# Interaction-Aligned Robot Learning from Human Videos with Structured Graph Modeling

**Jiawei Fu and Lin Yang**
CUHK
fujiawei0724@gmail.com

## Abstract

Recent advances in robotics foundation models have shown promising multi-task generalization, yet their reliance on specialized embodiment-specific data and model fundamentally prevents them from exploiting the far richer physical interaction knowledge present in abundant human manipulation videos. Existing approaches that leverage human videos suffer from misaligned representations that preclude explicit unified interaction modeling over human video and robot demonstration, failing to capture unified physical interaction dynamics essential for cross-domain transfer. We address this by constructing a unified representation that aligns human videos and robot tasks through scene point clouds with hand-to-gripper mapping via spatial tracking and hand pose estimation. Leveraging this representation, we propose a structured graph transformer that explicitly models spatial, semantic, and intentional interaction entities through graph attention mechanisms. The modeled multi-level interaction features then enable interaction-aligned cross-domain learning, transferring manipulation knowledge from human videos to robot tasks via discrete interaction abstraction and hierarchical distribution alignment. Experiments on RLBench, ManiSkill2, and real-world robot demonstrate state-of-the-art performance on diverse benchmarks with fewer demonstrations and effectiveness of our designed modules. Remarkably, our method achieves zero-shot transfer on robot tasks when trained exclusively on human video, providing strong evidence for effective human-to-robot transfer through interaction alignment.

## 1 Introduction

Robotics foundation models [1, 2, 3, 4, 5, 6, 7] have recently demonstrated remarkable multi-task generalization capabilities in robot manipulation by training on large-scale robot data collected across diverse platforms [8]. The paired robot perception with proprioception data across robotic platforms enables robotics foundation models with specialized structures [2, 4, 9] to learn within-embodiment physical interaction patterns and behaviors across diverse scenes, boosting generalization performance through robot-data pre-training and following robot-demonstration fine-tuning.

Nevertheless, acquiring sufficient paired perception and proprioception data for robust physical interaction reasoning remains fundamentally constrained by robot data collection scalability. In contrast, human manipulation videos [10, 11, 12, 13] offer a broader and orders-of-magnitude richer source of interaction entities, encoding diverse object geometries, task contexts, and intentional behaviors, forming heterogeneous interaction structures far beyond robot datasets. This asymmetry naturally motivates pre-training on human video, as underlying hand-object manipulation dynamics are grounded in the same physical interactions shared across embodiments. Yet realizing this potential demands a methodology that faithfully transfers interaction knowledge across embodiments, a requirement that hinges on two ingredients: representing human and robot domains in a unified space, and reasoning explicitly over interaction entities to comprehensively align them across embodiments.

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

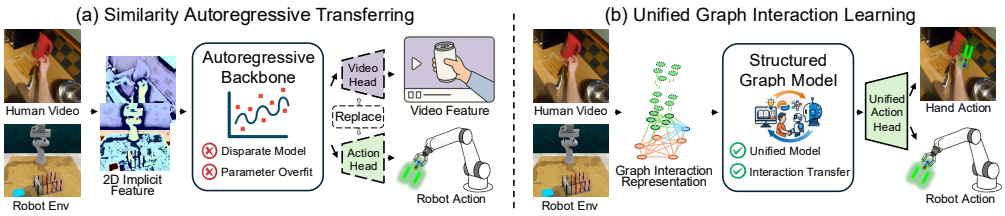


Figure 1: (a) Baselines use implicit features with disparate heads, causing representation mismatch. (b) Our structured graph enables explicit interaction modeling and unified human-robot prediction.

Recent approaches [5, 6, 2, 3] pursue this goal through autoregressive pre-training on abundant human data, recognizing implicit sequence similarities with text semantics between human video and robot manipulation tasks as a bridge across embodiments. However, this implicit alignment falls short of the unified representation and explicit interaction reasoning the problem demands, exhibiting a representation mismatch between their pre-training objective and the manipulation transfer target: autoregressive backbones with disparate replaceable heads optimize next-frame continuity, whereas manipulation competence is governed by discrete physical action and interaction patterns (reach, grasp, transport, place) poorly captured by frame-level signals. Pre-training on trajectory similarity therefore tends to overfit parameters to superficial appearance correlations rather than reasons the underlying unified physical interaction and dynamics transferable across embodiments (Figure 1).

To overcome these limitations, we introduce a unified representation that encodes human videos and robot scenes with corresponding hand and gripper actions, capturing spatio-temporal physical interaction across domains. As physical interaction is inherently relational and heterogeneous, entangling spatial, semantic, and intentional entities that resist joint encoding, each entity warrants explicit and disentangled reasoning, motivating a structured graph modeling over the unified representation (Figure 1). Moreover, bridging human and robot domains demands alignment of interaction knowledge at both fine-grained metric and coarse abstraction levels, which we cast as an interaction-aligned cross-domain optimization framework to match continuous and discrete interaction distributions.

Specifically, we employ spatial tracker [14] with monocular depth estimation [15] to construct 4D spatio-temporal point clouds [2], while concurrently leveraging in-the-wild hand joint estimation [16] to obtain hand pose and scene point cloud alignment, yielding spatio-temporally consistent representations with physical interaction features. Upon this representation, we instantiate structured interaction modeling as a graph transformer with three dedicated edge groups—spatial, semantic, and intentional—each updated by edge-conditioned graph attention in a three-stage schedule that progressively composes geometric, task-conditioned, and intent-aware action features. For multi-granular interaction transfer over disentangled features, we adopt a two-stage optimization: (i) pre-training on human videos where an Interaction Abstraction Module (IAM) distills interaction features into discrete primitives via vector quantization codebooks, while accumulating continuous reference statistics, (ii) fine-tuning on robot data under a hierarchical alignment loss combining kernel MMD [17] on continuous embeddings with KL divergence, transferring interaction knowledge across domains.

We evaluate our approach on RLBench [18] and ManiSkill2 [19], our method achieves state-of-the-art performance on both, outperforming existing human-to-robot and point cloud baselines by 32.9% and 19.3%, respectively. Comprehensive ablation studies validate the effectiveness of graph modeling of human video pre-training paradigm and interaction alignment effectiveness. Notably, we conduct a zero-shot transfer experiment on RLBench, training exclusively on human video data with precise hand-scene alignment from DexYCB [20], without robot demonstration fine-tuning. Our framework successfully accomplishes multiple tasks in this challenging setting, demonstrating cross-domain transfer capability of our unified representation with structured graph modeling. For real-world robot experiments, our approach demonstrates comparable performance across multiple manipulation tasks with fewer demonstrations than alternative methods. We summarize our contributions as follows:

- We identify unified representation and explicit interaction reasoning as the missing ingredients for human-to-robot transfer, and introduce a unified representation via spatial tracking with hand estimation, upon which we build a structured graph transformer explicitly modeling spatial, semantic, and intentional interaction entities across human and robot domains.

- We propose a cross-domain framework aligning human and robot interactions jointly at continuous and discrete levels, where an Interaction Abstraction Module quantizes graph features into embodiment-invariant primitives and a hierarchical loss combines kernel MMD with KL divergence.
- Our method achieves state-of-the-art performance on RLBench and ManiSkill2 multi-task benchmark, with successful real-world deployment. Remarkably, our model trained exclusively on human video zero-shot transfer to robot tasks, validating effective cross-embodiment generalization.

## 2 Related Work

**Human Data Training for Robot Learning.** Robotics models trained on large-scale datasets [8], including RT-1 [1], RT-2 [21], Octo [22], OpenVLA [4], and LLARVA [3], demonstrate strong cross-embodiment transfer capabilities across diverse manipulation tasks. To leverage unlabeled human data, existing approaches broadly fall into two families. The first family [2, 23, 24, 25, 26, 27] pre-trains on human videos by learning implicit trajectory similarities between human and robot domains, for instance, aligning 2D hand keypoints with robot actions through separate prediction heads (AR-VRM [23]) or autoregressive modeling over 4D points with proprioception (ARM4R [2]). However, these methods suffer from representational misalignment between video-based feature prediction and robot action objectives, failing to capture unified physical interaction across domains. The second family [28, 29, 30, 31, 32] instead clones human actions directly to robot execution by replicating the same manipulation scenario, for instance, via hand-as-gripper overlay (Phantom [28]) or egocentric demonstration mimicking (EgoMimic [30]), requiring extensive scene-specific preprocessing that restricts applicability to limited and controlled settings. In contrast, our method introduces a unified representation with explicit interaction modeling over shared physical dynamics, simultaneously overcoming the representational misalignment of the first family through structured graph reasoning, and the scene-specific constraints of the second family through scalable pre-training on human video.

**Domain Adaptation in Robotics.** Domain adaptation techniques aim to transfer knowledge from a source to a target domain by minimizing distribution discrepancies [33, 34, 35], classically via adversarial training [33], maximum mean discrepancy [34], or correlation alignment [36]. In robotics, these techniques have been primarily applied to bridge the sim-to-real gap [37, 38, 39, 40] through randomization and policy transfer. A parallel line of work targets cross-embodiment and visual adaptation [41, 42, 43, 44], aligning perception models across camera viewpoints [45] or lighting conditions [46]. However, these methods focus on perception-level alignment while assuming shared action spaces or task structures. In contrast, our work addresses interaction-level adaptation across fundamentally different embodiments and observation modalities, bridging human and robot domains through hierarchical distribution matching over shared physical interaction primitives.

**Graph Neural Networks for Robotics.** Building on foundational graph learning architectures [47, 48, 49], graph networks have been employed in robotics for policy learning [50, 51], reward design [52, 53, 54], and object-centric manipulation [55, 56, 57]. Scene-graph approaches further leverage object-level relations for long-horizon planning and multi-object manipulation [58, 59]. Instant Policy [60] applies graph networks with in-context learning for demo-guided multi-task manipulation, while GraphMimic [61] constructs image-coordinate graphs from task-specific human demonstrations for robot execution. In contrast, our work leverages a structured graph over a unified human–robot representation to explicitly reason about spatial, semantic, and intentional interaction entities, enabling cross-embodiment transfer of shared multi-granular physical interaction primitives.

## 3 Method

We present a cross-domain framework that leverages human videos to enhance robot learning, premised on embodiment-shared interactions entangling heterogeneous relational entities that warrant explicit disentangled modeling and alignment. We operationalize this via a unified representation encoding domains through spatial tracking and hand-to-gripper transformation (Section 3.1), a structured graph transformer disentangling spatial, semantic, and intentional interaction modeling (Section 3.2), and an interaction-aligned scheme matching distributions at continuous and discrete levels (Section 3.3), with human video pre-training followed by robot demonstration fine-tuning.

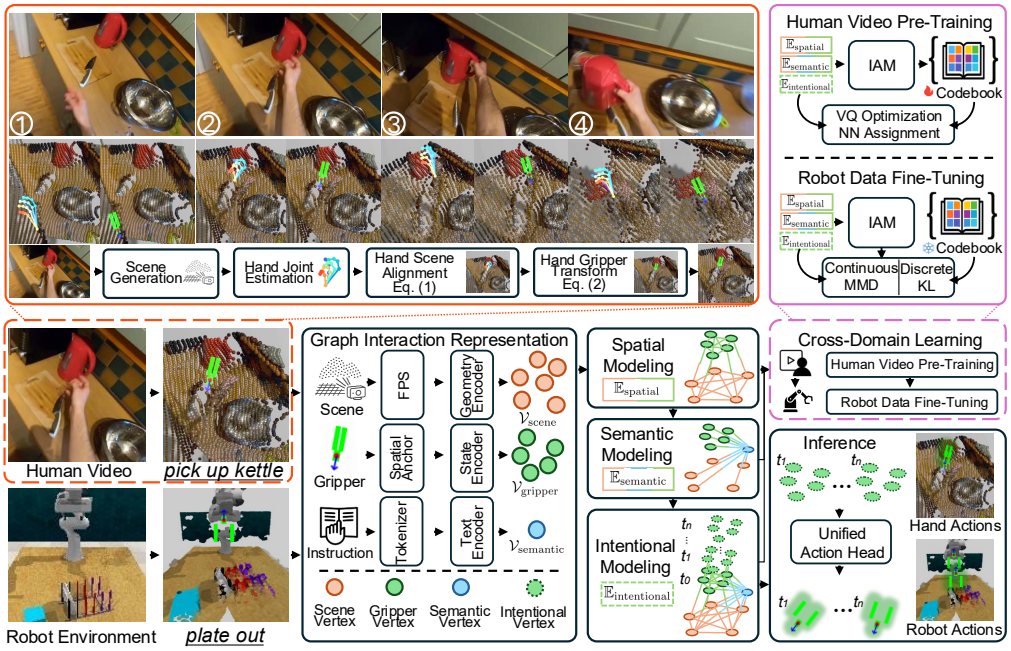


Figure 2: Our framework constructs a unified human-robot representation via hand-scene alignment (Eq. (1)) and hand-gripper transformation (Eq. (2)), upon which a structured graph transformer reasons over spatial ($\mathbb{E}_{\text{spatial}}$), semantic ($\mathbb{E}_{\text{semantic}}$), and intentional ($\mathbb{E}_{\text{intentional}}$) interaction entities to enable cross-domain learning from human videos to robot tasks using multi-granular alignment.

