipulation tasks.

C.9 Additional Baseline Comparisons

We further compare LSA with the representative 3D-FM-based approach VITRA [12] and recent robotic pretraining methods, including H-RDT [32] and Moto [67]. To fairly evaluate their pretrained representations, we freeze the pretrained components of each method and use the same action head, downstream demonstrations, and evaluation protocol. As shown in Table 12, LSA achieves the highest average performance on MetaWorld among these methods.

C.10 Additional Real-World Generalization Experiments

We extend the real-world OOD evaluation to Wipe PC and Place Toy. Background Change uses an unseen tablecloth, while Low-Light uses reduced ambient illumination unseen during training.

Table 12: Additional baseline comparisons on MetaWorld.

Method	HA	DT	AS	HI	PPW	ShP	DI	StP	Avg.
VITRA [12]	87.1±2.0	82.2±6.7	98.7±1.4	77.8±2.0	81.8±6.8	77.8±2.0	77.3±2.3	78.2±2.8	82.6
Moto [67]	21.8±3.3	64.0±2.3	87.6±3.4	58.2±0.8	57.8±4.7	64.0±2.3	64.4±4.3	80.5±2.0	62.3
H-RDT [32]	98.2±1.6	86.2±2.0	100.0±0.0	83.5±2.0	86.2±2.0	76.0±2.3	78.2±1.6	92.4±4.1	87.6
LSA (Ours)	**100.0±0.0**	**93.3±2.3**	**100.0±0.0**	**85.3±2.3**	**98.0±2.0**	**84.0±0.0**	**92.3±1.5**	**82.7±1.5**	**92.0**

Table 13: Real-world generalization under unseen backgrounds and low-light conditions. Values in parentheses denote absolute performance drops.

	Place Toy			Wipe PC			Average Drop	
Method	**Original**	**Background**	**Low-Light**	**Original**	**Background**	**Low-Light**	**Background**	**Low-Light**
HRP	65.0	35.0 (−30.0)	30.0 (−35.0)	60.0	35.0 (−25.0)	25.0 (−35.0)	−27.5	−35.0
LSA	**80.0**	**65.0** (−15.0)	**60.0** (−20.0)	**75.0**	**60.0** (−15.0)	**50.0** (−25.0)	−15.0	−22.5

Each policy is trained using the same 30 demonstrations, and each condition is evaluated over 20 independent trials. We compare LSA with HRP under identical settings. The results are reported in Table 13. LSA consistently achieves higher OOD success rates and smaller performance drops than HRP, extending the evaluation beyond object-appearance changes to unseen tabletop backgrounds and illumination conditions.

C.11 Computational Overhead

All computational costs are measured using NVIDIA RTX 4090 GPUs. The one-time offline processing of approximately 20K training sequences with VGGT and Depth Pro requires 0.56 GPU-hours and 36.37 GiB aggregate peak GPU memory, producing a 28.34 GiB feature cache. The 3D FMs are not used during downstream inference. LSA requires 4.34 ms per inference step with 0.341 GiB peak GPU memory.

D Further Discussion on Limitations

Embodiment Differences. LSA adopts future 3D wrist trajectories as action proxies primarily to bypass the embodiment gap between humans and robots. In our preliminary experiments, additionally supervising with fine-grained 3D hand meshes, such as MANO [29], degraded performance compared with using wrist trajectories alone. We attribute this to the structural mismatch between human fingers and robot end-effectors, such as parallel grippers. Embodiment-specific action representations impose an additional burden on the downstream policy rather than providing transferable guidance. A promising solution is to extract latent actions from human hand motion, so that the supervision captures high-level motion intent rather than embodiment-specific action forms.

Object Interaction Discrepancies. Although LSA's attentive guidance effectively transfers region-level attention from human hand–object interactions to task-relevant gripper–object regions, it primarily provides coarse spatial guidance. Extending this supervision to finer-grained interaction patterns is more challenging because differences in embodiment appearance and manipulation modes may limit the direct transfer of such detailed cues to robots. One potential solution is to incorporate explicit object-affordance supervision, such as object regions and candidate contact points. Such object-centric and embodiment-agnostic signals could complement the current attentive guidance and further strengthen the transfer of interaction knowledge to downstream robot tasks.

Real-World Distribution Shifts. While our results suggest that 3D representations learned from human videos benefit downstream tasks, scene discrepancies between human-video data and real-world robot environments may still limit their transferability. A possible solution is to jointly pretrain on unlabeled human and real-robot videos using shared depth-reconstruction and 3D-feature-alignment objectives, encouraging the visual encoder to learn a unified 3D scene-geometric representation across both domains.

