METIS: Multi-Source Egocentric Training for Integrated Dexterous Vision-Language-Action Model

Yankai Fu[1]*, Junkai Zhao[2]*, Ning Chen[1]*, Shaozhe Shan[1],
Guocai Yao[2], Pengwei Wang[2], Zhongyuan Wang[2], Shanghang Zhang[1]✉

[1]State Key Laboratory of Multimedia Information Processing, School of Computer Science, Peking University; [2]Beijing Academy of Artificial Intelligence
*Equal contribution, ✉Corresponding author

Project Webpage: https://aureleopku.github.io/METIS

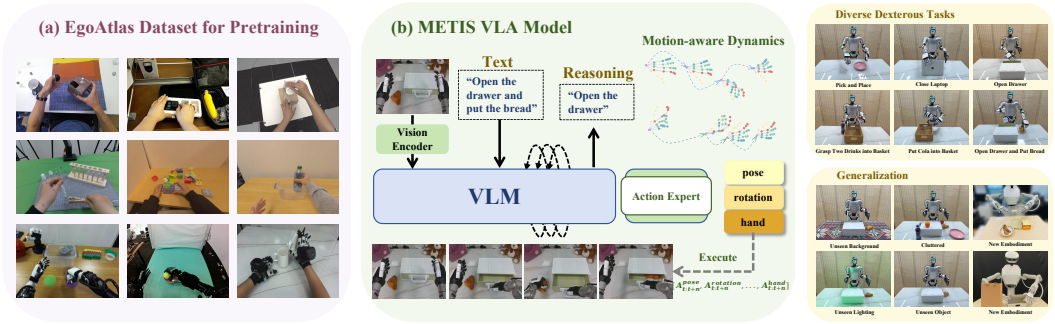


Figure 1: METIS is trained on a multi-source egocentric manipulation dataset EgoAtlas. It leverages motion-aware dynamics to extract manipulation-relevant dexterous features, and integrates reasoning and acting within a unified framework. METIS achieves strong performance across diverse dexterous manipulation tasks and exhibits remarkable generalization capability.

Abstract

Building a generalist robot that can perceive, reason, and act across diverse tasks remains an open challenge, especially for dexterous manipulation. A major bottleneck lies in the scarcity of large-scale, action-annotated data for dexterous skills, as teleoperation is difficult and costly. Human data, with its vast scale and diverse manipulation behaviors, provides rich priors for learning robotic actions. While prior works have explored leveraging human demonstrations, they are often constrained by limited scenarios and a large visual gap between human and robots. To eliminate these limitations, we propose METIS, a vision-language-action (VLA) model for dexterous manipulation pretrained on multi-source egocentric datasets. We first construct EgoAtlas, which integrates large-scale human and robotic data from multiple sources, all unified under a consistent action space. We further extract motion-aware dynamics, a compact and discretized motion representation, which provides efficient and expressive supervision for VLA training. Built upon them, METIS integrates reasoning and acting into a unified framework, enabling effective deployment to downstream dexterous manipulation tasks. Our method demonstrates exceptional dexterous manipulation capabilities, achieving highest average success rate in eight real-world tasks. Experimental results also highlight the superior generalization and robustness to out-of-distribution scenarios. These findings emphasize METIS as a promising step toward a generalist model for dexterous manipulation.

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

1 Introduction

Recent advances in vision-language-action (VLA) models have achieved remarkable progress toward general-purpose embodied intelligence [45, 19, 15, 68]. However, such models heavily rely on scarce and expensive real-world robot data, often collected via human teleoperation. Despite extensive efforts to build large-scale robotic datasets [34, 17, 5, 54], the effective scale of robot interaction data remains orders of magnitude lower than those used for LLM training [68], severely limiting the scalability and generalization of current VLAs. This issue is even more pronounced in dexterous manipulation, where acquiring high-quality demonstrations is extremely costly and complex. As a result, most existing VLA research focuses on simple gripper-based tasks [14, 26, 6, 30], leaving dexterous manipulation largely unexplored.

In contrast to the scarcity of robotic datasets, human data are vastly abundant and rich in semantic information, providing valuable behavioral priors. Previous works have explored learning task-relevant representations from human videos, such as affordance [1, 21], latent action [60, 6], and keypoint flow [65, 62]. Recent progress such as EgoVLA [59], Being-H0 [32], and H-RDT [2] leverage large-scale human datasets to pretrain VLA, providing a new paradigm for scaling robotic learning. However, human videos are often confined to specific household or workspace scenes (e.g., tabletop or kitchen), exhibiting uneven scene coverage and strong contextual biases. Moreover, there exists a substantial gap between human data and robot data, in both visual appearance and action space. Another line of research explores co-training strategies that jointly utilize self-collected human and robotic data. For example, HAT [39] learns a human policy and demonstrates that incorporating human data can significantly enhance the generalization and robustness of robotic policies, while MotionTrans [63] aligns human motions to robot-specific embodiments to achieve zero-shot skill transfer. These approaches collect manipulation-related human data but do not fully exploit the vast amount of human data available on the internet.

In this work, we investigate these problems by introducing **METIS, a vision-language-action (VLA) model** for dexterous manipulation pretrained on multi-source egocentric datasets. We first construct **EgoAtlas**, a multi-source egocentric dataset that covers large-scale internet human data, robot data, and enhanced human data collected through a wearable system. EgoAtlas spans four major categories and eight sources, all aligned under a unified action space. We further propose motion-aware dynamics, a compact and discretized representation designed for dexterous manipulation. It captures both visual and motion dynamics, providing efficient and expressive supervision for training VLA. Built upon them, METIS is pretrained on EgoAtlas, unifying reasoning and acting within a single framework. This design enables efficient fine-tuning and deployment on downstream dexterous tasks.

To comprehensively evaluate METIS, we conduct extensive real-world experiments on dexterous tasks. The comparative results show that our model achieves superior performance and efficiency, attaining the highest average success rate among all baseline VLAs. Besides, METIS also exhibits strong generalization to out-of-distribution scenarios, including unseen background, unseen object, unseen lighting condition, cluttered environment, and can be transferred to higher-DoF embodiments. Finally, ablation studies verify the contributions of both the multi-source egocentric dataset and the motion-aware dynamics to the overall system, highlighting the potential of learning robotic motion priors from human data. In summary, our contributions are as follows:

- We construct a multi-source egocentric manipulation dataset EgoAtlas, which integrates diverse human and robotic data sources under a unified state-action space.
- We present METIS, a VLA model for dexterous manipulation pretrained on EgoAtlas, which extracts motion-aware dynamics from dexterous hand motion and integrates reasoning and acting within a unified framework.
- We demonstrate the effectiveness and generalization of our method through a range of real-world experiments.

2 Related Work

2.1 Dexterous Manipulation

Dexterous manipulation is a key challenge in robotics, focusing on achieving human-like fine-grained manipulation skills. Traditional approaches are based on optimization and control algorithms

[22, 51, 27, 56], typically assuming access to known dynamics and object models, and focusing on trajectory planning under physical constraints. While achieving strong performance in specific scenarios, these methods often struggle to generalize across diverse real-world settings. Recent learning-based approaches, including reinforcement learning (RL) and imitation learning (IL), have made remarkable progress across various tasks such as grasping [55, 48], in-hand manipulation [37, 36], and tool use [41, 61]. RL methods learn precise and dexterous skills through large-scale training in simulation, but often suffer from a significant sim-to-real gap. In contrast, IL methods leverage expert demonstrations to achieve robust performance in real-world dexterous manipulation tasks [28, 10, 18, 9], but typically rely on expensive human teleoperation data. In this work, we address this data bottleneck by pretraining our model on large-scale multi-source egocentric data that combines human and robotic demonstration to learn motion priors efficiently.

2.2 Learning Dexterity from Human Data

Human data offer valuable priors for dexterous manipulation, featuring abundant samples, fine-grained hand motion, and rich semantic cues. There has been works on learning task-relevant representations from human videos, such as affordance [1, 21], latent action [60, 6], and keypoint flow [65, 62]. Ego-Only [50] utilizes the MAE framework [11] to extract actionable information from egocentric videos, while LAPA [60] and UniVLA [6] apply VQ-VAE [47] to extract latent action from large-scale unlabeled human data. However, learning directly from human videos introduces redundant, manipulation-irrelevant content and underemphasizes the critical role of hand motion. To mitigate this gap, HAT [39] and MotionTrans [63] adopt a co-training strategy on both human and robotic data with explicit motion information, improving the robustness of the policy and enabling zero-shot skill transfer. In this work, we learn action priors from human data through joint modeling of visual and motion dynamics to support VLA learning.

2.3 Vision-Language-Action Model

Vision-Language-Action (VLA) models have achieved unprecedented progress in recent years [45, 19, 15, 40], driven by advances in Vision-Language Models (VLMs) [42, 44, 57] and the availability of large-scale robotic datasets [34, 17, 5, 54, 13]. By processing multimodal inputs—such as visual observations and language instructions—these models enable robots to autonomously perform a wide range of tasks. Representative VLA models, such as RT-2 [68] and OpenVLA [19] generate action sequences autoregressively. In contrast, π_0 [4] and DexVLA [53] employ a diffusion-based policy, fitting continuous action distributions via iterative denoising. However, these approaches primarily focus on gripper-based manipulation, overlooking the rich motion and interaction dynamics inherent in dexterous tasks. Recently, several studies have extended VLA frameworks to dexterous manipulation. For example, GR00T N1 [3] learns latent representations from hybrid data to train a humanoid manipulation policy, while EgoVLA [59] and Being-H0 [32] leverage large-scale human demonstrations for VLA pretraining, enabling the acquisition of motion priors. Despite their impressive results, the scene homogeneity of human videos leads to bias and a large visual gap from robotic observations. To overcome this issue, we leverage multi-source egocentric data, introduce an enhanced human dataset, and develop an integrated VLA model for unified reasoning and acting.