### 3.1 Unified Representation for Aligned Interaction Construction

To bridge human video and robot learning domains with physical interaction reasoning, we construct a shared representation capturing geometric structure, semantic instruction pattern, and intentional behavior through spatial point cloud tracking and hand-to-gripper transformation.

**Scene Point Cloud Construction.** We extract 4D tracked scene point clouds $\{\mathcal{P}_t\}_{t=0}^{T}$ (each point $\rho \in \mathcal{P}_t, \rho \in \mathbb{R}^3$) from human manipulation videos using spatial tracking [14] with monocular depth estimation [15]. We process 76K video clips from Epic-Kitchens [12], tracking salient points across frames to construct spatio-temporal representations in camera coordinates. Tracked points inherently encode both object-relative and camera-relative motion while mitigating egocentric video artifacts.

**Hand Joints Estimation and Transformation.** To establish correspondence between domains, we extract hand poses using InterWild [16] following the MANO definition [62], yielding 21 joint positions $\mathcal{J} = \{\mathbf{j}_i = (x_i, y_i, z_i)\}_{i=1}^{21}$ in 2.5D space. We align hand joints with scene point clouds via depth optimization to correct metric-scale ambiguity from monocular estimation:

$$\mathcal{J}^* = \arg\min_{\mathcal{J}'} \sum_{i=1}^{21} \min_{\rho \in \mathcal{P}} \left\| \mathbf{j}_i' - \rho \right\|_2, \quad \mathbf{j}_i' = (x_i,\, y_i,\, z_i'), \tag{1}$$

where only depth $z_i'$ is optimized with 2D projections fixed, initialized from corresponding point cloud depth estimates. The hand-as-gripper pose $\{\mathbf{p}, \mathbf{R}, o\}$ is then computed as:

$$\mathbf{p} = \mathbf{j}_w + \lambda(\mathbf{j}_p - \mathbf{j}_w), \quad \mathbf{R} = \text{LookAt}(\mathbf{j}_m, \mathbf{j}_w, \mathbf{j}_t), \quad o \propto \|\mathbf{j}_f - \mathbf{j}_t\|_2 \in [0, 1]. \tag{2}$$

Position $\mathbf{p}$ interpolates between wrist $\mathbf{j}_w$ and palm center $\mathbf{j}_p$ to match the tool-contact point of a parallel gripper; orientation $\mathbf{R}$ is a rigid frame defined by middle-finger base $\mathbf{j}_m$, wrist $\mathbf{j}_w$, and index fingertip $\mathbf{j}_t$, aligning the grasp approach axis with the gripper's approach direction; and openness $o$ normalizes the spread between thumb fingertip $\mathbf{j}_f$ and index fingertip $\mathbf{j}_t$ to parallel binary gripper commands. Together, the scene point cloud and hand-as-gripper pose yield a unified representation $\{\mathcal{P}_t, \mathbf{p}_t, \mathbf{R}_t, o_t\}_{t=0}^{T}$ with a task language instruction $l$ that preserves the functionally relevant degrees of freedom for parallel-gripper manipulation, providing a shared geometric and action space across human and robot domains for aligned physical interaction reasoning in subsequent graph modeling and cross-domain alignment.

### 3.2 Structured Graph Modeling for Disentangled Interaction Entities

Upon the unified representation, we employ a structured graph transformer that disentangles spatial, semantic, and intentional entities through a heterogeneous graph construction coupled with a three-stage decomposed reasoning design, with each stage dedicated to one interaction entity via its own edge group (Figure 2). The stage-wise disentangled interaction reasoning forms multi-level features that are directly used for both unified action prediction and the cross-domain alignment in Section 3.3.

**Heterogeneous Graph Construction.** Given scene point clouds $\mathcal{P}_0$, gripper state $\{\mathbf{p}_0, \mathbf{R}_0, o_0\}$, and language instruction $l$, we construct a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with four vertex types, whose vertex set $\mathcal{V}$ partitions into four functionally distinct types capturing scene geometry, current embodiment state, task semantics, and future action intent:

$$\mathcal{V} = \{\mathcal{V}_{\text{spatial}}^{\text{scene}}, \mathcal{V}_{\text{spatial}}^{\text{gripper}}, \mathcal{V}_{\text{semantic}}, \mathcal{V}_{\text{intentional}}^{1:H}\}, \tag{3}$$

where $\mathcal{V}_{\text{spatial}}^{\text{scene}}$ represents scene points sampled via Furthest Point Sampling (FPS) [63] and encoded with pre-trained PointNet++ [64]; $\mathcal{V}_{\text{spatial}}^{\text{gripper}}$ encodes current gripper state through spatial anchor keypoints at the gripper frame and fingertips; $\mathcal{V}_{\text{semantic}}$ embeds the language instruction via frozen CLIP [65]; and $\mathcal{V}_{\text{intentional}}^{1:H}$ represents future gripper intentional actions over horizon $H$, initialized with learnable embeddings. The graph edges $\mathcal{E}$ employs three edge types: spatial edges $\mathcal{E}_{\text{spatial}}$ connect geometrically proximate vertices via $k$-nearest neighbors; semantic edges $\mathcal{E}_{\text{semantic}}$ form bidirectional links between the semantic vertex and all scene/gripper vertices; and intentional edges $\mathcal{E}_{\text{intentional}}$ establish directed links from the current gripper state to each future timestep and between consecutive future states. Each edge carries a learnable relational feature $\mathbf{e}_{ij}^r \in \mathbb{R}^{d_e}$ indexed by its endpoints $i, j \in \mathcal{V}$ and edge type $r \in \{\text{spatial}, \text{semantic}, \text{intentional}\}$, with $d_e$ denoting the feature dimension.

**Three-Stage Interaction Reasoning.** We process the graph through three specialized stages that progressively build interaction understanding via edge-conditioned graph attention [66] $\mathrm{GAttn}$:

$$\mathcal{V}^{(1)} = \mathrm{GAttn}\big(\mathcal{V}, \mathcal{E}_{\text{spatial}}\big); \quad \mathcal{V}^{(2)} = \mathrm{GAttn}\big(\mathcal{V}^{(1)}, \mathcal{E}_{\text{semantic}}\big); \quad \mathcal{V}^{(3)} = \mathrm{GAttn}\big(\mathcal{V}^{(2)}, \mathcal{E}_{\text{intentional}}\big). \tag{4}$$

Each stage updates only vertices incident to the active edge set, so $\mathcal{V}^{(s)}$ denotes the full vertex set after stage $s$. Within each $\mathrm{GAttn}$ call, vertex hidden representations $\mathbf{h}_i \in \mathbb{R}^d$ with dimension $d$ are iteratively updated over $L$ message-passing layers. For an edge set $\mathcal{E}^r$ with relational edge features $\mathbf{e}_{ij}^r \in \mathbb{R}^{d_e}$, the per-layer attention coefficient and aggregated message from neighbor $j \in \mathcal{N}^r(i)$ to vertex $i$ are:

$$\alpha_{ij}^r = \mathrm{softmax}_{j \in \mathcal{N}^r(i)}\left(\frac{(\mathbf{W}_Q \mathbf{h}_i)^\top (\mathbf{W}_K \mathbf{h}_j + \mathbf{W}_E^{(a,r)} \mathbf{e}_{ij}^r)}{\sqrt{d}}\right), \quad \mathbf{m}_{ij}^r = \mathbf{W}_V \mathbf{h}_j + \mathbf{W}_E^{(v,r)}\, \mathbf{e}_{ij}^r, \tag{5}$$

with the update rule $\mathbf{h}_i \leftarrow \mathbf{h}_i + \mathrm{FFN}\big(\sum_{j \in \mathcal{N}^r(i)} \alpha_{ij}^r\, \mathbf{m}_{ij}^r\big)$ applied once per layer, where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d}$ are shared query/key/value projections and $\mathbf{W}_E^{(a,r)}, \mathbf{W}_E^{(v,r)} \in \mathbb{R}^{d \times d_e}$ are relation-type-specific projections injecting edge features into attention and value computations. After $L$ layers in stage $s$, an MLP head summarizes $\mathcal{V}^{(s)}$ into a dense stage embedding, yielding $\mathbb{E}_{\text{spatial}}$, $\mathbb{E}_{\text{semantic}}$, and $\mathbb{E}_{\text{intentional}}$ for cross-domain alignment through disentangled interaction reasoning.

The three disentangled interaction modeling stages are realized through graph reasoning as follows. *Stage 1 (Spatial Modeling)* activates $\mathcal{E}_{\text{spatial}}$ so that each vertex aggregates local geometric neighborhood information, yielding $\mathcal{V}^{(1)}$ that encodes spatial interaction entity, where scene vertices capture geometric context and gripper vertices develop structural awareness, providing the low-level feature for cross-domain alignment. *Stage 2 (Semantic Modeling)* activates $\mathcal{E}_{\text{semantic}}$ to couple the task-level semantic vertex with spatial entities through bidirectional flow, simultaneously broadcasting task objectives and collecting spatial configuration, yielding $\mathcal{V}^{(2)}$ that encodes semantic instruction-level interaction patterns, where task-conditioned activation concentrates on task-relevant vertices, providing the mid-level feature for alignment. *Stage 3 (Intentional Modeling)* activates $\mathcal{E}_{\text{intentional}}$ to extend reasoning along the time axis, propagating information from the current gripper state to each future timestep and between consecutive future states, yielding $\mathcal{V}^{(3)}$ that encodes intentional interaction entity, where coherent multi-step action sequences respect geometric and semantic constraints, providing the high-level feature for cross-domain transfer and unified human-robot action prediction.

This three-stage design progressively composes physical interaction from geometric to task-aware to horizon-aware abstractions, where the complementary stage embeddings $\mathbb{E}_{\text{spatial}}, \mathbb{E}_{\text{semantic}}, \mathbb{E}_{\text{intentional}}$

collectively characterize the manipulation spectrum across embodiments and feed directly into cross-domain alignment (Section 3.3). The explicit modeling of $\mathcal{V}^{1:H}_{\text{intentional}}$ further enables seamless unified action prediction for both human and robot via simple MLP heads without additional modules.

### 3.3 Cross-Domain Learning via Hierarchical Interaction Alignment

To transfer physical interaction knowledge from human videos to robot tasks at both fine-grained continuous and coarse abstract levels in an embodiment-invariant manner, we introduce a cross-domain learning framework over multi-level graph interaction embeddings $\mathbb{E}_{\text{spatial}}, \mathbb{E}_{\text{semantic}}, \mathbb{E}_{\text{intentional}}$, where an Interaction Abstraction Module quantizes discrete primitives, upon which a hierarchical multi-granular alignment loss jointly matches both distributions via MMD [17] and KL divergence, optimized in a two-stage pipeline of human video pre-training and robot demonstration fine-tuning.