E Additional Visualizations

This section provides additional visualizations. We include representative simulation and real-world trajectories, as well as more attentive guidance visualizations on both human-object interaction and robotic manipulation scenes.

E.1 Simulation Trajectory Visualizations

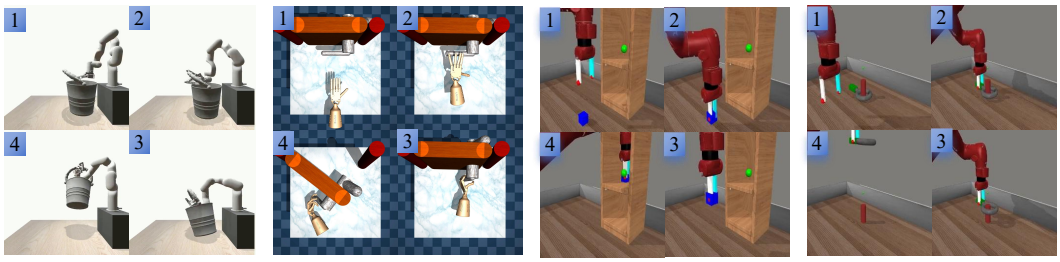
Fig. 11 shows representative task trajectories in simulation. These examples qualitatively demonstrate that the policy pretrained by LSA can execute diverse manipulation behaviors across different simulated environments. The visualized trajectories cover both tabletop manipulation and dexterous manipulation settings, complementing the quantitative results reported in the main paper.

E.2 Real-World Trajectory Visualizations

Fig. 12 presents representative real-world trajectories executed by LSA. The examples correspond to the real-world tasks described in the main paper, including *Wipe PC*, *Pick Trash*, *Place Toy*, and *Pluck Flower*. These visualizations provide qualitative evidence that the learned representation can transfer from human-video pretraining to real-world robot manipulation.

E.3 More Attentive Guidance Visualizations

Fig. 13 presents additional visualization results of attentive guidance on human-object interaction scenes. The model learns to focus on hand-centric interaction regions, often covering both the hand and the interacting object. This indicates that the hand prior is not used to impose a fixed hand heatmap, but to mine an internally emerged interaction-aware attention target. Moreover, Fig. 14 shows that such attention patterns can transfer to robotic manipulation scenes, where the model places stronger emphasis on task-relevant regions, such as the gripper and manipulated objects, while reducing attention to less relevant background regions.



Dexart-Bucket *Adroit-Door* *Metaworld-Shelf Place* *Metaworld-Disassemble*

Figure 11: Visualization of representative simulation task trajectories. These examples illustrate the execution behavior of LSA in simulated manipulation environments.