3 EgoAtlas Dataset

3.1 Wearable System for Enhanced Human Data

Traditional hand motion capture systems rely on multi-camera or VR setups, which are limited by viewpoint dependency, occlusion, and constrained capture space. To overcome these issues, we develop a wearable glove–tracker system for portable, high-fidelity motion capture.

Wearable Hand Motion Collection System. We use Manus Quantum Metagloves to record precise 3D hand keypoints (25 per hand). A VIVE Tracker mounted on each glove provides the 6-DoF wrist pose for global localization. To capture egocentric visual observations, a head-mounted camera is used to record the first-person view during manipulation, providing visual information closely aligned with the operator's perspective, while another VIVE Tracker is attached to the headset to perform extrinsic calibration. This calibration aligns the

motion-capture and ego-camera coordinate frames, enabling the entire system to operate in the wild and ensuring scene diversity. We also provide a top-down third-person camera view which is optional and is not used for calibration, training, or inference in this work; it records RGB videos only for future research purposes. The entire system operates at 20 Hz, balancing motion fidelity with reliable multi-sensor synchronization.

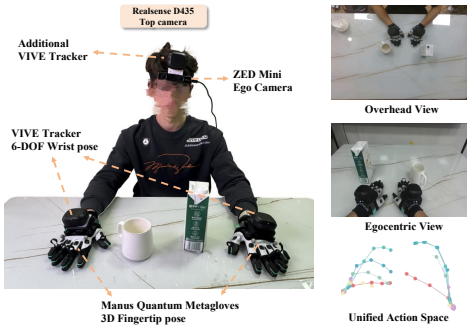


Figure 2: **Wearable Hand Motion Collection System.**

Subtask-Level Annotation. We performed subtask-level annotation to enrich the semantic structure of the dataset. Each trajectory is paired with a task-level language instruction, along with a fine-grained segmentation into multiple subtasks. This annotation design enables the VLA to focus on hand motion and supports hierarchical reasoning over long-horizon manipulation tasks. More details can be found in Appendix A.5.

3.2 Data Statistics and Processing

Built on Section 3.1, we introduce EgoAtlas, a large-scale multi-source egocentric dataset for VLA training. EgoAtlas integrates data from 8 heterogeneous sources spanning human and robot domains, including 10K self-collected high-fidelity trajectories using wearable system. The final dataset contains 343K trajectories and 89.72M image–action pairs, as summarized in Table 8. We apply weighted sampling to balance source distributions. More details are provided in Appendix A.7.

Different embodiments operate in distinct state and action space. To bridge these heterogeneous sources, we construct a unified state–action representation that abstracts away embodiment-specific differences while preserving manipulation-relevant motion. The state–action space is composed of two components: wrist pose and fingertip pose. The wrist pose is represented as an 18-dimensional vector consisting of a 3D position and a 6D continuous rotation per hand in the egocentric camera frame. The hand configuration is represented by a 30-dimensional fingertip pose, defined by the 3D positions of individual fingertips in the wrist coordinate frame. We apply robot-specific FK/IK (Forward Kinematics and Inverse Kinematics) and human motion retargeting algorithms to project heterogeneous motion representations into a shared state–action space, while aligning all data within a consistent coordinate system. More details can be found in Appendix A.3, A.4 and A.6.

4 Method

4.1 Problem Formulation

Our goal is to learn a VLA model for dexterous manipulation from multi-source egocentric data. Formally, given a collection of human and robot trajectories $D = D_{robot} \cup D_{human}$ from the EgoAtlas dataset, each trajectory can be represented as a sequence of observation–action pairs $\tau = \{(o_t, a_t)\}_{t=1}^{T}$, where observation $o_t = \{I_t, S_t\}$ consists of the egocentric image input and the proprioceptive state. The training objective is to learn a policy $\pi_\phi(a_t|o_t, l)$ that predicts actions conditioned on egocentric observations o_t and language instruction l.

To bridge the embodiment gap, we construct a unified state–action space which includes an 18D wrist pose P_t^w (3D position and 6D rotation vector per hand) and a 30D finger pose P_t^f capturing the 3D position of individual fingertips. Here we describe the design of the three components in detail.

- Image Observation I_t: We use egocentric image from a first-person camera, which provides a viewpoint that focuses on fine-grained interaction details.
- Wrist Pose P_t^w: We represent the wrist pose as the 3D position and the 6D rotation vector following Zhou et al. [67]. They are relative to the camera coordinate frame.
- Finger Pose P_t^f: We represent the finger pose using 3D fingertip positions, defined in the wrist coordinate frame.

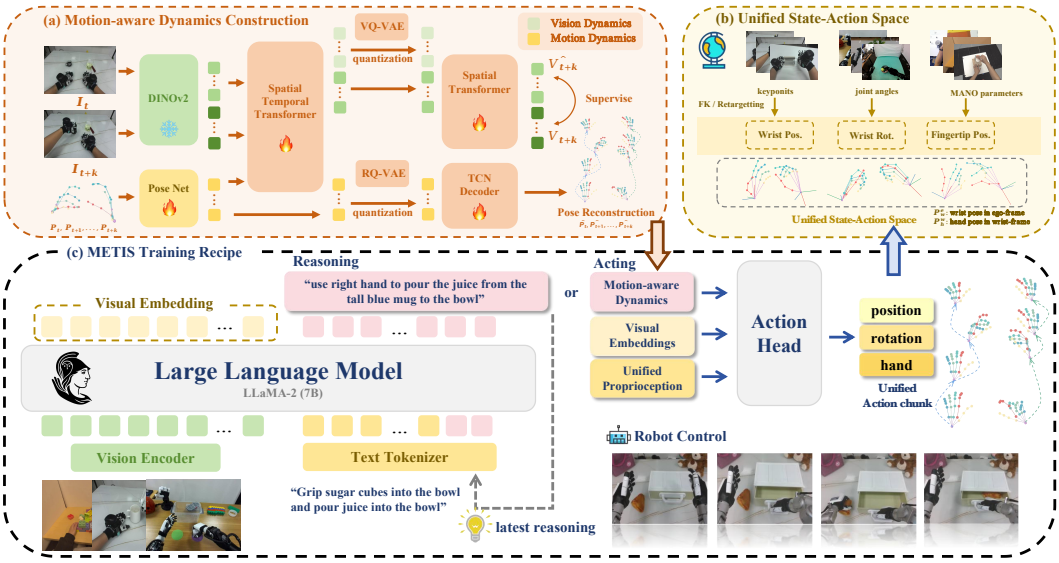


Figure 3: **Overview Framework** (a) We construct an expressive yet compact representation to capture the dynamics in dexterous manipulation. (b) METIS is pretrained on multi-source EgoAtlas dataset, where human and robot actions are align under a unified action space. (c) METIS integrates reasoning and acting within a framework, enabling effective deployment to downstream tasks.

4.2 Motion-Aware Dynamics Construction

Recent VLA models typically employ generic tokenizers to discretize continuous actions, enabling autoregressive sequence modeling for policy learning. However, as the number of action chunks and system degrees of freedom increase, discrete action representations inevitably lead to longer token sequences, which in turn increases the computational cost and latency of autoregressive generation. Additionally, this tokenization approach often struggles to capture fine-grained motion details, which is particularly critical in dexterous manipulation, where subtle finger movements and contact interactions are essential.

To address these challenges, we propose a compact and discretized representation that effectively constructs motion-aware dynamics for dexterous manipulation. This representation provides an efficient and expressive supervision signal for VLA pretraining. The proposed dynamics model consists of two key components, detailed as follows.

Visual Dynamics Discretization. A key challenge in egocentric dexterous manipulation is modeling how hand motion induces structured visual change. Instead of discretizing raw visual observations, we explicitly model *motion-conditioned visual dynamics* that capture the causal effect of manipulation on future visual states. Specifically, we employ an Inverse Dynamics Model based encoder $\mathcal{I}(D_{vis}|I_t, I_{t+k}, P_{t,t+1,\ldots,t+k})$ and a Forward Dynamics Model based decoder $\mathcal{F}(I_{t+k}|I_t, D_{vis})$. The encoder captures motion-relevant visual dynamics by integrating visual observations with continuous motion information, while the decoder is trained to predict future observation given the constructed visual dynamics. Our encoder includes both spatial and temporal transformer while the decoder only contains spatial transformer.

The visual dynamics D_{vis} are quantized with the VQ-VAE objective [47]. It maps continuous latent features into a finite set of discrete codebook embeddings, thereby enabling efficient and stable auto-regressive training for VLA. Following Bu et al. [6], we do not reconstruct image pixels, as raw pixels contain substantial redundancy and details irrelevant to the manipulation task. Instead, we use pretrained DINOv2 [33] to extract high-level semantic representations, that better reflect task-relevant visual dynamics.

Motion Dynamics Quantization. Learning motion priors is essential for VLA models. To this end, we introduce a separate set of discretized codebook embeddings D_{mot} to focus on motion capturing. We discretize hand motion trajectories into compact dynamics tokens that effectively capture proprioceptive information and reflect the fine-grained features of dexterous manipulation. We propose PoseNet as the encoder for 3D hand motion, combining multi-scale temporal convolutions

with trajectory self-attention to capture spatio-temporal dynamics. The continuous motion features are then quantized using RQ-VAE [25], which not only effectively prevents codebook collapse but also captures hierarchical motion patterns, from coarse to fine levels. A temporal convolutional network (TCN) [24] is used as the decoder to reconstruct motion trajectories during training, as illustrated in Figure 3(a). More details about motion-aware dynamics can be found in Appendix B.