**Interaction Abstraction Module.** We introduce an Interaction Abstraction Module (IAM) that quantizes continuous interaction representations into discrete primitives via vector quantization [67]. We construct a unified interaction embedding by concatenating features from the three graph transformer stages: $\mathbb{E}_{\text{int}} = \left[\mathbb{E}_{\text{spatial}}, \mathbb{E}_{\text{semantic}}, \mathbb{E}_{\text{intentional}}\right]$, where $\mathbb{E}_{\text{spatial}}$, $\mathbb{E}_{\text{semantic}}$, and $\mathbb{E}_{\text{intentional}}$ encode geometric, task-aware, and dynamic interaction patterns respectively. Specifically, IAM maps $\mathbb{E}_{\text{int}}$ to abstraction $\bar{\mathcal{A}}_{\text{int}}$ from a learnable codebook $\mathcal{C} = \{c_k\}_{k=1}^{K}$ via nearest-neighbor (NN) assignment (Figure 2):

$$\bar{\mathcal{A}}_{\text{int}} = c_{k*}, \quad k* = \arg\min_{k} \|\mathbb{E}_{\text{int}} - c_k\|_2, \tag{6}$$

and optimized via the vector-quantization objective:

$$\mathcal{L}_{\text{VQ}} = \left\|\text{sg}[\mathbb{E}_{\text{int}}] - c_{k*}\right\|_2^2 + \beta \left\|\mathbb{E}_{\text{int}} - \text{sg}(c_{k*})\right\|_2^2, \tag{7}$$

where $\text{sg}(\cdot)$ is the stop-gradient operator and $\beta$ balances the codebook and commitment terms. The resulting discrete abstraction primitives $\bar{\mathcal{A}}_{\text{int}}$ capture interaction structure shared across embodiments, which we anticipate to approximate generic manipulation phases (e.g., reaching, grasping, pushing), serving as the coarse abstraction alignment target across human and robot domains.

**Hierarchical Multi-Granular Interaction Alignment.** Building upon $\mathbb{E}_{\text{int}}$ and $\bar{\mathcal{A}}_{\text{int}}$ from IAM, we conduct hierarchical alignment of human-robot interaction distributions at two complementary granularities: kernel MMD [17] on continuous embeddings and KL divergence on discrete abstraction primitive usage. During human video pre-training, we maintain two reference statistics over the human interaction embedding $\mathbb{E}^{\mathcal{H}}_{\text{int}}$ and its codebook assignment $\bar{\mathcal{A}}^{\mathcal{H}}_{\text{int}}$: a reservoir $\mathcal{X}^{\mathcal{H}}$ of continuous embeddings aggregated via exponential moving average, and a primitive usage distribution $P^{\mathcal{H}}(k)$ accumulated over codebook index $k$ with momentum $\alpha$. Together they capture fine-grained interaction geometry and coarse primitive usage, and are frozen before robot fine-tuning begins to serve as alignment targets. During robot fine-tuning, we compute the alignment loss $\mathcal{L}_{\text{algn}}$ over the robot-side mini-batch embeddings $\mathcal{X}^{\mathcal{R}}$ and primitive usage $P^{\mathcal{R}}(k)$, penalizing continuous embedding drift via MMD and discrete distribution shift via KL divergence:

$$\begin{aligned} \widehat{\text{MMD}}^2(\mathcal{R}, \mathcal{H}) &= \tfrac{1}{m^2}\textstyle\sum_{i,j} \kappa(x_i^{\mathcal{R}}, x_j^{\mathcal{R}}) - \tfrac{2}{mn}\sum_{i,j} \kappa(x_i^{\mathcal{R}}, x_j^{\mathcal{H}}) + \tfrac{1}{n^2}\sum_{i,j} \kappa(x_i^{\mathcal{H}}, x_j^{\mathcal{H}}), \\ \mathcal{L}_{\text{algn}} &= \widehat{\text{MMD}}^2(\mathcal{R}, \mathcal{H}) + \lambda_{\text{kl}} \sum_{k=1}^{K} P^{\mathcal{R}}(k) \log \frac{P^{\mathcal{R}}(k)}{P^{\mathcal{H}}(k)}, \end{aligned} \tag{8}$$

where $\mathcal{R}$ and $\mathcal{H}$ denote the robot and human domains, $x_i^{\mathcal{R}} \in \mathcal{X}^{\mathcal{R}}$ and $x_j^{\mathcal{H}} \in \mathcal{X}^{\mathcal{H}}$ are robot and human interaction embeddings sampled at batch indices $i, j$ with batch sizes $m$ and $n$, and $\kappa(x, y) = \exp(-\|x - y\|^2/2\sigma^2)$ is an RBF kernel with bandwidth $\sigma$ set via the median heuristic [17]. The MMD estimator measures distributional discrepancy between continuous embedding spaces, while the KL term with weight $\lambda_{\text{kl}}$ aligns primitive usage distributions $P^{\mathcal{D}}(k)$ over codebook entries $c_k$ across domains $\mathcal{D} \in \{\mathcal{H}, \mathcal{R}\}$ with $K$ denoting the codebook size. Together, both terms align human and robot interaction distributions jointly at fine-grained continuous and coarse discrete granularities.

**Two-Stage Optimization Pipeline.** Our framework is optimized in two distinct stages: *Stage 1 (Pre-Training on Human Videos)* trains on human manipulation videos with:

$$\mathcal{L}_{\text{human}} = \mathcal{L}^{\mathcal{H}}_{\text{action}} + \lambda_{\text{VQ}}\, \mathcal{L}_{\text{VQ}}, \tag{9}$$

where $\mathcal{L}^{\mathcal{H}}_{\text{action}}$ is action prediction MSE loss and $\lambda_{\text{VQ}}$ balances the VQ-VAE objective. $\mathcal{L}^{\mathcal{H}}_{\text{action}}$ supervises action prediction on human videos by $\{\mathbf{p}_t, \mathbf{R}_t, o_t\}_{t=1}^{T}$, while $\mathcal{L}_{\text{VQ}}$ trains IAM to learn discrete

Table 1: Multi-task comparison results with state-of-the-art methods in RLBench environment. $^{1}$, $^{10}$, and $^{100}$ denote the number of demonstrations.

| Method | OD | SB | SD | MG | TT | PD | CJ | DS | StB | ScB | PS | PW | PC | SS | PB | IP | SC | PM | RD | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ManiGaussian [68] | 76 | 24 | 64 | 60 | 56 | 48 | 28 | 92 | 12 | 32 | 74 | 52 | 38 | 14 | 20 | 8 | 12 | 41 | 88 | 44.2 |
| PerAct [69] | 88 | 74 | 52 | 70 | 88 | 51 | 55 | 90 | 26 | 18 | 86 | 45 | 28 | 17 | 93 | 6 | 2 | 38 | 72 | 52.6 |
| HiveFormer [70] | 52 | 64 | 28 | **100** | 80 | 68 | 52 | 76 | 8 | 8 | 76 | 80 | 32 | 8 | 84 | 0 | 0 | 34 | 68 | 48.3 |
| PolarNet [71] | 84 | 56 | 52 | **100** | 80 | 32 | 36 | 92 | 4 | 44 | 84 | 40 | 12 | 12 | 96 | 4 | 8 | 36 | 70 | 49.6 |
| RVT [72] | 71 | 82 | 72 | 88 | 94 | 88 | 52 | 99 | 29 | 48 | 91 | 91 | 50 | 36 | **100** | 11 | 26 | 58 | 82 | 66.7 |
| Act3D [73] | 78 | 96 | 86 | 95 | 94 | 91 | 97 | 81 | 4 | 33 | 95 | 59 | 67 | 30 | 94 | 24 | 10 | 52 | 78 | 66.5 |
| 3D Diffuser [74] | 90 | 98 | 84 | 97 | 99 | 96 | 96 | **100** | 68 | 82 | 98 | **94** | 86 | 44 | 98 | 66 | 47 | 76 | 88 | 84.6 |
| LLARVA [3] | 60 | **100** | 84 | 80 | 56 | 42 | 28 | 52 | 0 | 8 | 54 | 12 | 28 | 6 | 56 | 10 | 8 | 44 | 52 | 41.1 |
| ARM4R [2] | 89 | 86 | 72 | 94 | 62 | 58 | 24 | 78 | 4 | 10 | 62 | 36 | 44 | 8 | 67 | 18 | 14 | 92 | 78 | 52.4 |
| MimicPlay [32] | 48 | 30 | 35 | 52 | 58 | 40 | 22 | 61 | 4 | 8 | 44 | 28 | 30 | 4 | 52 | 6 | 4 | 38 | 55 | 32.6 |
| EgoMimic [30] | 55 | 42 | 44 | 61 | 64 | 47 | 31 | 68 | 6 | 12 | 51 | 35 | 38 | 5 | 60 | 10 | 8 | 45 | 62 | 39.2 |
| ours$^{1}$ | 77 | 32 | 53 | 68 | 88 | 63 | 67 | 83 | 23 | 42 | 72 | 48 | 58 | 3 | 97 | 28 | 8 | 47 | 81 | 54.6 |
| ours$^{10}$ | 89 | 98 | 82 | 97 | 99 | 91 | 88 | 92 | 61 | 73 | 99 | 78 | 83 | 67 | **100** | 77 | 66 | 87 | 93 | 85.3 |
| ours$^{100}$ | **97** | **100** | **94** | **100** | **100** | **98** | **97** | **100** | **82** | **90** | **100** | 93 | **96** | **84** | **100** | **91** | **86** | **97** | **100** | **95.0** |

Table 2: Comparison of point cloud methods on ManiSkill2 benchmark.

| Method (Policy) | PC | SC | TF | PISG | Al | In | Ex | Ha | Po | Fi | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SpUNet [75] (ACT [76]) | 74 | 22 | 39 | 81 | 28 | 1 | 3 | 84 | 10 | 66 | 40.8 |
| PointNet [77] (ACT [76]) | 84 | 35 | 0 | 77 | 40 | 1 | 27 | 83 | 14 | 91 | 45.2 |
| SpUNet [75] (Diffusion [78]) | 71 | 4 | 32 | 82 | 9 | 1 | 17 | 67 | 0 | 21 | 30.4 |
| PointNet [77] (Diffusion [78]) | 70 | 0 | 36 | 83 | 16 | 1 | 24 | 72 | 0 | 68 | 37.0 |
| PonderV2 [79] (ACT [76]) | 87 | 35 | 27 | 65 | 23 | 2 | 38 | 83 | 11 | 73 | 44.4 |
| SpUNet$^{*}$ [75] (Diffusion [78]) | 74 | 22 | 39 | 81 | 28 | 1 | 11 | 80 | 10 | 66 | 41.2 |
| PointNet$^{*}$ [77] (Diffusion [78]) | 90 | 24 | 37 | 87 | 29 | 1 | 28 | 78 | 13 | 69 | 45.6 |
| PointNet$^{\dagger}$ [77] (Diffusion [78]) | 94 | 15 | 55 | 89 | 14 | 0 | 28 | 82 | 14 | 74 | 46.5 |
| ours$^{1}$ | 63 | 28 | 42 | 71 | 31 | 3 | 22 | 68 | 17 | 58 | 40.3 |
| ours$^{10}$ | **96** | 52 | 73 | **94** | 58 | **8** | 51 | **89** | 42 | **95** | 65.8 |
| ours$^{100}$ | **96** | **78** | **88** | 92 | **74** | 7 | **68** | 87 | **61** | 93 | **74.4** |

interaction primitives from $\mathbb{E}_{\text{int}}$. *Stage 2 (Fine-Tuning on Robot Demonstrations)* freezes the IAM codebook $\mathcal{C}$ and human reference statistics $\mathcal{X}^{\mathcal{H}}, P^{\mathcal{H}}(k)$ accumulated in Stage 1, then fine-tunes with:

$$\mathcal{L}_{\text{robot}} = \mathcal{L}^{\mathcal{R}}_{\text{action}} + \lambda_{\text{algn}} \mathcal{L}_{\text{algn}}, \tag{10}$$

where $\mathcal{L}^{\mathcal{R}}_{\text{action}}$ is the robot loss supervised by robot demonstration data and $\lambda_{\text{algn}}$ weights alignment objective. The hierarchical alignment loss guides robot statistics $\mathcal{X}^{\mathcal{R}}, P^{\mathcal{R}}(k)$ toward the frozen human references $\mathcal{X}^{\mathcal{H}}, P^{\mathcal{H}}(k)$, enabling multi-granular cross-domain transfer of interaction knowledge.

# 4 Experiment

## 4.1 Experiment Setup

**Benchmark and Metric.** We evaluate our method on 19 RLBench [18] and 10 ManiSkill2 [19] tasks using a unified model. For both RLBench and ManiSkill2, we conduct 100 rollouts per task with randomized object poses per episode, following the standard evaluation protocol of each benchmark. Success rates are computed based on these rollout trials. For each task, we compute success rate based on these 100 rollout trials with 5 different seeds (Appendix D.1 for benchmark and task details).