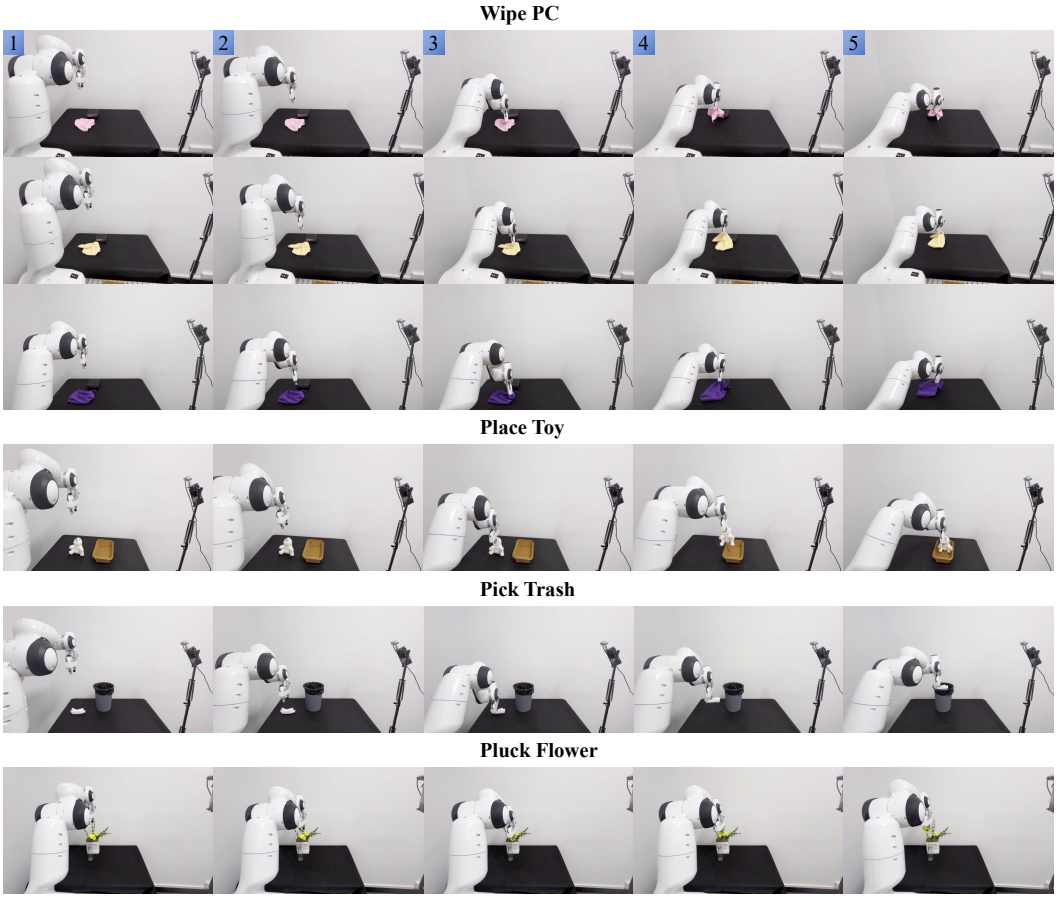


Figure 12: Visualization of representative real-world task trajectories. The examples show LSA executing real-world manipulation tasks with the Franka Research 3 robot.

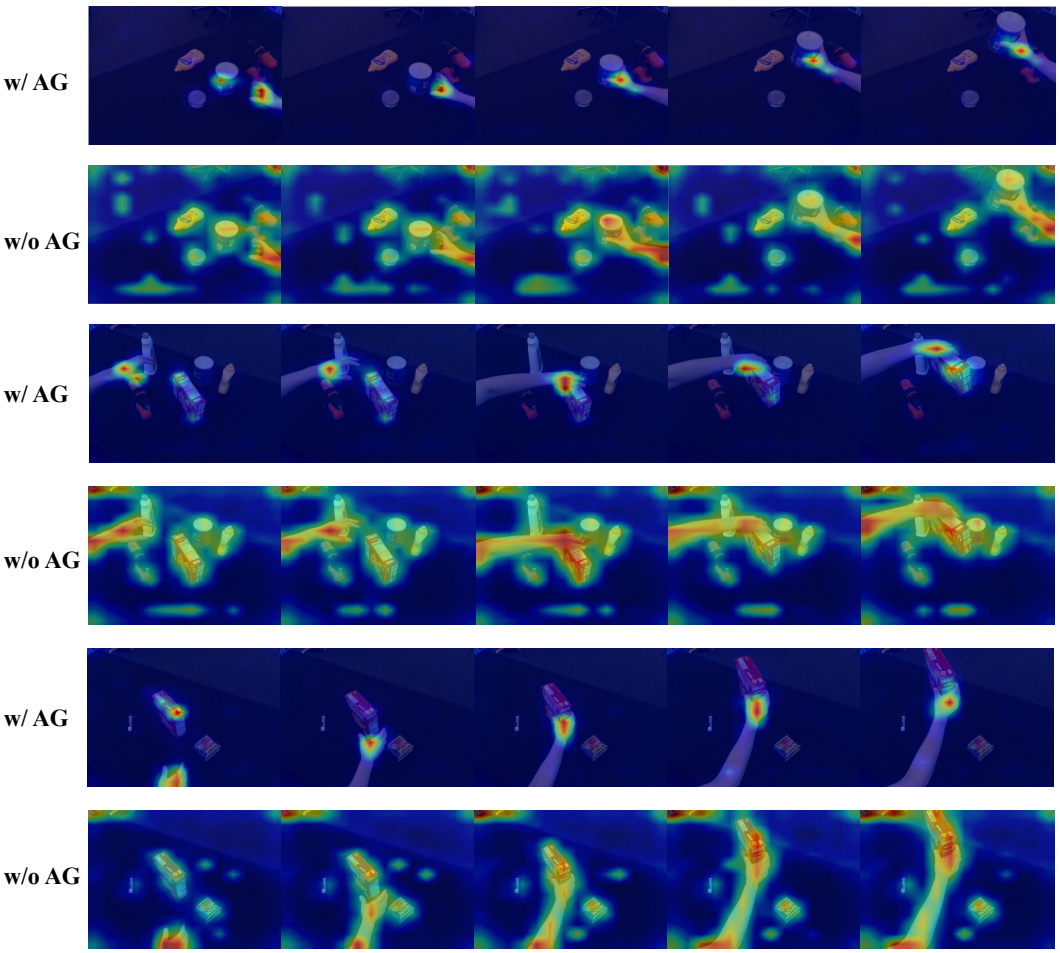


Figure 13: Additional visualization results of attentive guidance. The model learns to focus on hand-centric interaction regions, often covering both the hand and the interacting object.