4.3 METIS Model

As shown in Figure 3(c), our proposed METIS model is built upon a VLM, where the parameters are initialized from the Prismatic-7B [16]. It incorporates a hybrid vision encoder that integrates SigLIP [64] and DINOv2 [33], capturing both global semantics and fine-grained spatial details. The resulting visual features $f^{SigLIP} \in \mathbb{R}^{N_v \times 1152}$ and $f^{DINO} \in \mathbb{R}^{N_v \times 1024}$, where N_v represents the visual token dimension, are concatenated along the channel dimension. A projection layer is applied to align visual embeddings with the language modality. The 7B LLaMA-2 [46] large language model is adopted as the LLM backbone, which employs a decoder-only Transformer architecture with 32 sequential blocks for auto-regressive language modeling.

Previous work such as OpenVLA [19] and RT-2 [68] map the infrequently used words in LLaMA-2 to action bins uniformly distributed within $[-1, 1]$. While effective for low-dimensional control, they struggle to scale to high-dimensional, continuous actions, often resulting in inefficient and unstable behavior. METIS first extends the LLaMA tokenizer vocabulary with $|C_1| + |C_2|$ special tokens, where $|C_1|$ and $|C_2|$ correspond to the codebook sizes of the visual dynamics tokens and motion dynamics tokens, respectively. Using the dynamics model introduced in the previous section, we discretize each egocentric manipulation sequence into motion-aware action tokens, denoted as D_{vis} and D_{mot}. Each dynamics feature is assigned to its nearest codebook entry, and the resulting discrete index is mapped to a unique special token. This design fully leverages the original VLM architecture, enabling autoregressive supervision for training. By preserving the language priors of the language model while injecting motion information, METIS learns to model the fine-grained dynamics essential for dexterous manipulation. The auto-regressive objective of METIS π is to minimize the sum of next-dynamics negative log-probabilities:

$$\mathcal{L}_{ar} = \mathbb{E}_{o_t, l, a_{d,<i}}[-\sum_{i=1}^{N} log\ \pi_\phi(\hat{a}_{d,i|o_t,l,a_{d,<i}})] \tag{1}$$

where N represents the total length of dynamic tokens, which is 44 in our settings (4 for visual dynamics and 40 for motion dynamics). More details can be found in Appendix.

Action Decoder. The Action Decoder translates motion-aware dynamic tokens into executable low-level actions. It takes as input the dynamics token, visual embeddings, and the current proprioception expressed in the unified space. visual and dynamic features are aggregated through multi-head attention pooling, while proprioceptive inputs are projected into the hidden space via a two-layer MLP. The fused representation is passed through a linear projection head to predict a sequence of actions over one second (32 future steps). The final training objective is $\mathcal{L} = \mathcal{L}_{ar} + \lambda \mathcal{L}_{action}$.

Chain-of-Thought Reasoning for Action. Inspired by chain-of-thought prompting in large language models, METIS decomposes high-level manipulation instructions into shorter subtasks. The subtask is accompanied by fine-grained hand-level descriptions that explicitly guide action prediction (e.g., "grip sugar cubes into the bowl"). To support this reasoning process, we manually annotate our self-collected dataset with subtask-level labels. Each demonstration trajectory is segmented into a sequence of temporally contiguous subtask segments, each accompanied by a natural-language description specifying the manipulated object, the acting hand, and the intended motion.

METIS operates in two distinct, adaptively switched modes: Reasoning Mode and Acting Mode. This dual-mode design allows the model to dynamically balance high-level subtask reasoning with low-level action execution, while avoiding unnecessary reasoning at every timestep and thereby reducing inference latency. Specifically, Mode switching is controlled by a token-gated mechanism. We introduce two special tokens: *beginning of reasoning* ($[BOR]$) and *beginning of dynamics* ($[BOD]$). When a subtask transition is detected, METIS first predicts $[BOR]$ and enters the reasoning mode, during which it auto-regressively performs subtask reasoning. The generated reasoning is incorporated into the language context for subsequent decisions. Conversely, If the model deems reasoning unnecessary, such as continuing the execution of an ongoing subtask, it directly predicts the

Table 1: **Main results of six real-world tasks.** Each experiment is evaluated with 20 trials. SR denotes Success Rate, and PSR denotes Progress Success Rate.

Method	Pick and Place	Close Laptop	Open Drawer	Grasp Two Drinks into Basket		Put Cola into Basket		Open Drawer and Put Bread	
	SR	SR	SR	SR	PSR	SR	PSR	SR	PSR
ACT	35.0%	65.0%	**95.0%**	25.0%	40.0%	50.0%	53.3%	5.0%	5.0%
OpenVLA-OFT	50.0%	80.0%	10.0%	40.0%	57.5%	55.0%	56.7%	0.0%	1.0%
$\pi_{0.5}$	60.0%	85.0%	70.0%	65.0%	72.5%	**75.0%**	**76.7%**	60.0%	65.0%
GR00T N1.5	70.0%	80.0%	80.0%	65.0%	70.0%	70.0%	73.3%	70.0%	72.5%
METIS (Ours)	**85.0%**	**95.0%**	90.0%	**75.0%**	**85.0%**	70.0%	**76.7%**	**75.0%**	**82.5%**

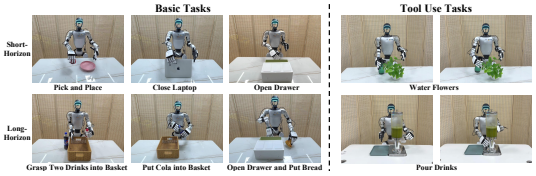


Figure 4: **Visualization of dexterous tasks.**

Table 2: **Performance on tool-use tasks.**

Method	Pour Drinks		Water Flowers	
	SR	PSR	SR	PSR
$\pi_{0.5}$	40.0%	41.7%	15%	18.3%
GR00T N1.5	60.0%	75.0%	50%	53.3%
METIS (Ours)	**80.0%**	**83.3%**	**60%**	**63.3%**

$[BOD]$ token to switch into the acting mode. Following this, METIS outputs motion-aware dynamics tokens which are decoded by the action decoder into low-level actions for execution.

To supervise the gate at each training step, we construct token targets from the annotated subtask boundaries. Inspired by the probabilistic interval supervision of OneTwoVLA [29], we use a symmetric sampling schedule within a transition window of $[-10, +10]$ frames around each boundary τ_k. The probability of assigning $[BOR]$ is

$$p_{\mathrm{BOR}}(t) = 1 - \frac{|t - \tau_k|}{20}, \qquad |t - \tau_k| \leq 10. \tag{2}$$

This probability equals 1.0 at the boundary and decreases linearly to 0.5 at either end of the window. Frames within the window are assigned $[BOR]$ with probability $p_{\mathrm{BOR}}(t)$ and $[BOD]$ otherwise, while frames outside all transition windows are assigned $[BOD]$. Each transition therefore provides up to 21 candidate frames for $[BOR]$ supervision, rather than only a single boundary frame.

5 Experiments

5.1 Experiment Setup

Hardware Platform. For hardware platform, we use a Unitree G1 humanoid robot equipped with a pair of Inspire 6-DoF dexterous hands for fine-grained manipulation. An Intel RealSense D435 camera is mounted on the robot's head to capture egocentric RGB observations.

Self-collected Robot Data. We collect robot demonstrations through human teleoperation with a glove-tracker system [49]. The tracker attached to the operator's wrist captures precise wrist poses, which are converted into the robot arm's joint configurations via inverse kinematics(IK). Simultaneously, a motion-capture glove records the fingertip trajectories, which are mapped to the dexterous hand joint space using an IK-based retargetting algorithm [38].

Tasks. We evaluated METIS on six dexterous manipulation tasks including three short-horizon and three long-horizon tasks, as shown in Figure 4: (1) Pick and Place, (2) Close Laptop, (3) Open Drawer, (4) Grasp Two Drinks into Basket, (5) Put Cola into Basket, (6) Open Drawer and Put Bread. We further evaluate METIS on more fine-grained tool-use tasks (e,g. Pour Drinks and Water Flowers). Each task is collected with 100 high-quality demonstrations and is evaluated with 20 trials by default. We provide more details about these tasks in Appendix D.

Baselines. We compare METIS with four representative baselines: (1) ACT [66], an action chunking transformer that learns low-level visuomotor policy. (2) OpenVLA-OFT [20], an enhanced open-source VLA model via optimized fine-tuning; (3) $\pi_{0.5}$ [15], a VLA flow model for general robot control; (4) GR00T N1.5 [3], an improved VLA model for generalist humanoid robots.

Evaluation Metrics. We use two metrics to evaluate model performance: Success Rate (SR), indicating the entire task is successfully completed, and Progress Rate (PSR), capturing the average completion ratio of subtasks relative to the overall task in long-horizon settings.

Table 3: **Ablation on multi-source pretraining.**

Method	Pick and Place	Put Bread
METIS-NoPretrain	60.0%	35.0%
METIS-HumanPretrain	70.0%	60.0%
METIS-HumanRobotPretrain	80.0%	65.0%
METIS-FullPretrain	**85.0%**	**75.0%**

Table 4: **Ablation on reasoning mechanism.**

Method	Open Drawer and Put Bread	Pour Drinks	Throughput (Hz)
No-reasoning	70.0%	65.0%	28.30
Always-reasoning	**75.0%**	75.0%	17.27
Adaptive-reasoning	**75.0%**	**80.0%**	**23.58**

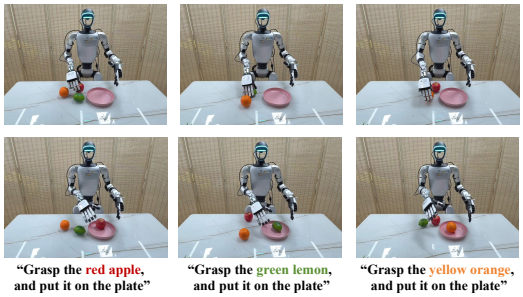


Figure 5: **Instruction following results.** Each task is collected with 100 demonstrations, jointly trained, and evaluated using different instructions.