**Baselines.** We compare against baselines in two categories split by whether they leverage human video data, matching the horizontal division in Table 1. *Robot-demonstration-only methods* train exclusively on robot data: ManiGaussian [68], PerAct [69], HiveFormer [70], PolarNet [71], RVT [72], and Act3D [73] voxelize or featurize the 3D workspace for action prediction, while 3D Diffuser [74] applies diffusion policies over scene representations. *Human-video pre-training methods* transfer knowledge from human demonstrations to robot tasks: LLARVA [3] and ARM4R [2] pre-train on human videos via trajectory-similarity objectives, while MimicPlay [32] and EgoMimic [30] clone human actions directly to robot execution, which are further fine-tuned in the same setting as our method. For ManiSkill2, baselines adopt ACT [76] and diffusion policy [78] with different perception encoders [75, 77, 79] (Appendix D.2 for baseline configurations and encoder variants).

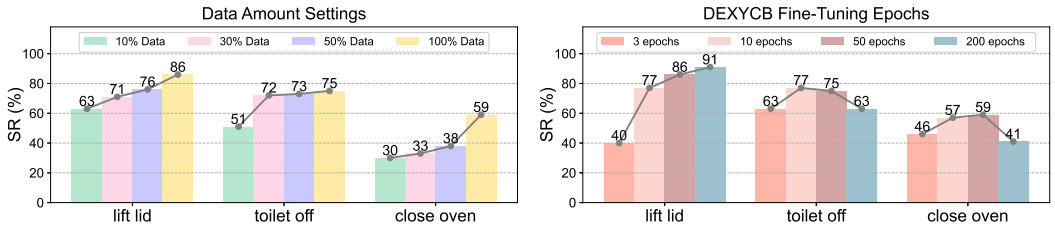


Figure 3: Zero-shot success rate of using human video pre-training with DexYCB correction.

Table 3: Performance across different pre-training configurations. "pt": full pre-training, "h-pt": half pre-training, "wo-pt": no pre-training. Left: RLBench tasks. Right: ManiSkill2 tasks.

| **RLBench** | 1 Demo | | | 10 Demos | | | **ManiSkill2** | 1 Demo | | | 10 Demos | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | pt | h-pt | wo-pt | pt | h-pt | wo-pt | | pt | h-pt | wo-pt | pt | h-pt | wo-pt |
| *lift lid* | 100 | 100 | 91 | 100 | 100 | 100 | *pick cube* | 63 | 54 | 43 | 96 | 90 | 86 |
| *toilet off* | 87 | 87 | 44 | 100 | 98 | 99 | *turn faucet* | 42 | 33 | 20 | 73 | 66 | 62 |
| *close oven* | 81 | 79 | 63 | 97 | 95 | 90 | *peg insertion* | 71 | 61 | 51 | 94 | 89 | 87 |
| *meat off grill* | 68 | 61 | 62 | 97 | 90 | 91 | *hang* | 68 | 58 | 46 | 89 | 83 | 81 |
| *turn tap* | 88 | 80 | 73 | 99 | 99 | 92 | *stack cube* | 28 | 21 | 13 | 52 | 44 | 40 |
| *put in safe* | 72 | 73 | 62 | 99 | 90 | 82 | *pour* | 17 | 11 | 5 | 42 | 34 | 31 |
| **Average** | **82.7** | **80.0** | **65.8** | **98.6** | **95.3** | **92.3** | **Average** | **48.2** | **39.7** | **29.7** | **74.3** | **67.7** | **64.5** |

### 4.2 Multi-Task Benchmark Comparison

As shown in Tables 1 and 2, our method achieves state-of-the-art performance across both benchmarks with strong data efficiency, Appendix D.3 for complete results with standard deviation. On RLBench, our method attains 85.3% average success rate with only 10 demonstrations, outperforming all baselines including 3D Diffuser [74] (84.6%) and ARM4R [2] (52.4%), scaling further to 95% with 100 demonstrations. Even with a single demonstration, our method achieves 54.6%, surpassing all human data pre-training baselines and demonstrating strong few-shot capability. On ManiSkill2, our method achieves 65.8% and 74.4% average success rates with 10 and 100 demonstrations, respectively, substantially outperforming point cloud-based baselines including PointNet$^{\dagger}$ (Diffusion) (46.5%) and PonderV2 (ACT) (44.4%). These consistent gains, particularly pronounced in low-data regimes, validate that unified representation, structured graph modeling, and hierarchical alignment together enable effective cross-embodiment transfer of interaction knowledge from human videos.

### 4.3 Model and Training Analysis

**Zero-Shot Performance.** We conduct zero-shot experiments on three RLBench tasks using a model pre-trained on Epic-Kitchens [12] and fine-tuned on DexYCB [20], without task-specific robot demonstrations. Figure 3 shows performance across pre-training data amounts and fine-tuning epochs. Increasing pre-training data from 10% to 100% consistently improves success rates on all three tasks, confirming that interaction priors learned from human videos transfer without task-specific supervision. DexYCB fine-tuning effects diverge by structural match: *lift lid* improves monotonically as it aligns with DexYCB's grasp-lift primitive, while *toilet off* and *close oven* peak and then decline due to overfitting to DexYCB's limited primitive coverage. Our method attains over 70% zero-shot average success rate at the best configuration, validating the cross-domain transferability of our unified interaction representation (Appendix E for protocol, task scope, and interpretation details).

**Human Manipulation Video Pre-Training.** We validate the impact of human video pre-training across different data amounts and demonstration settings. Table 3 shows results on 6 RLBench and 6 ManiSkill2 tasks under three configurations: full pre-training (pt), half pre-training (h-pt, training on half of Epic-Kitchens), and no pre-training (wo-pt), each evaluated with 1 and 10 demonstrations. Scaling from no pre-training to full pre-training yields substantial gains on RLBench: +16.9% (65.8%→82.7%) for 1-demo and +6.3% (92.3%→98.6%) for 10-demo settings. Consistent gains are observed on ManiSkill2: +18.5% (29.7%→48.2%) for 1-demo and +9.8% (64.5%→74.3%) for 10-demo settings. The markedly larger improvements in low-data regimes across both benchmarks demonstrate that human video pre-training particularly enhances few-shot transfer capability.

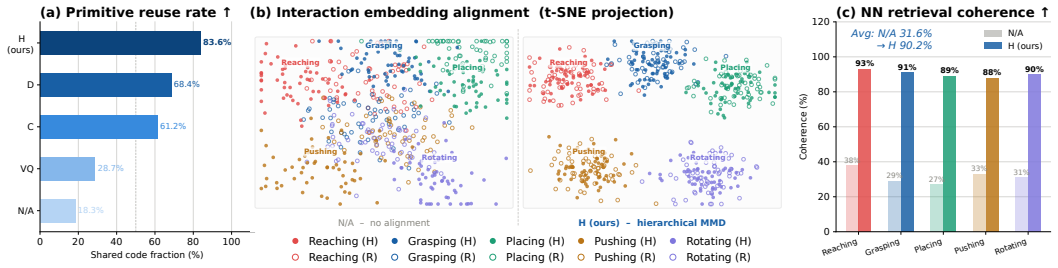


Figure 4: Cross-domain primitive alignment analysis on RLBench across five configurations (N/A: no alignment; VQ: codebook only; C: continuous MMD; D: discrete KL; H: hierarchical, ours). (a) Top-10 codebook reuse fraction between human and robot clips. (b) t-SNE of $\mathbb{E}_{\text{int}}$ by action phase (human: filled; robot: open), N/A vs. H. (c) NN retrieval coherence (N/A 31.6% $\rightarrow$ H 90.2%).

Table 4: Ablation of graph transformer vs. standard transformer variants on all 19 RLBench tasks (10 demos). ST: standard transformer. ST-TE: standard transformer with type embeddings. GT-S: graph transformer with single unified edge set. Ours: full three-stage structured graph transformer.

| Method | OD | SB | SD | MG | TT | PD | CJ | DS | StB | ScB | PS | PW | PC | SS | PB | IP | SC | PM | RD | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ST | 84 | 90 | 78 | **98** | 94 | 86 | 78 | 90 | 52 | 66 | 89 | 74 | 75 | 64 | 96 | 76 | 57 | 78 | 89 | 79.7 |
| ST-TE | **89** | 93 | 80 | 96 | **100** | 84 | 84 | 89 | 56 | 71 | 97 | 71 | 79 | 58 | 99 | 71 | 61 | 86 | 91 | 81.8 |
| GT-S | 87 | 97 | **84** | 96 | 96 | 90 | 87 | **93** | 58 | 70 | 95 | 77 | 82 | 66 | 98 | 75 | 64 | 84 | 92 | 83.7 |
| **Ours** | **89** | **98** | 82 | 97 | 99 | **91** | **88** | 92 | **61** | **73** | **99** | **78** | **83** | **67** | **100** | **77** | **66** | **87** | **93** | **85.3** |

**Cross-Domain Hierarchical Interaction Alignment.** To verify whether our hierarchical alignment achieves embodiment-invariant representations for interaction transfer, we ablate five configurations against three diagnostics shown in Figure 4: *primitive reuse rate* (fraction of top-10 codes shared between human and robot clips), *interaction embedding alignment* (t-SNE clustering of human-robot embeddings by action), and *NN retrieval coherence* (fraction of robot embeddings whose nearest human neighbor shares the same action phase). Without alignment (N/A), reuse and coherence are 18.3% and 31.6% (Figure 4(a),(c)), confirming that unregularized fine-tuning yields domain-specific patterns. VQ alone raises reuse to 28.7% via passive vocabulary sharing, while adding continuous MMD (C) or discrete KL (D) on top pushes reuse to 61.2% and 68.4%. This gap suggests that sharing a codebook alone is insufficient to induce consistent primitive usage across embodiments. Our hierarchical approach (H) achieves the best results across these diagnostics: reuse 83.6% and coherence 90.2%, with tight per-phase human-robot clustering in Figure 4(b) and reuse rate tracking success rate across strategies. In the corresponding six-task studies with 10 robot demonstrations per task, H improves average success over N/A from 93.2% to 98.6% on RLBench and from 65.7% to 74.3% on ManiSkill2 (Appendix D.4). The agreement between representation-level diagnostics and downstream success provides complementary evidence that the shared interaction structure is useful for policy transfer (Appendix D.5 for the full alignment diagnostics).

**Graph Transformer vs. Standard Transformer** A natural question is whether the structured graph transformer provides measurable benefit over a standard transformer that processes the same input tokens without explicit relational structure. To isolate this, we compare four architecture variants on all 19 RLBench tasks with 10 demonstrations. **Standard Transformer (ST)** is a vanilla transformer encoder that concatenates scene point features, gripper state, and language token as a flat sequence with full pairwise self-attention across all tokens. **ST with Type Embeddings (ST-TE)** augments ST with learnable type embeddings to distinguish scene, gripper, and language tokens, without explicit edge constraints. **Graph Transformer with Single Edge (GT-S)** applies our graph transformer with a single unified edge set connecting all vertices, removing the three-stage spatial/semantic/intentional decomposition. **Ours** is the complete three-stage structured graph transformer with dedicated spatial, semantic, and intentional edge groups. Table 4 shows a consistent improvement in average success rate at each step of the architectural hierarchy (79.7% $\rightarrow$ 81.8% $\rightarrow$ 83.7% $\rightarrow$ 85.3%). Type embeddings primarily help language-grounded tasks where token-type cues disambiguate instruction from scene, while graph structure and the three-stage decomposition primarily help spatially demanding and long-horizon tasks (*stack blocks*, *stack cups*, *insert peg*); on saturated short-horizon tasks the marginal

gain of each component is small or null, with occasional within-noise fluctuations on individual tasks. The full three-stage variant achieves the best overall average and the best per-task score on the majority of tasks, confirming that disentangling spatial, semantic, and intentional edge groups captures complementary interaction patterns that a single unified edge set conflates.

### 4.4 Real-Robot Experiment

**Experiment Setup.** We deploy our model on a 6-DoF industrial robotic manipulator equipped with an electrically actuated parallel-jaw gripper. Scene point clouds are reconstructed via VGGT [80] from a ZED2 external camera and an Intel RealSense 435i wrist-mounted camera, then masked by SAM [81]. We design four tasks assessing core competencies in object manipulation, semantic transfer, and spatial cognition: *move red bowl to plate*, *rotate cup*, *move blue bowl to plate*, and *clean trash*. For each task, we first refine the human video pre-trained model on a shared multi-task dataset, then fine-tune for 100 epochs with 5 or 10 demonstration trajectories per downstream task.