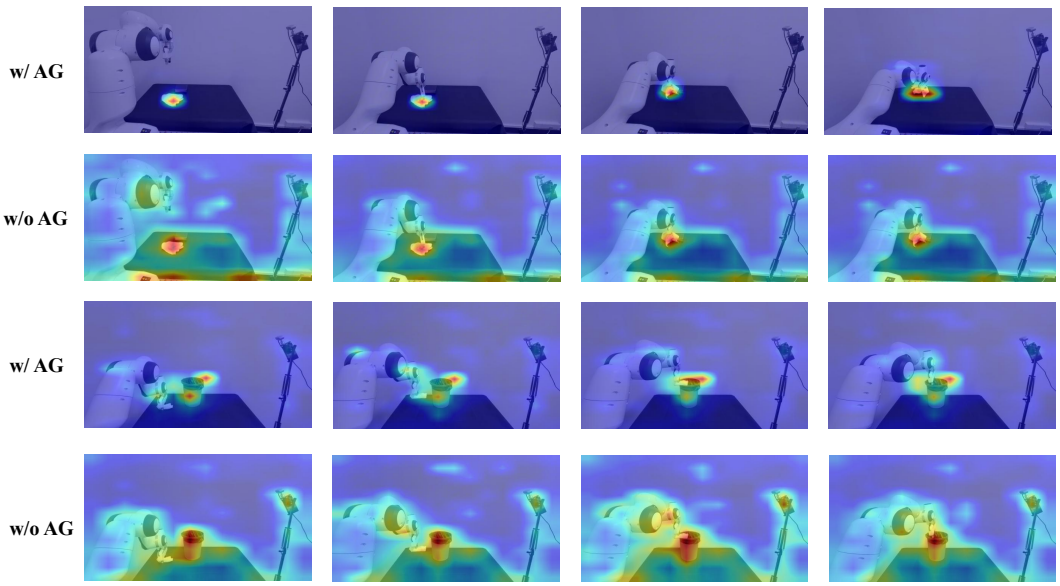


Figure 14: Additional visualization results of attentive guidance in robotic manipulation scenes. The model consistently places stronger attention on task-relevant regions, such as the gripper and manipulated objects, while reducing attention to background regions.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes], [No], or [N/A].
- [N/A] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for [N/A]).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will also be asked to include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While [Yes] is generally preferable to [No], it is perfectly acceptable to answer [No] provided a proper justification is given (e.g., error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering [No] or [N/A] is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately describe the proposed Lift-See-Act framework, including 3D representation alignment, structural guidance, attentive guidance, and motion planning. These claims are supported by the method design in Section 3 and the empirical evaluations reported in Section 4.

   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: We discuss the limitations and future directions of the proposed framework in the conclusion.

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: This paper does not present formal theoretical results, theorems, or proofs. The contribution is mainly methodological and empirical, focusing on robot policy pretraining with hierarchical 3D supervision.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

Justification: The paper provides the main algorithmic components, including the data engine, dense representation alignment, depth-map reconstruction, attentive guidance, motion planning objective, and overall loss formulation. The experimental protocol, datasets, metrics, and implementation details are described in Section 4 and the appendix.

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: [No]

Justification: The code and processed data are not publicly released at the time of submission, although the paper states that the code will be released. We provide detailed methodological and experimental descriptions to support reproducibility before the code release.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The paper specifies the major training and evaluation settings, including datasets, metrics, model components, loss terms, and key hyperparameters. Additional dataset and implementation details are provided in the appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The current version reports the main experimental results and provides error bars.

   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: The current version provides complete compute-resource information, such as GPU type and memory.

Guidelines:

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

9. **Code of ethics**

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

Answer: [Yes]

Justification: The research is conducted in accordance with the NeurIPS Code of Ethics. The work uses existing datasets and models for robot policy pretraining and does not involve deceptive data collection, unsafe human-subject experiments, or intentional misuse.

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: The paper discusses the societal impacts. To the best of our knowledge, this work has no obvious negative social impact.

Guidelines:

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

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release high-risk generative models, scraped datasets, or deployed robotic systems that directly enable misuse. The proposed framework is a robot policy pretraining method, and any real-world deployment should still be conducted under controlled safety constraints.

    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: The paper cites the existing datasets, foundation models, and prior codebases used in the study. We follow the corresponding licenses and terms of use for all external assets, and will include explicit license information in the released code and documentation.

    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 introduce a new benchmark dataset or newly collected human-subject dataset. The main contribution is a pretraining framework and associated training objectives built on existing data and models.

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 conduct new research with human subjects or collect new data from participants.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not conduct new research with human subjects or collect new data from participants. Therefore, IRB approval or equivalent human-subject review is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

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

Guidelines:

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