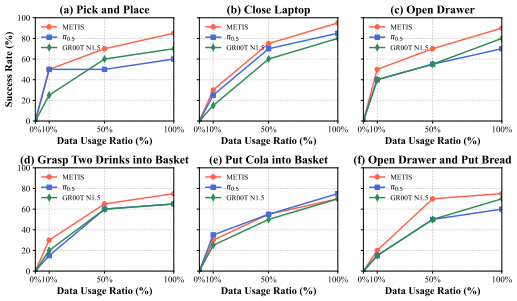


Figure 6: **Experimental results of efficiency.** We train the VLA model with an increasing number of demonstrations.

5.2 Performance in Real-World Experiments

We conduct experiments on real-world dexterous manipulation tasks, as shown in Table 1. Our model, achieves the highest average success rate and consistently outperforms existing state-of-the-art VLA models on most tasks. The specialist model ACT performs well on short-horizon tasks but struggles on long-horizon tasks. $\pi_{0.5}$ underperforms on tasks requiring precise dexterous manipulation, as it has not been pretrained on large-scale dexterous datasets. GR00T N1.5 achieves competitive results through large-scale pretraining on humanoid robot and human data with implicit latent representations. Nevertheless, the absence of explicit reasoning and feedback correction limits its performance on long-horizon tasks. METIS, benefiting from motion-aware dynamics and explicit reasoning mechanism, achieves outstanding performance across both short and long-horizon tasks. Additionally, METIS attains highest PSR across all long-horizon tasks, demonstrating it can reason and act coherently across long-horizon sequences, maintaining stable control and minimizing error accumulation during task execution.

Dexterous Capability. We evaluate model performance on tool-use tasks to better assess fine-grained dexterous manipulation capability. As shown in Table 2, dexterous VLA (e.g., GR00T N1.5 and METIS) consistently outperform the gripper-based VLA method ($\pi_{0.5}$). Among them, METIS achieves the best overall performance, benefiting from large-scale pretraining on human tool-use demonstrations and the extraction of compact dexterous motion representations.

Instruction Following. We further evaluate the instruction following capability of METIS. In this setup, three fruits of different colors (apples, orange, and lemon) are placed on the table. As shown in Figure 5, when given different language instructions (e.g., "place the red apple on the plate"), METIS can identify the target fruit and execute the corresponding grasping action. These results demonstrate the ability of METIS to connect object-level language understanding with dexterous manipulation in scenes containing multiple candidate objects.

5.3 Efficiency

Sample Efficiency. We evaluate the sample efficiency of METIS by training it with varying amounts of data on downstream dexterous tasks, as shown in Figure 6. METIS exhibits excellent efficiency, achieving superior performance even with limited training data. Notably, when fine-tuned with only 10% data, METIS attains a 50% success rate on the Pick and Place and Open Drawer task. This results strongly indicate that pretraining on diverse multi-source egocentric data with unified action space provides valuable priors, including spatial reasoning, visual-hand coordination, and motion dynamics. Such priors form consistent visuomotor representations and facilitate rapid transfer to new downstream tasks.

Table 5: **Ablation on motion-aware dynamics (success rate vs. inference efficiency).**

Method	Token Count	Throughput (Hz)	Pick and Place	Open Drawer and Put Bread	Pour Drinks
METIS w/ Naive Tokenizer	1440	0.93	15.0%	10.0%	0.0%
METIS w/ Fast Tokenizer	138	9.87	50.0%	25.0%	15.0%
METIS w/ Motion-aware Dynamics	**44**	**28.30**	**60.0%**	**35.0%**	**30.0%**

Table 6: **Generalization results of OOD scenarios.**



Method	Background	Lighting	Object	Cluttered
$\pi_{0.5}$	50.0%	**70.0%**	65.0%	55.0%
GR00T N1.5	65.0%	65.0%	65.0%	60.0%
METIS (Ours)	**70.0%**	65.0%	**70.0%**	65.0%

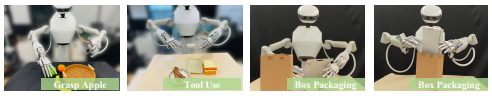


Figure 7: **Cross-embodiment generalization.** METIS demonstrates transferability to 22-DoF dexterous hands, achieving stable performance on three dexterous tasks.

Inference Efficiency. Motion-aware dynamics provide a compact representation of hand motion, significantly reducing the number of tokens generated by the VLM and thereby improving inference efficiency. We evaluate METIS against other tokenizers (Naive and Fast [35]) under identical settings on a single NVIDIA H100 GPU. As shown in Table 5, when predicting an action chunk of length 32, METIS with motion-aware dynamics reduces the token count from 1440 and 138 to only 44 tokens, achieving a throughput of 28.30 Hz (action steps per second). This demonstrates the effectiveness of motion-aware dynamics for efficient autoregressive generation.

5.4 Generalization

Besides the remarkable effectiveness and efficiency, METIS also showcases excellent generalization capabilities. We evaluate our model in four *out-of-distribution*(OOD) scenarios, including (1) **Unseen background**, we cover the tabletop with a colorful patterned tablecloth to introduce significant visual distractions and background textures unseen during training. (2) **Unseen lighting condition**, we illuminate the scene with color-changing and flickering lights to simulate dynamic and diverse lighting situations. (3) **Unseen object**, we replace the original scone bread with a visually distinct croissant to assess object-level generalization. (4) **Cluttered environment**, we randomly place distractor objects such as a plate and an apple near the drawer, increasing visual and spatial complexity. Taking the Open Drawer and Put Bread task as an example, we compare and report the success rates of METIS and GR00T N1.5 across all four out-of-distribution scenarios, as shown in Table 6. The results demonstrate that METIS effectively adapts to various distributional shifts, maintaining stable visuomotor grounding and task execution even under significant visual or physical variations.

Cross-Embodiment Generalization. METIS also generalizes effectively to higher-DoF dexterous hands, demonstrating strong adaptability across embodiments. We evaluate METIS on the Sharpa Beta embodiment equipped with a pair of 22-DoF SharpaWave Dexterous hands, as shown in Figure 7. The model achieves 85.0% success rate on the Grasp Apple into Basket task and 70.0% on the Tool Use task, respectively. We further evaluate METIS on a long-horizon and highly dexterous *Box Packaging* task, which requires precise grasping, coordinated bimanual manipulation, folding of packaging materials, and accurate placement into a box. Despite the increased task complexity, METIS achieves a 15% success rate, while the baseline GR00T N1.5 fails to complete the task in all 20 trials. As METIS predicts fingertip trajectories rather than direct joint angles, the policy is naturally transferable and remains unaffected by variations in hand kinematics.

5.5 Ablations

We have demonstrated the effectiveness of METIS in terms of overall performance, efficiency, and generalization. However, the contribution of its core components remains unexplored. In this section, we perform ablation studies focusing on three aspects: (1) the effect of multi-source egocentric pretraining, (2) the impact of the motion-aware dynamics, (3) the effect of reasoning mechanism, and (4) the visual dynamics token budget.

Multi-source Egocentric Pretraining. To investigate the role of multi-source egocentric pretraining, we evaluate four variants of the model on downstream tasks: (a) METIS without pretraining, (b) METIS pretrained only on the open-source human data, (c) METIS pretrained on the human and robot

Table 7: **Ablations of visual dynamics.** (a) Success rates with different dynamics components. (b) Effect of the visual-token budget on codebook usage and visual reconstruction error.

(a) Dynamics components

Method	Open Drawer and Put Bread	Pour Drinks
Visual-only	45.0%	50.0%
Motion-only	60.0%	55.0%
Motion-aware	**75.0%**	**80.0%**

(b) Visual-token budget

Visual Tokens	Codebook Usage	Reconstruction Error ↓
2	0.80	$1.2e^{-2}$
4	1.00	$6.0e^{-4}$
6	0.95	$3.9e^{-4}$
8	0.95	$2.4e^{-4}$

data, excluding our self-collected dataset, (d) METIS pretrained on EgoAtlas dataset. As presented in Table 3, pretraining on multi-source egocentric datasets improves downstream performance, indicating that training on diverse visual and action distributions enables the model to learn more robust visuomotor priors and generalize effectively across a wide range of tasks. Notably, although the no-pretraining VLA achieves a certain level of success rate on some tasks, it exhibits significant loss fluctuations during the post-training stage and unstable joint jitters during real-world deployment.

Motion-aware Dynamics. We conduct ablation experiments on the motion-aware dynamics. During post-training, we compare motion-aware dynamics with naive and FAST tokenizers under the same setting without pretraining. As shown in Table 5, METIS with motion-aware dynamics consistently achieves better performance, especially on long-horizon and fine-grained tasks, highlighting its advantage in modeling motion details.

To assess the contribution of each component, we compare visual-only, motion-only, and full motion-aware dynamics on two manipulation tasks. As shown in Table 7(a), the full formulation achieves success rates of 75.0% on Open Drawer and Put Bread and 80.0% on Pour Drinks, outperforming both visual-only dynamics (45.0% and 50.0%) and motion-only dynamics (60.0% and 55.0%). These results support the complementary roles of the two components: visual dynamics models scene evolution and object-state transitions, while motion dynamics directly supervises wrist and fingertip kinematics. We provide more quantitative results about codebook usage, motion reconstruction and visualizations of motion-aware tokens, please refer to Appendix G.