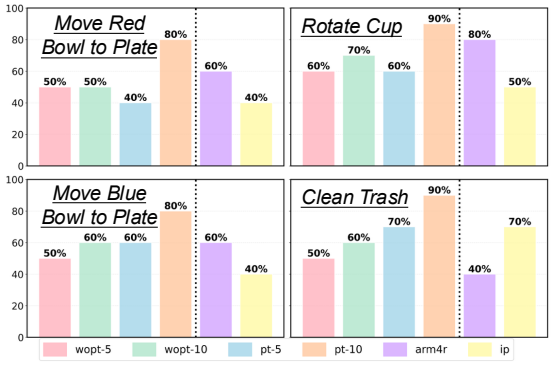


Figure 5: Success rate of our method using different training strategies and two baselines (arm4r [2] and ip [60]) in real-robot for 4 tasks.

**Real-World Results and Analysis.** In Figure 5, we compare our model with (pt) and without (wopt) human video pre-training under 5-shot and 10-shot settings against two baselines, ARM4R [2] and Instant Policy [60] for 10 trials. Our pt-10 configuration consistently achieves the highest success rate on all four tasks (80–90%), outperforming ARM4R by about 20% and Instant Policy by about 30% on average. The pt-vs-wopt gap of 20–30% at 10-shot directly validates the benefit of human video pre-training when sufficient demonstrations are available, while the consistent margins over both baselines confirm the advantage of our structured graph modeling with explicit interaction alignment over prior human-to-robot transfer approaches.

## 5 Conclusion

We address representational mismatch and the absence of explicit unified interaction reasoning in prior human-to-robot transfer through a unified representation via spatial tracking and hand-to-gripper estimation, over which a structured graph transformer disentangles spatial, semantic, and intentional interaction entities through three-stage decomposed reasoning over dedicated edge groups. To transfer interaction knowledge across domains, we introduce an Interaction Abstraction Module producing shared discrete primitives and a hierarchical multi-granular alignment loss combining kernel MMD with KL divergence at continuous and discrete granularities. Experiments on RLBench and ManiSkill2 establish state-of-the-art performance with strong data efficiency, achieving 95.0% and 74.4% with 100 demonstrations, and surpassing all human-video-to-robot transfer baselines even at the 1-demonstration setting. Real-world deployment further confirms the advantage of our method, outperforming prior transfer baselines by 10–50% on all four tasks, while zero-shot results show that models trained exclusively on human video can execute robot manipulation without robot-specific demonstrations, pointing toward scaling robot learning with internet-scale data. Our framework currently targets parallel-jaw grippers and primitives well covered by Epic-Kitchens (grasp-and-transport and grasp-and-rotate), so extending it to multi-fingered dexterous hands, sub-millimeter insertion, tool-tip contact, non-prehensile skills, and long-horizon tasks (our zero-shot results cover only three structurally aligned RLBench tasks) will require learned hand retargeting and broader or primitive-specific human video data. In addition, our point-cloud representation depends on monocular depth estimation and tracking that may degrade under severe occlusion or reflective and transparent objects, and scaling to noisier, in-the-wild internet video will require more robust perception and data filtering.

## References


[1] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alexander Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *Robotics: Science and Systems XIX*, 2023.

[2] Dantong Niu, Yuvan Sharma, Haoru Xue, Giscard Biamby, Junyi Zhang, Ziteng Ji, Trevor Darrell, and Roei Herzig. Pre-training auto-regressive robotic models with 4d representations. In *International Conference on Machine Learning*, pages 46693–46711. PMLR, 2025.

[3] Dantong Niu, Yuvan Sharma, Giscard Biamby, Jerome Quenum, Yutong Bai, Baifeng Shi, Trevor Darrell, and Roei Herzig. Llarva: Vision-action instruction tuning enhances robot learning. In *Conference on Robot Learning*, pages 3333–3355. PMLR, 2025.

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

[5] Tete Xiao, Ilija Radosavovic, Trevor Darrell, and Jitendra Malik. Masked visual pre-training for motor control. *arXiv preprint arXiv:2203.06173*, 2022.

[6] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, et al. Latent action pretraining from videos. In *The Thirteenth International Conference on Learning Representations*.

[7] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: A 3d vision-language-action generative world model. In *International Conference on Machine Learning*, pages 61229–61245. PMLR, 2024.

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

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[10] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018.

[11] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2020.

[12] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022.

[13] 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.

[14] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20406–20417, 2024.

[15] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

[16] Gyeongsik Moon. Bringing inputs to shared domains for 3d interacting hands recovery in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17028–17037, 2023.

[17] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The journal of machine learning research*, 13(1):723–773, 2012.

[18] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020.

[19] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. In *The Eleventh International Conference on Learning Representations*.

[20] 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.

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

[22] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[23] Dejie Yang, Zijing Zhao, and Yang Liu. Ar-vrm: Imitating human motions for visual robot manipulation with analogical reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6818–6827, 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] Yangcen Liu, Woo Chul Shin, Yunhai Han, Zhenyang Chen, Harish Ravichandar, and Danfei Xu. Immimic: Cross-domain imitation from human videos via mapping and interpolation. In *Conference on Robot Learning*, pages 834–858. PMLR, 2025.

[26] Chengbo Yuan, Rui Zhou, Mengzhen Liu, Yingdong Hu, Shengjie Wang, Li Yi, Chuan Wen, Shanghang Zhang, and Yang Gao. Motiontrans: Human vr data enable motion-level learning for robotic manipulation policies. *arXiv preprint arXiv:2509.17759*, 2025.

[27] Haoyang Weng, Yitang Li, Nikhil Sobanbabu, Zihan Wang, Zhengyi Luo, Tairan He, Deva Ramanan, and Guanya Shi. Hdmi: Learning interactive humanoid whole-body control from human videos. *arXiv preprint arXiv:2509.16757*, 2025.

[28] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. In *Conference on Robot Learning*, pages 4545–4565. PMLR, 2025.

[29] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8802–8810. IEEE, 2025.

[30] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13226–13233. IEEE, 2025.

[31] 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.

[32] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *Conference on Robot Learning*, pages 201–221. PMLR, 2023.

[33] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of machine learning research*, 17(59):1–35, 2016.

[34] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[35] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7167–7176, 2017.

[36] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[37] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 23–30. IEEE, 2017.

[38] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 3803–3810. IEEE, 2018.

[39] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12627–12637, 2019.

[40] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 international conference on robotics and automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[41] Abhishek Gupta, Coline Devin, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Learning invariant feature spaces to transfer skills with reinforcement learning. In *International Conference on Learning Representations*, 2017.

[42] 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.

[43] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Addressing appearance change in outdoor robotics with adversarial domain adaptation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1551–1558. IEEE, 2017.

[44] Qiang Zhang, Tete Xiao, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Learning cross-domain correspondence for control with dynamics cycle-consistency. In *International Conference on Learning Representations*.

[45] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4243–4250. IEEE, 2018.

[46] Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Pieter Abbeel, Sergey Levine, Kate Saenko, and Trevor Darrell. Adapting deep visuomotor representations with weak pairwise constraints. In *Algorithmic Foundations of Robotics XII: Proceedings of the Twelfth Workshop on the Algorithmic Foundations of Robotics*, pages 688–703. Springer, 2020.

[47] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[48] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[49] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. Pmlr, 2017.

[50] Tingwu Wang, Renjie Liao, Jimmy Ba, and Sanja Fidler. Nervenet: Learning structured policy with graph neural networks. In *International conference on learning representations*, 2018.

[51] Carmelo Sferrazza, Dun-Ming Huang, Fangchen Liu, Jongmin Lee, and Pieter Abbeel. Body transformer: Leveraging robot embodiment for policy learning. In *8th Annual Conference on Robot Learning*.

[52] Maximilian Sieb, Zhou Xian, Audrey Huang, Oliver Kroemer, and Katerina Fragkiadaki. Graph-structured visual imitation. In *Conference on Robot learning*, pages 979–989. PMLR, 2020.

[53] De-An Huang, Suraj Nair, Danfei Xu, Yuke Zhu, Animesh Garg, Li Fei-Fei, Silvio Savarese, and Juan Carlos Niebles. Neural task graphs: Generalizing to unseen tasks from a single video demonstration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8565–8574, 2019.

[54] Sateesh Kumar, Jonathan Zamora, Nicklas Hansen, Rishabh Jangir, and Xiaolong Wang. Graph inverse reinforcement learning from diverse videos. In *Conference on Robot Learning*, pages 55–66. PMLR, 2023.

[55] Haochen Shi, Huazhe Xu, Zhiao Huang, Yunzhu Li, and Jiajun Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks. *The International Journal of Robotics Research*, 43(4):533–549, 2024.

[56] Isabella Huang, Yashraj Narang, Ruzena Bajcsy, Fabio Ramos, Tucker Hermans, and Dieter Fox. Defgraspnets: Grasp planning on 3d fields with graph neural nets. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5894–5901. IEEE, 2023.

[57] Yixuan Wang, Yunzhu Li, Katherine Driggs-Campbell, Li Fei-Fei, and Jiajun Wu. Dynamic-resolution model learning for object pile manipulation. In *Robotics: Science and Systems*, 2023.

[58] Yifeng Zhu, Jonathan Tremblay, Stan Birchfield, and Yuke Zhu. Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6541–6548. Ieee, 2021.

[59] Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6322–6329. IEEE, 2022.

[60] Vitalis Vosylius and Edward Johns. Instant policy: In-context imitation learning via graph diffusion. In *The Thirteenth International Conference on Learning Representations*.

[61] Guangyan Chen, Te Cui, Meiling Wang, Chengcai Yang, Mengxiao Hu, Haoyang Lu, Yao Mu, Zicai Peng, Tianxing Zhou, Xinran Jiang, et al. Graphmimic: Graph-to-graphs generative modeling from videos for policy learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 1756–1768, 2025.

[62] Javier Romero, Dimitrios Tzionas, and Michael J Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Transactions on Graphics (TOG)*, 36(6):1–17, 2017.

[63] Yuval Eldar, Michael Lindenbaum, Moshe Porat, and Yehoshua Y Zeevi. The farthest point strategy for progressive image sampling. *IEEE transactions on image processing*, 6(9):1305–1315, 1997.

[64] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[65] 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.

[66] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in neural information processing systems*, 32, 2019.

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

[68] Guanxing Lu, Shiyi Zhang, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Manigaussian: Dynamic gaussian splatting for multi-task robotic manipulation. In *European Conference on Computer Vision*, pages 349–366. Springer, 2024.

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

[70] Pierre-Louis Guhur, Shizhe Chen, Ricardo Garcia Pinel, Makarand Tapaswi, Ivan Laptev, and Cordelia Schmid. Instruction-driven history-aware policies for robotic manipulations. In *Conference on Robot Learning*, pages 175–187. PMLR, 2023.

[71] Shizhe Chen, Ricardo Garcia, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. In *7th Conference on Robot Learning (CoRL 2023)*, 2023.

[72] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023.

[73] Theophile Gervet, Zhou Xian, Nikolaos Gkanatsios, and Katerina Fragkiadaki. Act3d: 3d feature field transformers for multi-task robotic manipulation. In *7th Annual Conference on Robot Learning*.

[74] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. In *Conference on Robot Learning*, pages 1949–1974. PMLR, 2025.

[75] Spconv Contributors. Spconv: Spatially sparse convolution library, 2022.

[76] Tony Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *Robotics: Science and Systems XIX*, 2023.

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

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

[79] Haoyi Zhu, Honghui Yang, Xiaoyang Wu, Di Huang, Sha Zhang, Xianglong He, Hengshuang Zhao, Chunhua Shen, Yu Qiao, Tong He, et al. Ponderv2: Improved 3d representation with a universal pre-training paradigm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[80] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[81] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[82] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.

[83] Haoyi Zhu, Yating Wang, Di Huang, Weicai Ye, Wanli Ouyang, and Tong He. Point cloud matters: Rethinking the impact of different observation spaces on robot learning. *Advances in Neural Information Processing Systems*, 37:77799–77830, 2024.

# A Implementation

**Graph Construction.** We sample $N_s = 512$ scene points via Furthest Point Sampling from each input point cloud. The spatial edge graph uses $k = 16$ nearest neighbors per vertex. All vertex features are projected to a shared hidden dimension $d = 256$ with edge feature dimension $d_e = 64$. Each $\mathrm{GAttn}$ stage runs $L = 4$ message-passing layers.