Reasoning Mechanism. We conduct a quantitative evaluation of the reasoning mechanism by comparing three variants: no-reasoning, always-reasoning, and our adaptive-reasoning. The success rate and inference latency are reported in Table 4. For long-horizon dexterous tasks, reasoning improves planning capability and overall performance. While always-reasoning achieves competitive success rates, it incurs higher latency due to generating both reasoning and actions at every step, which can be detrimental in highly dynamic settings.

Visual Dynamics Token Budget. We further investigate the visual-token budget through codebook usage and visual reconstruction error. As reported in Table 7(b), increasing the budget from four to eight tokens reduces reconstruction error from 6.0 to 2.4, while codebook usage decreases slightly from 1.0 to 0.95. Although additional tokens improve reconstruction fidelity within this range, they also increase the sequence length and associated decoding cost. We adopt four visual dynamics tokens as a practical trade-off between reconstruction fidelity and inference efficiency. Qualitative visualizations in Appendix Figure 12 further show that similar motion patterns, such as downward or rightward movements, can be represented by the same codewords across different embodiments and scene appearances. This suggests that the learned visual dynamics tokens capture motion semantics that are partially invariant to embodiment and visual context.

6 Conclusion

In this paper, we present METIS, a vision-language-action model for dexterous manipulation pretrained on multi-source egocentric dataset, integrating reasoning and acting within a unified framework. To support large-scale pretraining, we construct EgoAtlas, a comprehensive multi-source egocentric dataset that aligns human and robotic data under a consistent action space. By extracting motion-aware dynamics from manipulation trajectories, METIS acquires compact and expressive representations of hand motion, enabling precise and coordinated action generation. Experimental results demonstrate that METIS achieves strong performance across diverse dexterous manipulation tasks and exhibits robust generalization to out-of-distribution scenarios.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (62476011) and Beijing Natural Science Foundation (L252060). We thank BAAI for providing hardware and computational resources for our experiments. We are also grateful to Lei Zhang and Shaoxuan Xie for their assistance with data collection.

References

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

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

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

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

[5] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025.

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

[7] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, Steven Palma, Pepijn Kooijmans, Michel Aractingi, Mustafa Shukor, Dana Aubakirova, Martino Russi, Francesco Capuano, Caroline Pascal, Jade Choghari, Jess Moss, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. `https://github.com/huggingface/lerobot`, 2024.

[8] Zicong Fan, Omid Taheri, Dimitrios Tzionas, Muhammed Kocabas, Manuel Kaufmann, Michael J Black, and Otmar Hilliges. Arctic: A dataset for dexterous bimanual hand-object manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12943–12954, 2023.

[9] Yankai Fu, Qiuxuan Feng, Ning Chen, Zichen Zhou, Mengzhen Liu, Mingdong Wu, Tianxing Chen, Shanyu Rong, Jiaming Liu, Hao Dong, et al. Cordvip: Correspondence-based visuomotor policy for dexterous manipulation in real-world. *arXiv preprint arXiv:2502.08449*, 2025.

[10] Koffivi Fidèle Gbagbe, Miguel Altamirano Cabrera, Ali Alabbas, Oussama Alyunes, Artem Lykov, and Dzmitry Tsetserukou. Bi-vla: Vision-language-action model-based system for bimanual robotic dexterous manipulations. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 2864–2869. IEEE, 2024.

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[12] Ryan Hoque, Peide Huang, David J Yoon, Mouli Sivapurapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

[13] Chengkai Hou, Kun Wu, Jiaming Liu, Zhengping Che, Di Wu, Fei Liao, Guangrun Li, Jingyang He, Qiuxuan Feng, Zhao Jin, et al. Robomind 2.0: A multimodal, bimanual mobile manipulation dataset for generalizable embodied intelligence. *arXiv preprint arXiv:2512.24653*, 2025.

[14] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

[15] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

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

[17] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[18] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Goal-conditioned dual-action imitation learning for dexterous dual-arm robot manipulation. *IEEE Transactions on Robotics*, 40: 2287–2305, 2024.

[19] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[20] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[21] Yuxuan Kuang, Junjie Ye, Haoran Geng, Jiageng Mao, Congyue Deng, Leonidas Guibas, He Wang, and Yue Wang. Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation. *arXiv preprint arXiv:2407.04689*, 2024.

[22] Vikash Kumar, Yuval Tassa, Tom Erez, and Emanuel Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6808–6815. IEEE, 2014.

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

[24] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks for action segmentation and detection. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[25] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization, 2022. URL `https://arxiv.org/abs/2203.01941`.

[26] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023.

[27] Yunshuang Li, Yiyang Ling, Gaurav S Sukhatme, and Daniel Seita. Learning geometry-aware nonprehensile pushing and pulling with dexterous hands. *arXiv preprint arXiv:2509.18455*, 2025.

[28] Davide Liconti, Yasunori Toshimitsu, and Robert Katzschmann. Leveraging pretrained latent representations for few-shot imitation learning on a dexterous robotic hand. *arXiv preprint arXiv:2404.16483*, 2024.

[29] Fanqi Lin, Ruiqian Nai, Yingdong Hu, Jiacheng You, Junming Zhao, and Yang Gao. Onetwovla: A unified vision-language-action model with adaptive reasoning. *arXiv preprint arXiv:2505.11917*, 2025.

[30] Jiaming Liu, Hao Chen, Pengju An, Zhuoyang Liu, Renrui Zhang, Chenyang Gu, Xiaoqi Li, Ziyu Guo, Sixiang Chen, Mengzhen Liu, et al. Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model. *arXiv preprint arXiv:2503.10631*, 2025.

[31] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21013–21022, 2022.

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

[33] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[34] 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. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[35] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[36] Johannes Pitz, Lennart Röstel, Leon Sievers, and Berthold Bäuml. Dextrous tactile in-hand manipulation using a modular reinforcement learning architecture. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1852–1858. IEEE, 2023.

[37] Haozhi Qi, Ashish Kumar, Roberto Calandra, Yi Ma, and Jitendra Malik. In-hand object rotation via rapid motor adaptation. In *Conference on Robot Learning*, pages 1722–1732. PMLR, 2023.

[38] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. In *Robotics: Science and Systems*, 2023.

[39] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J Yoon, Ryan Hoque, Lars Paulsen, et al. Humanoid policy ~ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

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

[41] Kenneth Shaw, Yulong Li, Jiahui Yang, Mohan Kumar Srirama, Ray Liu, Haoyu Xiong, Russell Mendonca, and Deepak Pathak. Bimanual dexterity for complex tasks. *arXiv preprint arXiv:2411.13677*, 2024.

[42] BAAI RoboBrain Team, Mingyu Cao, Huajie Tan, Yuheng Ji, Xiansheng Chen, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, et al. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025.

[43] Fourier ActionNet Team and Yao Mu. Actionnet: A dataset for dexterous bimanual manipulation. 2025.

[44] Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[45] 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.

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

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

[48] Weikang Wan, Haoran Geng, Yun Liu, Zikang Shan, Yaodong Yang, Li Yi, and He Wang. Unidexgrasp++: Improving dexterous grasping policy learning via geometry-aware curriculum and iterative generalist-specialist learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3891–3902, 2023.

[49] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

[50] Huiyu Wang, Mitesh Kumar Singh, and Lorenzo Torresani. Ego-only: Egocentric action detection without exocentric transferring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5250–5261, 2023.

[51] Ruicheng Wang, Jialiang Zhang, Jiayi Chen, Yinzhen Xu, Puhao Li, Tengyu Liu, and He Wang. Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation. *arXiv preprint arXiv:2210.02697*, 2022.

[52] Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, et al. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20270–20281, 2023.

[53] Junjie Wen, Yichen Zhu, Jinming Li, Zhibin Tang, Chaomin Shen, and Feifei Feng. Dexvla: Vision-language model with plug-in diffusion expert for general robot control. *arXiv preprint arXiv:2502.05855*, 2025.

[54] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinuo Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024.

[55] Yinzhen Xu, Weikang Wan, Jialiang Zhang, Haoran Liu, Zikang Shan, Hao Shen, Ruicheng Wang, Haoran Geng, Yijia Weng, Jiayi Chen, et al. Unidexgrasp: Universal robotic dexterous grasping via learning diverse proposal generation and goal-conditioned policy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4737–4746, 2023.

[56] Fan Yang, Thomas Power, Sergio Aguilera Marinovic, Soshi Iba, Rana Soltani Zarrin, and Dmitry Berenson. Multi-finger manipulation via trajectory optimization with differentiable rolling and geometric constraints. *IEEE Robotics and Automation Letters*, 2025.

[57] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, et al. Magma: A foundation model for multimodal ai agents. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14203–14214, 2025.

[58] Lixin Yang, Kailin Li, Xinyu Zhan, Fei Wu, Anran Xu, Liu Liu, and Cewu Lu. Oakink: A large-scale knowledge repository for understanding hand-object interaction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20953–20962, 2022.

[59] 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.

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

[61] Zhao-Heng Yin, Changhao Wang, Luis Pineda, Francois Hogan, Krishna Bodduluri, Akash Sharma, Patrick Lancaster, Ishita Prasad, Mrinal Kalakrishnan, Jitendra Malik, et al. Dexteritygen: Foundation controller for unprecedented dexterity. *arXiv preprint arXiv:2502.04307*, 2025.

[62] Chengbo Yuan, Chuan Wen, Tong Zhang, and Yang Gao. General flow as foundation affordance for scalable robot learning. *arXiv preprint arXiv:2401.11439*, 2024.

[63] 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.

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

[65] Anran Zhang, Hanzhi Chen, Yannick Burkhardt, Yao Zhong, Johannes Betz, Helen Oleynikova, and Stefan Leutenegger. Actron3d: Learning actionable neural functions from videos for transferable robotic manipulation. *arXiv preprint arXiv:2510.12971*, 2025.

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

[67] 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.

[68] 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.

Appendix

A EgoAtlas Dataset

A.1 Data Calibration

We perform a comprehensive calibration procedure using three VIVE trackers within our motion-capture setup. Two trackers are mounted on the wrists to provide the 6-DoF global wrist poses in the world coordinate frame. Another tracker is rigidly attached to the head-mounted egocentric camera via a custom-designed mount. Given this fixed transformation, we obtain the extrinsic parameters of the egocentric camera with respect to the global coordinate frame. By combining these measurements, we transform the wrist poses from the world coordinate frame into the egocentric camera coordinate frame, achieving consistent spatial correspondence between hand motion and egocentric observations. For the finger poses, the Manus Quantum gloves directly provide 3D fingertip positions relative to the wrist frame.

A.2 Data Collection Interface

To ensure data quality, we built a data collection interface that visualizes the video stream, wrist pose and fingertip keypoints in real time. This platform allows the operator to monitor the tracking status of all sensors. When the operator identifies tracking failures or degraded quality, the recording session can be manually terminated anytime. The entire system operates at 20Hz, balancing motion fidelity with reliable multi-sensor synchronization.

A.3 Coordinate System Definition

The egocentric camera coordinate system follows the conventional computer vision convention: the origin is located at the optical center of the camera, the z-axis points forward along the optical axis (outward from the camera), the x-axis points to the right of the camera, and the y-axis points downward in the image plane.

When computing the fingertip positions of the dexterous hand, we employ forward kinematics (FK) with the root joint of the hand as the origin. However, the base of the robotic hand is typically not perfectly aligned with the wrist pose. To resolve this, we apply a fixed transformation between the wrist and the hand base to ensure geometric consistency. For clarity, we define wrist coordinate systems as follows:

- **Left Hand**: the **x-axis** points toward the fingertips, the **y-axis** from the palm to the back of the hand, and the **z-axis** from the little finger toward the thumb.
- **Right Hand**: the **x-axis** points along the forearm direction, the **y-axis** from the palm to the back of the hand, and the **z-axis** from the little finger toward the thumb.

We provide a visualization of the hand motion in our dataset (see Figure 8).

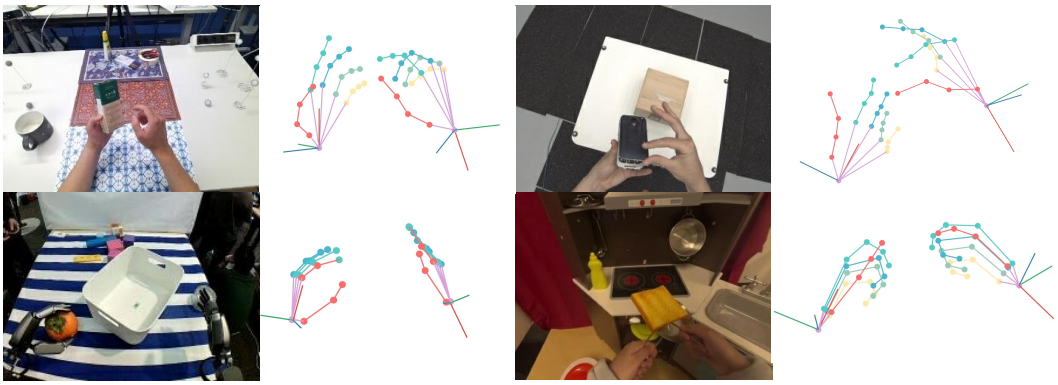


Figure 8: **Visualization of hand motion.** We use 3D keypoints to represent hand motion. To ensure consistency across embodiments, the wrist frames of human and robot hands are strictly aligned.

A.4 Unifying Action Space

In this section, we describe how to map hand motion information from different configurations and embodiments into a unified action space. Existing action representations for hand motion can generally be categorized into three classes.

- **MANO-based human hand representations.**
- **Keypoint-based hand representations.**
- **Robot joint-space representations.**

MANO-based representations model human hand motion using a low-dimensional parametric hand model, where pose and shape are encoded by a set of continuous parameters. Such representations are typically collected via vision-based motion capture systems [8, 23, 52, 58, 31], including multi-view RGB cameras or RGB-D sensors, which estimate hand pose by fitting the MANO model to visual observations. During data processing, the MANO pose parameters are first passed through the parametric hand model to recover 3D joint locations (MANO FK). A subset of anatomically meaningful position, corresponding to the fingertips of each finger, is then selected to represent fine-grained hand motion. To decouple global hand motion from local finger configurations, all fingertip positions are expressed in a wrist-centric coordinate frame by removing the root rotation and translation.

In parallel, the wrist pose is extracted from the MANO root parameters and represented as a 6D rigid transformation. Using known camera extrinsics, the wrist pose is further transformed from the world coordinate frame into the egocentric camera frame. The resulting wrist pose and fingertip positions jointly form an explicit, viewpoint-consistent spatial representation, which serves as a unified state-action representation for policy learning.

Keypoint-based hand representations using explicit 3D joint positions, typically acquired through VR systems [12] or motion-capture gloves. In these settings, egocentric visual observations are typically provided by a head-mounted VR device or an external camera, while head and wrist poses are obtained through VR systems or dedicated motion trackers. During data processing, hand and finger joint poses are initially provided in a global or world coordinate frame. Using the calibrated camera extrinsics, all joint poses are first transformed into the egocentric camera frame, ensuring alignment with the visual observations. A subset of finger keypoints corresponding to fingertip endpoints is selected to capture fine-grained hand articulation, and these points are expressed in wrist-local coordinates by removing the wrist translation and rotation.

Robot joint-space representations. Robotic arms and hands usually represent actions in joint space [43], where each action corresponds to target joint angles. Given the robot kinematic model defined by its URDF, forward kinematics can be applied to recover the end-effector pose relative to the robot base frame. Similarly, forward kinematics over the hand kinematic chain enables the computation of fingertip positions expressed in the wrist coordinate frame. To ensure consistency with egocentric perception, the end-effector pose is further transformed into an egocentric coordinate frame using the calibrated extrinsic parameters of the head-mounted camera.

A.5 Subtask-Level Annotation

To enable fine-grained understanding of long-horizon manipulation behaviors, we annotate each demonstration video with subtask-level temporal labels. Each video is decomposed into a sequence of semantically meaningful atomic actions, where each subtask corresponds to a continuous temporal segment associated with a natural language description.

Annotators manually partition the video into a set of non-overlapping temporal intervals. Each interval is defined by a start frame and an end frame, forming a contiguous segment that captures a coherent manipulation primitive. The segmentation strictly preserves the temporal order of the task execution and covers the entire video without gaps.

For each temporal segment, annotators assign a concise description that specifies:

- The action primitive (e.g., grasp, move, pour, place).
- The active effector (left hand or right hand).

- The target object when applicable.

Each episode concludes with an explicit terminal label (e.g., End) to mark task completion. This provides a clear boundary for long-horizon reasoning and facilitates sequence-level modeling. Figure 9 visualizes the subtask-level annotations of representative data samples.



Figure 9: **Visualization of subtask-level annotations.**

A.6 Hand IK Retargetting

We use Pinocchio to parse the hand URDF and perform kinematic computations, including forward kinematics and fingertip Jacobian evaluation, which are then used in our iterative IK solver. Given the hand URDF and predefined master–mimic relations, the retargeting pipeline is deterministic and geometry-based, and consistently applicable across embodiments (validated by cross-embodiment results). The IK solver may introduce approximation errors, especially for high-DoF dexterous hands where redundancy, joint limits, and embodiment-specific kinematic constraints can lead to multiple feasible solutions or no exact solution. To mitigate this issue, our solver is initialized from the current hand state and optimized with joint-limit and smoothness constraints, which biases the solution toward physically feasible and temporally consistent motions. When an exact fingertip target is unreachable, the solver returns the closest stable solution rather than an arbitrary pose, avoiding abrupt joint changes and reducing self-collisions. Any residual projection error is further absorbed during embodiment-specific fine-tuning, where the policy adapts to the reachable workspace and kinematic constraints of the target hand.

A.7 Data Sources and Statistics

EgoAtlas integrates data from four major sources, covering both human and robotic domains with diverse sensing modalities: **(1) Vision-based motion capture datasets** [8, 23, 52, 58, 31], which use multi-camera optical systems to capture precise 3D hand annotations and object interactions. However, they are typically confined to small tabletop environments with limited diversity. **(2) VR-based human datasets** [12], which utilize on-device SLAM and calibrated cameras to capture hand and wrist pose. VR-based setups exhibit fewer scene constraints and allow more flexible

data collection across different environments. **(3) Teleoperated robot data** [43, 39], which involve human operators remotely controlling dexterous robotic hands to execute manipulation tasks. **(4) Self-collected enhanced motion dataset**, as described in Section 3.1, we collected 10K high-fidelity human hand motion trajectories. This enhanced dataset brings three main benefits: (1) it is robust to occlusions and visual ambiguities; (2) it introduces appearance variation via wearable gloves; (3) it provides rich semantic annotations. As shown in Figure 11 (a), the self-collected dataset covers diverse task categories, including single-object and multi-object pick-and-place, assembly, tool use, and articulated manipulation. 57.2% of trajectories are single-arm and 42.8% are bimanual. The distribution is balanced, with higher proportions of tool use (25%) and precise manipulation (19%), reflecting an emphasis on complex dexterous skills.