**Interaction Abstraction Module.** The IAM codebook $\mathcal{C}$ contains $K = 512$ entries, each of dimension $d = 768$, initialized via $k$-means on pre-training embeddings. For the VQ objective in Eq. (7), the commitment coefficient is $\beta = 0.25$, which weights the second term and supplies the gradient signal to the encoder through the stop-gradient operator. The first term, which nominally pulls each codebook entry $c_{k*}$ toward its assigned encoder output, is realized in practice via an exponential moving average update on $c_{k*}$ with decay $\gamma = 0.99$, following the standard EMA variant of VQ-VAE [67]; this provides a more stable codebook than direct gradient descent on the first term while remaining mathematically equivalent in expectation under stationary assignments. Concretely, for each codebook entry $c_k$ we maintain assignment count $N_k$ and accumulated embedding $m_k$, both updated as $N_k \leftarrow \gamma N_k + (1-\gamma)\, n_k$ and $m_k \leftarrow \gamma m_k + (1-\gamma) \sum_{i:\, k_i^* = k} \mathbb{E}_{\text{int},i}$, where $n_k$ is the number of mini-batch embeddings assigned to $c_k$, and the codebook entry is set to $c_k = m_k / N_k$. The human reservoir $\mathcal{X}^{\mathcal{H}}$ is maintained with momentum $\alpha = 0.99$ over $4096$ stored embeddings per level.

**Alignment Hyperparameters.** The RBF kernel bandwidth $\sigma$ is set per mini-batch via the median heuristic. The KL divergence weight in Eq. (8) is $\lambda_{\text{kl}} = 0.1$, and the overall alignment loss weight during robot fine-tuning in Eq. (10) is $\lambda_{\text{algn}} = 0.5$. The VQ loss weight during pre-training in Eq. (9) is $\lambda_{\text{VQ}} = 1.0$.

**Action Prediction.** The action head is a two-layer MLP predicting gripper pose $\{\mathbf{p}, \mathbf{R}, o\}$ at each future horizon step $h \in \{1, \ldots, H\}$, where $H = 8$. Position $\mathbf{p} \in \mathbb{R}^3$ and openness $o \in [0, 1]$ are regressed directly; orientation $\mathbf{R} \in SO(3)$ is parameterized as a 6D continuous rotation representation [82]. The action loss $\mathcal{L}_{\text{action}}$ is a sum of $\ell_2$ losses on position and openness plus geodesic loss on orientation.

**Training Setup.** We use the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.95$, weight decay $= 10^{-4}$) with a learning rate of $10^{-4}$ for pre-training and $5 \times 10^{-5}$ for fine-tuning, with cosine decay and $1000$ warmup steps. Pre-training runs for $50$ epochs on 76K Epic-Kitchens clips with batch size $64$. Robot fine-tuning runs for $100$ epochs. The IAM codebook $\mathcal{C}$ and human reference statistics $\mathcal{X}^{\mathcal{H}}, P^{\mathcal{H}}(k)$ are frozen at the start of fine-tuning.

**Geometry Encoder.** Scene vertices are encoded with a frozen pre-trained PointNet++ [64] backbone, producing per-point features of dimension $256$ before projection to $d = 256$. The spatial anchor keypoints for the gripper vertex consist of $9$ points sampled at the gripper frame origin, fingertip positions, and palm center, encoded via a linear layer.

# B Human Video Processing Visualization

Figure 6, 7, and 8 demonstrate the human video processing pipeline and results, which contain scene point cloud (column 2 and 3), hand joints (column 2), and hand as gripper pose (column 3). This human video processing method facilitates the construction of a unified representation within our graph-based learning architecture that bridges human demonstrations and robotic environments. The systematic integration of off-the-shelf vision models enables robust extraction of unified representations from human manipulation videos across varying illumination conditions and environmental contexts.

# C Sensitivity Analysis of Hand-to-Gripper Mapping

The hand-to-gripper mapping in Eq. (2) involves three design choices: the interpolation parameter $\lambda$ for position, the $\mathrm{LookAt}$ frame construction from selected hand joints, and the thumb-index spread as a proxy for gripper openness. We provide sensitivity analyses for each component and a discussion of scope.



Figure 6: Human video visualization about "pour water".

**Interpolation Parameter $\lambda$.** The parameter $\lambda$ controls where along the wrist-to-palm axis the effective tool-contact point is placed, with $\lambda = 0$ corresponding to the wrist joint and $\lambda = 1$ to the palm center. We sweep $\lambda \in \{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ and evaluate on six RLBench tasks with 10 demonstrations. As shown in Table 5, performance is relatively stable for $\lambda \in [0.3, 0.6]$, with a peak at $\lambda = 0.4$, which we adopt throughout. Performance degrades noticeably at the extremes ($\lambda = 0.0$ and $\lambda = 1.0$), confirming that the interpolation matters but that the method is not sensitive to the precise value within a reasonable range.

Table 5: RLBench average success rate (10 demos, 6 tasks) across interpolation parameter $\lambda$.

| $\lambda$ | 0.0 | 0.2 | 0.4 | 0.6 | 0.8 | 1.0 |
|---|---|---|---|---|---|---|
| Avg. SR (%) | 91.3 | 96.8 | **98.6** | 97.9 | 95.2 | 90.7 |

**LookAt Frame Construction.** The orientation $\mathbf{R}$ is defined by a rigid frame from three hand joints: middle-finger base $\mathbf{j}_m$, wrist $\mathbf{j}_w$, and index fingertip $\mathbf{j}_t$. We compare this choice against two alternatives: (i) using the thumb-base instead of the middle-finger base, and (ii) using a PCA-derived frame from all 21 hand joints. Results on the same six RLBench tasks are reported in Table 6. Our chosen frame achieves the best performance, attributable to the middle-finger base providing a more stable and task-relevant approach-axis estimate than the thumb-base (which moves substantially during grasping) or the PCA frame (which lacks a consistent semantic grounding in the grasp axis).

Table 6: RLBench average success rate (10 demos, 6 tasks) across orientation frame variants.

| Frame | Ours ($\mathbf{j}_m, \mathbf{j}_w, \mathbf{j}_t$) | Thumb-base variant | PCA frame |
|---|---|---|---|
| Avg. SR (%) | **98.6** | 95.1 | 93.4 |

**Openness Proxy.** The gripper openness $o$ is computed from the Euclidean distance between thumb fingertip $\mathbf{j}_f$ and index fingertip $\mathbf{j}_t$, normalized to $[0, 1]$. We compare against two alternatives: (i)

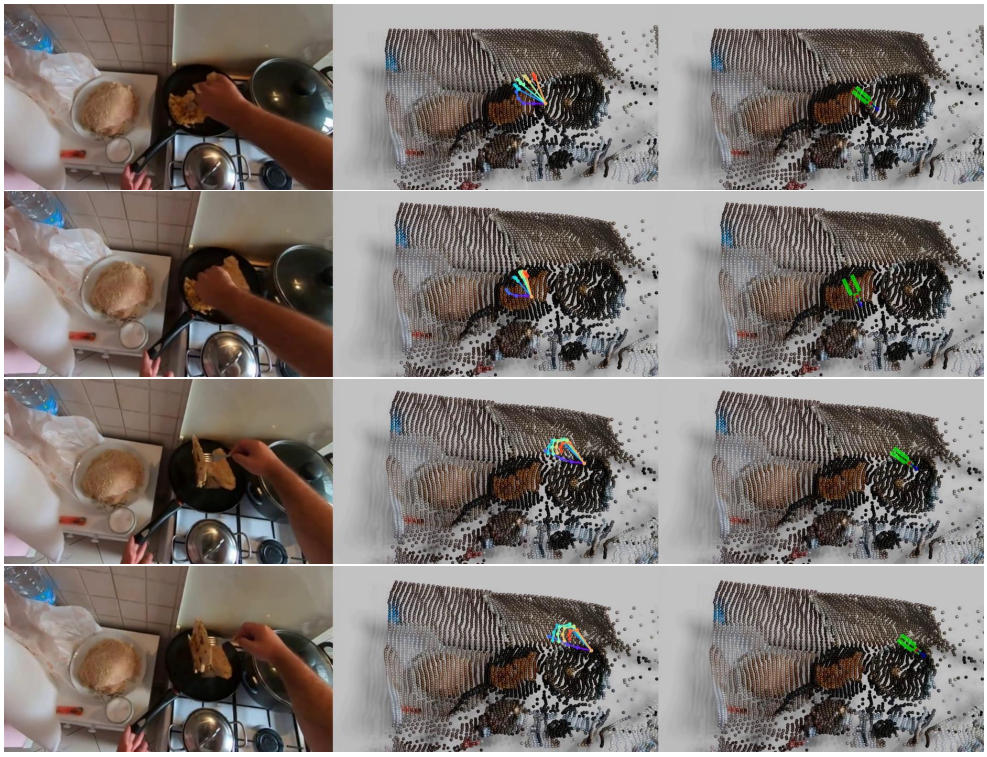

Figure 7: Human video visualization about “pick meat”.

mean pairwise distance across all five fingertips, and (ii) a learned openness regressor trained on the DexYCB [20] grasp annotations. Table 7 shows that our thumb-index proxy performs comparably to the learned regressor and slightly better than the mean pairwise distance, while requiring no additional supervision. The comparable performance of the learned regressor confirms that thumb-index spread is a reliable proxy for binary parallel-gripper state under the manipulation types present in Epic-Kitchens.

Table 7: RLBench average success rate (10 demos, 6 tasks) across openness proxy variants.

| Proxy | Thumb-index (ours) | Mean pairwise | Learned regressor |
|---|---|---|---|
| Avg. SR (%) | **98.6** | 96.3 | 98.1 |

**Scope and Failure Cases.** The hand-as-gripper mapping is designed for parallel-jaw grippers and manipulation primitives geometrically similar to pinch-like or power grasps, which constitute the dominant interaction type in both Epic-Kitchens and our robot benchmarks. The mapping is expected to degrade for (i) multi-fingered dexterous hands, where the binary openness signal discards functionally critical finger configuration, (ii) tool-use scenarios where the functional contact point lies at the tool tip rather than the hand, and (iii) non-prehensile manipulation such as pushing or surface wiping, where gripper orientation is less constrained by hand pose. In our real-robot experiments (Figure 5), the *rotate cup* task, which requires precise wrist orientation control, shows higher variance than pick-and-place tasks, consistent with orientation mapping being the least constrained degree of freedom. Extending the mapping to cover these cases—for example, via learned contact point estimation or dexterous hand retargeting—is a meaningful direction for future work.



Figure 8: Human video visualization about “open tap”.

# D Experiment

## D.1 Benchmark Details and Task Alias

We evaluate our method on two established robotics manipulation benchmarks: RLBench [18] and ManiSkill2 [19]. RLBench is a large-scale benchmark built on CoppeliaSim, offering a diverse suite of 100+ manipulation tasks with RGB-D observations and proprioceptive state inputs. We follow the standard multi-task evaluation protocol and report success rate (%) across 19 tasks under 1, 10, and 100 demonstrations per task. The 19 RLBench tasks and their abbreviations used in Tables 1 and 8 are as follows: OD=open drawer, SB=slide block, SD=sweep dustpan, MG=meat off grill, TT=turn tap, PD=put in drawer, CJ=close jar, DS=drag stick, StB=stack blocks, ScB=screw bulb, PS=put in safe, PW=place wine, PC=put in cupboard, SS=sort shape, PB=push buttons, IP=insert peg, SC=stack cups, PM=put money, and RD=reach and drag.

ManiSkill2 [19] is a unified benchmark for learning generalizable manipulation skills, featuring a wide variety of task categories including rigid-body, soft-body, and contact-rich manipulation with point cloud and RGB-D observations. We evaluate on 10 tasks spanning pick-and-place, insertion, and tool-use scenarios, and report success rate (%) under 1, 10, and 100 demonstrations per task. The 10 ManiSkill2 tasks and their abbreviations used in Tables 2 and 9 are as follows: PC=pick cube, SC=stack cube, TF=turn faucet, PISG=peg insertion side grasp, Al=align, In=insert, Ex=excavate, Ha=hang, Po=pour, and Fi=fill.

## D.2 ManiSkill2 Baselines Illustration

All point cloud baselines in Table 2 are adapted from OBSBench [83], which provides standardized encoders and policies on ManiSkill2. The encoders are SpUNet [75], a 3D sparse U-Net, and PointNet [77], a per-point MLP with symmetric aggregation, each paired with ACT [76] or Diffusion Policy [78]. We clarify the row symbols as follows:

Table 8: Multi-task comparison results with state-of-the-art methods in the RLBench environment (full results with standard deviation over 5 runs for our method). $^{1}$, $^{10}$, and $^{100}$ denote the number of demonstrations. Standard deviations are computed over 5 independent runs for our method; baseline results are reported as originally published. $^{\dagger}$ denotes results reproduced by running the official released code/model on tasks not reported in the original paper.

| Method | OD | SB | SD | MG | TT | PD | CJ | DS | StB | ScB |
|---|---|---|---|---|---|---|---|---|---|---|
| *3D Scene Representation Methods* | | | | | | | | | | |
| ManiGaussian [68] | 76 | 24 | 64 | 60 | 56 | 48$^{\dagger}$ | 28 | 92 | 12 | 32$^{\dagger}$ |
| PerAct [69] | 88 | 74 | 52 | 70 | 88 | 51 | 55 | 90 | 26 | 18 |
| HiveFormer [70] | 52 | 64 | 28 | **100** | 80 | 68 | 52 | 76 | 8 | 8 |
| PolarNet [71] | 84 | 56 | 52 | **100** | 80 | 32 | 36 | 92 | 4 | 44 |
| RVT [72] | 71 | 82 | 72 | 88 | 94 | 88 | 52 | 99 | 29 | 48 |
| Act3D [73] | 78 | 96 | 86 | 95 | 94 | 91 | 97 | 81 | 4 | 33 |
| 3D Diffuser [74] | 90 | 98 | 84 | 97 | 99 | 96 | 96 | **100** | 68 | 82 |
| *Human Video Pre-Training Methods* | | | | | | | | | | |
| LLARVA [3] | 60 | **100** | 84 | 80 | 56 | 42$^{\dagger}$ | 28 | 52 | 0 | 8 |
| ARM4R [2] | 89 | 86 | 72 | 94 | 62 | 58$^{\dagger}$ | 24 | 78 | 4 | 10 |
| MimicPlay$^{\dagger}$ [32] | 48 | 30 | 35 | 52 | 58 | 40 | 22 | 61 | 4 | 8 |
| EgoMimic$^{\dagger}$ [30] | 55 | 42 | 44 | 61 | 64 | 47 | 31 | 68 | 6 | 12 |
| ours$^{1}$ | 77±3.1 | 32±4.2 | 53±3.8 | 68±2.9 | 88±2.1 | 63±3.5 | 67±2.8 | 83±2.4 | 23±3.6 | 42±3.9 |
| ours$^{10}$ | 89±2.4 | 98±1.3 | 82±2.7 | 97±1.2 | 99±0.8 | 91±2.0 | 88±2.2 | 92±1.8 | 61±3.1 | 73±2.9 |
| ours$^{100}$ | **97**±1.1 | **100**±0.0 | **94**±1.4 | **100**±0.0 | **100**±0.0 | **98**±0.9 | **97**±1.0 | **100**±0.0 | **82**±2.0 | **90**±1.7 |

| Method | PS | PW | PC | SS | PB | IP | SC | PM | RD | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| *3D Scene Representation Methods* | | | | | | | | | | |
| ManiGaussian [68] | 74$^{\dagger}$ | 52$^{\dagger}$ | 38$^{\dagger}$ | 14$^{\dagger}$ | 20 | 8$^{\dagger}$ | 12$^{\dagger}$ | 41$^{\dagger}$ | 88 | 44.2 |
| PerAct [69] | 86 | 45 | 28 | 17 | 93 | 6 | 2 | 38$^{\dagger}$ | 72$^{\dagger}$ | 52.6 |
| HiveFormer [70] | 76 | 80 | 32 | 8 | 84 | 0 | 0 | 34$^{\dagger}$ | 68$^{\dagger}$ | 48.3 |
| PolarNet [71] | 84 | 40 | 12 | 12 | 96 | 4 | 8 | 36$^{\dagger}$ | 70$^{\dagger}$ | 49.6 |
| RVT [72] | 91 | 91 | 50 | 36 | **100** | 11 | 26 | 58$^{\dagger}$ | 82$^{\dagger}$ | 66.7 |
| Act3D [73] | 95 | 59 | 67 | 30 | 94 | 24 | 10 | 52$^{\dagger}$ | 78$^{\dagger}$ | 66.5 |
| 3D Diffuser [74] | 98 | **94** | 86 | 44 | 98 | 66 | 47 | 76$^{\dagger}$ | 88$^{\dagger}$ | 84.6 |
| *Human Video Pre-Training Methods* | | | | | | | | | | |
| LLARVA [3] | 54$^{\dagger}$ | 12 | 28$^{\dagger}$ | 6$^{\dagger}$ | 56 | 10$^{\dagger}$ | 8$^{\dagger}$ | 44 | 52 | 41.1 |
| ARM4R [2] | 62$^{\dagger}$ | 36 | 44$^{\dagger}$ | 8$^{\dagger}$ | 67 | 18$^{\dagger}$ | 14$^{\dagger}$ | 92 | 78 | 52.4 |
| MimicPlay$^{\dagger}$ [32] | 44 | 28 | 30 | 4 | 52 | 6 | 4 | 38 | 55 | 32.6 |
| EgoMimic$^{\dagger}$ [30] | 51 | 35 | 38 | 5 | 60 | 10 | 8 | 45 | 62 | 39.2 |
| ours$^{1}$ | 72±3.2 | 48±4.0 | 58±3.3 | 3±1.4 | 97±1.5 | 28±3.7 | 8±2.2 | 47±3.4 | 81±2.6 | 54.6±2.1 |
| ours$^{10}$ | **99**±0.7 | 78±2.5 | 83±2.3 | 67±3.0 | **100**±0.0 | 77±2.6 | 66±2.8 | 87±1.9 | 93±1.6 | 85.3±1.4 |
| ours$^{100}$ | **100**±0.0 | 93±1.3 | **96**±1.1 | **84**±1.8 | **100**±0.0 | **91**±1.5 | **86**±1.6 | **97**±0.9 | **100**±0.0 | **95.0**±0.8 |

- **No superscript**: default OBSBench configuration, using pre-sampling with Furthest Point Sampling, coordinate-only (XYZ) input, and training from scratch. *PonderV2 (ACT)* additionally uses a SpUNet34 backbone pre-trained with the PonderV2 [79] self-supervised objective.

- $*$: the optimal design identified in the OBSBench ablation, using *post-sampling* (FPS applied after the encoder) and *both color and coordinate* (RGB + XYZ) input channels.

- †: the strongest PointNet-based policy, starting from the PointNet$^{*}$ configuration and additionally adopting the end-effector coordinate frame with re-tuned diffusion hyperparameters.

### D.3 Benchmark Results with Standard Deviation Error

Tables 8 and 9 reproduce the main RLBench and ManiSkill2 results with standard deviations reported for our method across five independent evaluation runs. All baseline numbers are copied directly from the main paper as originally reported (no standard deviations available). The standard deviations confirm the statistical stability of our method, particularly at the 10- and 100-demonstration regimes.

Table 9: Comparison of point cloud methods on ManiSkill2 benchmark (full results with standard deviation over 5 runs for our method). Standard deviations are computed over 5 independent runs for our method; baseline results are reported as originally published.

| Method (Policy) | PC | SC | TF | PISG | Al |
|---|---|---|---|---|---|
| SpUNet [75] (ACT [76]) | 74 | 22 | 39 | 81 | 28 |
| PointNet [77] (ACT [76]) | 84 | 35 | 0 | 77 | 40 |
| SpUNet [75] (Diffusion [78]) | 71 | 4 | 32 | 82 | 9 |
| PointNet [77] (Diffusion [78]) | 70 | 0 | 36 | 83 | 16 |
| PonderV2 [79] (ACT [76]) | 87 | 35 | 27 | 65 | 23 |
| SpUNet* [75] (Diffusion [78]) | 74 | 22 | 39 | 81 | 28 |
| PointNet* [77] (Diffusion [78]) | 90 | 24 | 37 | 87 | 29 |
| PointNet† [77] (Diffusion [78]) | 94 | 15 | 55 | 89 | 14 |
| ours$^{1}$ | 63±3.8 | 28±4.1 | 42±3.5 | 71±3.2 | 31±3.9 |
| ours$^{10}$ | **96**±1.2 | 52±3.0 | 73±2.6 | **94**±1.4 | 58±2.8 |
| ours$^{100}$ | **96**±1.0 | **78**±2.1 | **88**±1.7 | 92±1.5 | **74**±2.3 |

| Method (Policy) | In | Ex | Ha | Po | Fi | Avg. |
|---|---|---|---|---|---|---|
| SpUNet [75] (ACT [76]) | 1 | 3 | 84 | 10 | 66 | 40.8 |
| PointNet [77] (ACT [76]) | 1 | 27 | 83 | 14 | 91 | 45.2 |
| SpUNet [75] (Diffusion [78]) | 1 | 17 | 67 | 0 | 21 | 30.4 |
| PointNet [77] (Diffusion [78]) | 1 | 24 | 72 | 0 | 68 | 37.0 |
| PonderV2 [79] (ACT [76]) | 2 | 38 | 83 | 11 | 73 | 44.4 |
| SpUNet* [75] (Diffusion [78]) | 1 | 11 | 80 | 10 | 66 | 41.2 |
| PointNet* [77] (Diffusion [78]) | 1 | 28 | 78 | 13 | 69 | 45.6 |
| PointNet† [77] (Diffusion [78]) | 0 | 28 | 82 | 14 | 74 | 46.5 |
| ours$^{1}$ | 3±1.2 | 22±3.6 | 68±3.4 | 17±2.8 | 58±3.7 | 40.3±2.3 |
| ours$^{10}$ | **8**±1.6 | 51±3.1 | **89**±1.8 | 42±2.9 | **95**±1.3 | 65.8±1.7 |
| ours$^{100}$ | 7±1.8 | **68**±2.4 | 87±1.9 | **61**±2.7 | 93±1.4 | **74.4**±1.3 |

Table 10: Cross-domain learning with different alignment strategies (10 demos). N/A: no alignment, VQ: vector quantization only (codebook without alignment loss), C: continuous MMD only, D: discrete KL only, H: hierarchical alignment combining both losses (ours).

| **RLBench** | N/A | VQ | C | D | H | **ManiSkill2** | N/A | VQ | C | D | H |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *lift lid* | 100 | 100 | 100 | 100 | 100 | *pick cube* | 88 | 89 | 94 | 90 | 96 |
| *toilet off* | 99 | 99 | 99 | 99 | 100 | *turn faucet* | 64 | 65 | 70 | 69 | 73 |
| *close oven* | 91 | 93 | 95 | 94 | 97 | *peg insertion* | 86 | 87 | 92 | 91 | 94 |
| *meat off grill* | 92 | 93 | 95 | 94 | 97 | *hang* | 82 | 83 | 87 | 80 | 89 |
| *turn tap* | 93 | 95 | 97 | 96 | 99 | *stack cube* | 42 | 44 | 49 | 49 | 52 |
| *put in safe* | 84 | 88 | 94 | 91 | 99 | *pour* | 32 | 34 | 38 | 39 | 42 |
| **Average** | **93.2** | **94.7** | **96.7** | **95.7** | **98.6** | **Average** | **65.7** | **67.0** | **71.7** | **69.7** | **74.3** |

### D.4 Cross-Domain Learning Success Rate Analysis

We extend the cross-domain alignment analysis from Section 4.3 by reporting downstream success rates for all five alignment configurations across 6 RLBench and 6 ManiSkill2 tasks under the 10-demonstration setting. Table 10 reveals a consistent ordering across both benchmarks: each component of the alignment loss contributes complementary gains, with the full hierarchical combination (H) outperforming all alternatives by +5.4% on RLBench (93.2%→98.6%) and +8.6% on ManiSkill2 (65.7%→74.3%) over the no-alignment baseline. Compared to VQ-only, which provides only the shared discrete vocabulary without explicit alignment pressure, H gains +3.9% on RLBench and +7.3% on ManiSkill2, confirming that the alignment losses themselves—not merely the codebook—drive transfer. Notably, continuous MMD alone (C) consistently outperforms discrete KL alone (D) on success rate (96.7% vs. 95.7% on RLBench; 71.7% vs. 69.7% on ManiSkill2), suggesting that continuous geometric alignment provides a richer training signal than discrete usage-distribution matching for downstream task success. The hierarchical combination achieves the best results on both benchmarks by leveraging complementary fine-grained continuous geometry and coarse discrete primitive structure.