The composition of EgoAtlas dataset is summarized in Table 8, which includes 8 sources with detailed annotations of hand motions. In total, EgoAtlas contains 343K trajectories and 89.72M image–action pairs. Weighted sampling is applied to maintain a balanced source distribution.

B Motion-aware dynamics tokens

To empower the Vision-Language Model (VLM) with enhanced motion comprehension during action execution, we introduce a **Motion-aware Dynamics Tokens** framework as a supervisory mechanism. This framework is structured to decompose dynamic information into two complementary components: **Vision Dynamics** and **Motion Dynamics**.

The **Vision Dynamics** component is designed to learn a compact, discrete representation of the visual change between consecutive frames. This process begins with an encoder $\text{Enc}_V(\cdot)$ based on a self-attention mechanism. The encoder's input is a concatenation of three elements: the visual features I_t, I_{t+k} from the image sequence, the corresponding motion features $P_{t,t+1,\ldots,t+k}$, and a set of randomly initialized, learnable parameters $\tilde{D}_{vis}$:

$$\hat{D}_{vis} = \text{Enc}_V(I_t, I_{t+k}, P_{t,t+1,\ldots,t+k}, \tilde{D}_{vis})$$

These parameters are then discretized using a VQ-VAE objective, resulting in a final representation composed of $V = 4$ tokens selected from a candidate codebook of size $|C_v| = 16$

$$D_{vis} = \mathbf{VQ}(\hat{D}_{vis})$$

This architecture allows the visual information to be deeply infused with motion characteristics. Subsequently, a decoder $\text{Dec}_V(\cdot)$ takes the first frame I_t along with the refined parameters D_{vis} and is tasked with reconstructing the final frame I_{t+k}:

$$\hat{I}_{t+k} = \text{Dec}_V(I_t, D_{vis})$$

This auto-encoding objective forces D_{vis} to encapsulate the essential visual transformation between the two frames.

Conversely, the **Motion Dynamics** component aims to capture more complex and granular motion patterns directly from raw action data. We employ a two-layer residual quantization (RQ) architecture for this purpose. To preserve the most authentic motion characteristics, the original action information

Table 8: **Statistics of EgoAtlas.** We use in-the-wild to denote data collected in diverse, unconstrained real-world scenes.

Method	Trajs	Frames	Hours	Pose	Subtask	Human	Robot	In-the-wild
ARCTIC	296	214.5K	2.0	✓	✗	100%	0%	✗
H2O	109	65.3K	0.6	✓	✓	100%	0%	✗
HoloAssist	100	777.3K	7.3	✓	✓	100%	0%	✗
Oakink	134	146K	1.4	✓	✓	100%	0%	✗
EgoDex	314.8K	77.9M	721.3	✓	✗	100%	0%	✓
PH2D	1.8K	416.5K	3.9	✓	✗	66.1%	33.9%	✓
ActionNet	15.7K	7.4M	68.4	✓	✗	0%	100%	✓
Ours	10K	2.8M	38.9	✓	✓	100%	0%	✓

$M_{t,t+1,\ldots,t+k}$ is first processed through a Pose Network PoseNet$(\cdot)$ and then fed directly into the RQ network RQ$(\cdot)$:

$$P_{t,t+1,\ldots,t+k} = \text{PoseNet}(M_{t,t+1,\ldots,t+k})$$
$$D_{mot} = \mathbf{RQ}(P_{t,t+1,\ldots,t+k})$$

This architecture selects a total of $R = 40$ tokens from a larger, shared codebook of size $|C_m| = 512$, enabling a hierarchical and fine-grained representation of motion. The motion dynamics tokens D_{mot} are then decoded using a Temporal Convolutional Network (TCN) to reconstruct the original motion sequence, serving as the supervision signal to ensure the quantized representation preserves essential temporal dynamics.

$$\hat{M}_{t,t+1,\ldots,t+k} = \text{TCN}(D_{mot})$$

A key design principle is uniformity: all codebooks maintain a unified feature dimension $d = 128$. This ensures consistent representation for downstream processing while simultaneously enabling each component to effectively capture the hierarchical spatiotemporal patterns crucial for advanced action-aware modeling.

C CoT-Style Reasoning and Acting

METIS operates in a closed-loop manner with distinct input–output flows depending on the active mode (i.e., Reasoning Mode and Acting Mode).

- **Reasoning Mode:** METIS takes as input: (1) the task language instruction, (2) the current egocentric observation, (3) the latest reasoning. To explicitly signal the transition into reasoning mode, the model first emits a dedicated special token, denoted as $[BOR]$ (*Beginning of Reasoning*). Following this, METIS autoregressively generates a subtask-level description (e.g., use the right hand to pour the juice from the tall blue mug into the bowl.) The generated reasoning tokens do not directly interact with the environment, they are appended to the model's context and maintained as the latest reasoning.
- **Acting Mode:** METIS receives (1) the task language instruction, (2) the current egocentric observation, (3) the latest reasoning, (4) the current unified proprioceptive state. Once triggered by the $[BOD]$ (*Beginning of Dynamics*), VLM autoregressively generates a sequence of motion-aware dynamics tokens, which consist of vision dynamics and motion dynamics. The generated dynamics tokens are subsequently interpreted by an action expert, which maps them into low-level executable control commands.

Reasoning Mode and acting mode are realized within a single unified autoregressive framework, where a *token-gated* mechanism determines the active generation stream (i.e., reasoning tokens or acting tokens). This design enables METIS to seamlessly interleave planning and control within a single model, while preserving a consistent input representation and a unified training objective.

D Task Details

Pick and Place. In this task, the robot performs a complete pick-and-place operation involving a common household object—an apple. The robot first identifies the position of the apple on the tabletop through egocentric visual observation and moves its hand toward the target. It then adjusts its wrist orientation and finger configuration to achieve a stable grasp. After successfully lifting the apple, the robot transports it to the target area and gently places it into a plate. This task assesses the model's capability to perceive and manipulate objects accurately, as well as its ability to coordinate visual perception and fine-grained hand control in a sequential manipulation process. Success is achieved if the apple is successfully placed into the plate.

Close Laptop. In this task, the robot is required to close a partially open laptop. The robot places four fingers along the back edge of the laptop cover and applies a downward motion by rotating the wrist, allowing the lid to close smoothly. This task evaluates the model's ability to control contact-rich interactions and perform coordinated multi-finger and wrist motions for precise object manipulation. Success is achieved if the laptop is fully closed.

Open Drawer. In this task, the robot is required to open a partially closed drawer. The robot first bends its four fingers and positions the hand so that the fingertips can reach into the handle of the

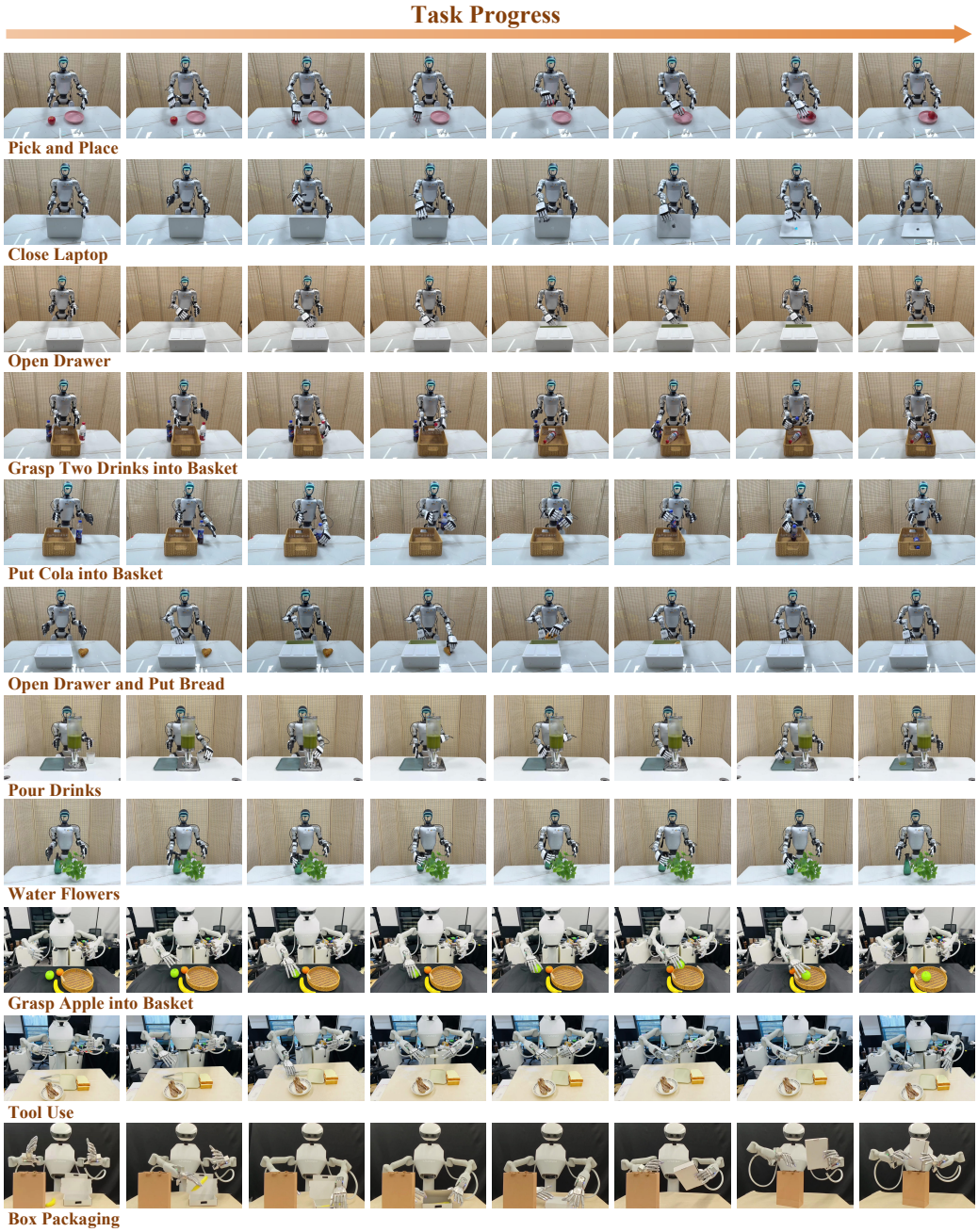


Figure 10: **Visualization of task progress across embodiments.**

drawer. It then moves the wrist backward while maintaining a stable grip on the handle, gradually pulling the drawer open. This task requires precise hand movements guided by visual feedback and involves contact-rich manipulation that demands fine coordination between finger articulation and wrist motion. Success is achieved if the drawer is pulled out to a fully open position.