### D.5 Cross-Domain Alignment: Detailed Diagnostics

*Setup.* We evaluate five alignment strategies—N/A (no alignment), VQ (codebook learned without alignment loss), C (continuous MMD only), D (discrete KL only), and H (hierarchical alignment combining both, ours)—on 6 RLBench tasks with 10 demonstrations each. Beyond downstream success rate (Table 10), we report three diagnostics that directly measure whether the learned interaction representation is embodiment-invariant.

*Primitive reuse rate.* For each strategy, we sample 500 matched pairs of human and robot clips annotated with the same action phase (reaching, grasping, placing, pushing, rotating) from a held-out split. For each pair we record the top-10 activated codebook entries and compute the Jaccard overlap. Without alignment (N/A) the average overlap is only 18.3%, confirming that unregularized fine-tuning yields largely domain-specific codes. VQ alone raises this marginally to 28.7% via passive vocabulary sharing through the shared codebook, but without alignment pressure the human-robot usage distributions remain largely distinct. Adding continuous MMD (C) and discrete KL (D) on top pushes overlap to 61.2% and 68.4% respectively—with D higher than C since KL directly optimizes codebook usage overlap—while the full hierarchical combination (H) reaches 83.6%, indicating that the majority of frequently activated codes are shared across embodiments.

*Nearest-neighbor retrieval coherence.* For each robot embedding $x \in \mathcal{X}^{\mathcal{R}}$ we retrieve its nearest neighbor in the frozen human reservoir $\mathcal{X}^{\mathcal{H}}$ under Euclidean distance and check whether the two segments share the same action-phase annotation. Under N/A this coherence is 31.6% on average across the five phases, barely above the 20% random baseline. Strategy H achieves coherence above 88% for every phase and 90.2% on average, demonstrating that robot embeddings are geometrically co-located with semantically corresponding human embeddings rather than merely projected into the same subspace by coincidence.

*MMD distance to human reservoir.* We report the biased $\widehat{\mathrm{MMD}}^2 \times 10^{-2}$ between the robot embedding batch $\mathcal{X}^{\mathcal{R}}$ and $\mathcal{X}^{\mathcal{H}}$ after fine-tuning, using an RBF kernel with bandwidth set by the median heuristic. MMD distance decreases monotonically from 8.74 (N/A) to 1.63 (H), and the residual distance under strategy D (4.52) is larger than under C (3.88) despite D achieving higher reuse, consistent with the view that continuous geometry carries richer transfer signal for downstream task success while discrete KL optimizes the codebook-overlap metric directly.

**Correlation with downstream performance.** Across the five alignment strategies, the strategy-level primitive reuse rate tracks the strategy-level average success rate on both benchmarks: Spearman $\rho = 0.90$ on RLBench and $\rho = 0.90$ on ManiSkill2, each with exact-permutation two-tailed $p = 0.083$ at $n = 5$. We report these correlations as descriptive summaries of the strategy ordering rather than as inferential claims: with only five alignment strategies, neither value reaches conventional significance, and we make no claim of statistical significance. The monotonic relationship between primitive reuse and downstream success rate is consistent with the interpretation that hierarchical alignment improves downstream performance by inducing genuinely embodiment-invariant interaction representations, rather than by regularization effects unrelated to cross-domain structure.

## E Zero-Shot Transfer: Protocol and Scope

### E.1 Zero-Shot Protocol Definition.

Zero-shot transfer in our setting refers to evaluation on robot tasks without any robot demonstration data at any stage. Concretely, the model is pre-trained on Epic-Kitchens [12] and refined on DexYCB [20] before evaluation on RLBench tasks without robot rollouts. Both Epic-Kitchens and DexYCB are human video datasets containing no robot demonstrations, no robot proprioception, and no robot-environment interaction. Zero-shot therefore refers specifically to the absence of robot data, which constitutes the meaningful transfer barrier in human-to-robot learning. Table 11 summarizes the protocol definitions.

The DexYCB refinement stage aligns hand-scene geometry to structured grasping annotations, which benefits tasks whose manipulation pattern overlaps with DexYCB coverage.

Table 11: Zero-shot protocol variants. EK: Epic-Kitchens pre-training; DexYCB: human video refinement; Robot FT: robot demonstration fine-tuning.

| Protocol | EK | DexYCB | Robot FT |
|---|---|---|---|
| Full model (10-demo) | ✓ | – | ✓ |
| Zero-shot (ours) | ✓ | ✓ | × |
| Zero-shot (strict) | ✓ | × | × |

### E.2 Task Scope of Zero-Shot Transfer.

Zero-shot evaluation is conducted on three RLBench tasks selected for their structural alignment with human manipulation priors in Epic-Kitchens: *lift lid*, *toilet off*, and *close oven*. These tasks share the grasp-and-transport or grasp-and-rotate primitive structure that is densely represented in Epic-Kitchens, making them the appropriate testbed for human-to-robot transfer without in-domain robot data.

Tasks requiring precise articulated-object interactions, long-horizon sequencing, or manipulation primitives absent from Epic-Kitchens (e.g., *insert peg*, *sort shape*) are not evaluated in the zero-shot setting, as the human interaction priors learned during pre-training do not provide sufficient coverage for reliable transfer without at least some in-domain demonstration data. This scoping reflects a genuine constraint of the current framework rather than a limitation of the evaluation design: the transferability of human priors is inherently task-structure-dependent, and the three evaluated tasks represent the regime where the claim is most directly supported by the learned representation.

### E.3 Interpretation.

The zero-shot results in Figure 3 support the specific claim that models trained exclusively on human video can execute robot manipulation tasks with grasp-and-transport structure without any robot demonstrations, achieving over 70% average success rate at the best configuration. This finding validates the core design principle that unified interaction representation with structured graph modeling captures physically grounded priors transferable across embodiments. Extending zero-shot transfer to more complex or structurally dissimilar tasks remains an open problem and a direction for future work.

## F Limitations and Future Work

While our framework achieves strong cross-embodiment transfer on the evaluated benchmarks, we acknowledge several scope boundaries that delimit where the current approach applies and point to directions for future extension.

**Gripper morphology.** The hand-to-gripper mapping in Eq. (2) is designed for parallel-jaw grippers with a binary openness signal. For multi-fingered dexterous hands, this abstraction discards functionally critical finger configuration information; extending the mapping to dexterous embodiments would require learned contact-point estimation or hand retargeting that preserves per-finger degrees of freedom. Our sensitivity analysis in Appendix C further examines the robustness of each component of this mapping.

**Manipulation primitive coverage.** Epic-Kitchens predominantly captures grasp-and-transport and grasp-and-rotate primitives, which align well with the manipulation patterns in RLBench and ManiSkill2. Manipulation skills that are structurally absent from Epic-Kitchens—such as precise insertion with sub-millimeter tolerances, tool-use where the functional contact point lies at the tool tip rather than the hand, and non-prehensile skills like pushing or surface wiping—are not adequately covered by current human video priors. Scaling human video coverage or introducing primitive-specific pre-training datasets is a meaningful direction for broadening applicability.

**Zero-shot scope.** Our zero-shot transfer results (Section 4.3) are demonstrated on three RLBench tasks selected for their structural alignment with Epic-Kitchens priors. Tasks requiring long-horizon

sequencing or primitives absent from pre-training remain outside the current zero-shot regime, reflecting a genuine constraint of human-video-only supervision rather than a limitation of the alignment framework itself.

**Observation modality.** Our representation relies on depth-informed point clouds derived from monocular video via depth estimation [15] and spatial tracking [14]. Degraded performance is expected under extreme visual conditions (severe occlusion, reflective surfaces, transparent objects) where these upstream components become unreliable. Incorporating multi-view or event-based sensing could mitigate these failure modes.

**Broader implications.** The ability to pre-train robot policies on internet-scale human video without robot-specific labels has positive implications for scaling robot learning, but also raises questions about data provenance and consent for human video datasets used in training. We use only publicly released research datasets (Epic-Kitchens, DexYCB) under their original licenses; downstream practitioners scaling this paradigm should similarly ensure dataset-level consent and licensing compliance.

## NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The introduction (Section 1) states three contributions—unified representation via spatial tracking and hand-to-gripper estimation, structured graph transformer disentangling spatial/semantic/intentional interactions, and hierarchical alignment combining MMD with KL divergence—all of which are supported by the method (Section 3) and validated by experiments (Section 4) on RLBench, ManiSkill2, and real-robot deployment.

   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: Limitations are discussed in detail in Appendix F, covering gripper morphology, manipulation primitive coverage, zero-shot scope, observation modality, and broader implications. Additional sensitivity analyses for the hand-to-gripper mapping are provided in Appendix C.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper is empirical and does not introduce theorems or formal theoretical results. All mathematical expressions in Section 3 (Eqs. 1–10) are method definitions with all symbols grounded in the surrounding prose.

   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 full method is described in Section 3 (unified representation, heterogeneous graph construction, three-stage interaction reasoning, IAM, and hierarchical alignment loss). Implementation details including all hyperparameters, codebook size, training schedule, optimizer settings, and encoder configurations are provided in Appendix Section A. Benchmark protocols and task aliases are detailed in Appendix D.1, and baseline configurations in Appendix D.2.

   Guidelines:

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

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

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

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

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

   Answer: [Yes]

   Justification: Code is `https://ia-graphmodeling.github.io`. All datasets used (Epic-Kitchens [12], DexYCB [20], RLBench [18], ManiSkill2 [19]) are publicly available under their original licenses, and preprocessing details are described in Section 3.1 and the Appendix.

   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: Training and test details are provided in Appendix Section A (graph construction, IAM, alignment hyperparameters, action prediction, optimizer, learning rate schedule, epochs, batch size). The evaluation protocol (100 rollouts per task, 5 seeds, randomized object poses) is described in Section 4 with full benchmark details in Appendix D.1.

   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: Standard deviations over 5 independent runs are reported for our method in Appendix D.3 (Tables 8 and 9). The Cross-Domain Alignment diagnostics in Appendix D.5 additionally report Spearman rank correlations ($\rho = 0.90$ on RLBench and $\rho = 0.90$ on ManiSkill2, each with exact-permutation two-tailed $p = 0.083$ at $n = 5$), reported as descriptive summaries of the strategy ordering rather than as inferential claims of statistical significance.

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: Compute resources and training schedule are specified in Appendix Section A: pre-training runs for 50 epochs on 76K Epic-Kitchens clips with batch size 64, robot fine-tuning runs for 100 epochs, using AdamW optimizer with cosine decay and 1000 warmup steps.

   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 uses only publicly released research datasets (Epic-Kitchens, DexYCB, RLBench, ManiSkill2) under their original licenses, involves no human subject experimentation by the authors, and aims at beneficial applications in robot learning that are aligned with the NeurIPS Code of Ethics.

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: Broader impacts are discussed in Appendix F under "Broader implications." The paper notes the positive impact of scaling robot learning through internet-scale human video, and also raises questions about data provenance and consent for human video datasets, recommending dataset-level licensing compliance for downstream practitioners.

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 new high-risk models or scraped datasets. The proposed method produces a robot manipulation policy trained on publicly released research datasets and evaluated on robotics simulators and a single real-robot platform; no generative content or dual-use artifacts are released.

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: All used datasets (Epic-Kitchens [12], DexYCB [20], RLBench [18], ManiSkill2 [19]) and pre-trained components (InterWild [16], PointNet++ [64], CLIP [65], ZoeDepth [15], SpatialTracker [14], VGGT [80], SAM [81]) are properly cited in the paper and used under their original research licenses.

    Guidelines:

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

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new datasets, pre-trained models, or other public assets.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or direct research with human subjects. Human video data is used only through pre-existing public research datasets (Epic-Kitchens and DexYCB), whose collection and consent protocols were handled by their original creators.

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. All human video data originates from publicly released research datasets whose original collection and IRB protocols are handled by their respective authors.

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: LLMs were used only for polishing the text of the manuscript and do not impact the core methodology, scientific rigor, or originality of the research.

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.