Grasp Two Drinks into Basket. In this bimanual manipulation task, the robot is required to sequentially grasp and place two bottled drinks into a basket. The robot first raises its left hand, reaches toward the bottle, and performs a stable grasp before lifting it and placing it into the basket. It then raises the right hand to repeat the same motion for the second bottle, ensuring symmetric and coordinated control between both hands. After both bottles are successfully placed, the robot closes both hands into fists and pushes the basket forward to complete the task. This task evaluates

the model's ability to perform coordinated bimanual manipulation and to plan sequential actions that involve precise grasping and contact-rich object placement. Success is achieved if both bottles are correctly placed into the basket and the basket is pushed forward.

Put Cola into Basket. This task requires precise bimanual cooperation to manipulate and place a bottle of cola into a basket. The robot first moves its left hand toward the cola bottle and grasps it from the lower side, lifting it to chest height. The right hand then approaches slowly and grasps the upper part of the bottle, ensuring a stable handover between both hands. After securing the bottle, the robot releases the left hand while the right hand moves toward the basket and places the bottle inside. This task evaluates the model's ability to perform coordinated bimanual-hand manipulation, including object handover, grasp stability, and visually guided placement. Success is achieved if the cola bottle is stably placed inside the basket.

Open drawer and put bread. This long-horizon task requires the robot to perform a sequence of coordinated actions involving both hands. The robot first bends the fingers of its right hand and moves the wrist forward so that the fingertips can reach into the drawer handle. It then pulls the drawer open smoothly through a controlled backward motion. Next, the robot moves its left hand to grasp a piece of bread randomly placed on the tabletop, lifts it, and positions it above the open drawer. The bread is then released into the drawer, after which the robot uses its right hand to close the drawer. This task challenges the model's ability to perform long-horizon visuomotor control, requiring precise spatial reasoning, bimanual coordination, and contact-rich manipulation across multiple sub-actions. Success is achieved if the bread is successfully placed inside the drawer and the drawer is fully closed.

Pour Drinks. In this task, the robot first grasps a cup from one side using its left hand and positions it beneath the beverage dispenser's outlet. The robot then actuates the dispenser switch with its left hand to pour the drink into the cup. The robot then uses its right hand to grasp the filled cup and place it onto the designated area on the right-side table. This task involves tool use, manipulation of transparent objects, and coordinated bimanual actions, placing high demands on precise control and fine-grained manipulation. Success is achieved when the filled cup is stably placed at the specified target location.

Water Flowers. In this task, the robot is required to perform fine-grained tool use to water flowers. The robot first grasps a watering sprayer from the table using a functionally appropriate grip, with the thumb supporting the handle while the index and middle fingers are positioned on the trigger. It then aligns the sprayer toward the flower pot and actuates the trigger by pressing the index and middle fingers, causing water to be sprayed onto the plant. This task evaluates the model's ability to execute precise finger-level control, functional grasping , and coordinated tool use. Success is achieved when water is correctly dispensed onto the plant through a stable and functionally correct grasp of the watering tool.

Grasp Apple into Basket. In this task, the robot is required to accurately identify the position of an apple among three objects (orange, apple, and banana) present simultaneously in its field of view, then precisely grasp the apple using four fingers. After successfully securing the apple, the robot transports it above a basket and gently places it inside. This task evaluates the model's capability for object perception and manipulation in cluttered environments, as well as its ability to execute targeted grasping and placement operations amid visual distractions. Success is achieved when the apple is securely deposited into the basket.

Tool Use. In this task, the robot is required to reposition a bread clip into an optimal grasping posture for functional use. The robot first precisely pinches the bread clip with its right hand, then transfers the tool's end to its left hand for stabilization. Finally, the right hand regrasps the bread clip at an operational angle suitable for precise manipulation. This task evaluates the model's capability for bimanual coordination and its ability to dynamically adjust grasping strategies during tool transfer. Success is achieved when the tool is securely held in a functionally optimal posture through seamless handover between both hands.

Box Packaging. In this task, the robot first grasps a toy banana and places it into a cardboard box. It then folds and assembles the box structure through a series of precise, coordinated bimanual actions. During the process, the robot must handle deformable and semi-rigid materials, continuously adjusting contact points and grasping strategies to maintain structural stability. After the box is fully formed, the robot accurately places the packaged box into a paper bag. This task evaluates the model's

capability for long-term reasoning, multi-stage task decomposition, and dexterous manipulation with high-DoF hands. Success is achieved when the box is fully assembled and correctly packaged through a coherent sequence of grasps, folds, and hand transitions.

More visualization of dexterous manipulation tasks can be found in Figure 10.

E Policy Implementation Details

E.1 Pretraining Details

We use the EgoAtlas dataset to pretrain METIS, which integrates eight heterogeneous data sources covering both human and robot egocentric manipulation trajectories. To ensuring training efficiency and maintain a balanced distribution across data domains, we apply sampling to construct the final training mixture. The detailed dataset composition and mixture weights are provided in table 9.

Table 9: The dataset name and sampling weights used in EgoAtlas pretraining dataset.

Training Dataset Mixture	
H2O [23]	0.8%
OAKINK [58]	1.9%
PH2D [39]	5.4%
ARCTIC [8]	2.8%
EgoDex [12]	40.3%
Holoassist [52]	10.6%
ActionNet [43]	13.4%
Our Enhanced Data	25.4%

During pretraining, we jointly optimize all model parameters, including the vision encoder, the LLM backbone, and the action decoder. We use a global batch size of 768 (32 per device) and use distributed training under a fully sharded data-parallel (FSDP) setup. Optimization is performed with AdamW using a learning rate of $2e^{-5}$, no weight decay, and gradient clipping at a max-norm of 1.0. The action loss weight λ is set to 1. We train on a cluster of 24 NVIDIA H100 GPUs. Empirically, we find that 60k training steps are sufficient to achieve strong performance, which requires approximately 72 hours.

E.2 Posttraining Details

For post-training, we use an 8-GPU setup with a per-device batch size of 4. We apply LoRA updates with a rank of 32 to both the LLM backbone and the vision encoder, while performing full-parameter fine-tuning on the action decoder. Only the LoRA-adapted weights and the action decoder parameters are included in the trainable set. We optimize the model using AdamW with a learning rate of $3.5e^{-4}$ and a weight decay of $1e^{-3}$. The learning rate schedule follows a StepLR policy, where the learning rate is decayed by a factor of 0.1 after 80% of the total training steps. We train the model for 80000 steps in total. To improve training efficiency, we adopt mixed-precision training: both the LLM backbone and the vision encoder are trained in bfloat16 precision, whereas the action decoder is kept in float32 for numerical stability. This configuration provides a favorable trade-off between speed and accuracy during post-training.

For the robot observation–action interface, we set the proprioceptive history to 1 and use an action chunk length of 32. Real-world data are collected at 30 Hz. The recorded images are first center-cropped to a resolution of 480×640, and subsequently resized to 224×224 before being fed into the visual encoder.

E.3 Baseline Settings

For OpenVLA-OFT [20], all experiments are conducted on 4 NVIDIA H100 GPUs. We use the full joint configuration of the dual-arm and dual-hand system as the policy action space (2*7 DoF for arm joints and 2*6 DoF for hand joints). We apply LoRA-based fine-tuning to the VLA and adopt an L1

regression–style action head for continuous action prediction. The action chunk length is set to 30. We use the following hyperparameters for training OpenVLA-OFT.

Table 10: Hyperparameters for OpenVLA-OFT.

Name	Value
Steps	40000
Batch Size	4
Learning Rate	$5e^{-4}$
Action Chunks	30
LoRA Rank	32
Action Head	$l1_regression$

For $\pi_{0.5}$ [15], post-training uses 1 NVIDIA H100 GPU across all tasks. We use wrist pose for arms and joints for dexterous hands (2*7 DoF for wrist poses and 2*6 DoF for hand joints). Action chunk size is set to 10. The hyperparameter configuration is as follows.

Table 11: Hyperparameters for $\pi_{0.5}$.

Name	Value
Steps	30000
Batch Size	32
Learning Rate	$5e^{-5}$
Action Chunks	10
Ema decay	0.999
Warm-up steps	10000

For GR00T N1.5 [3], All experiments are conducted on 4 NVIDIA H100 GPUs. We use joint angles as the policy action space, which contains two 7-DoF arm joints and two 6-DoF hand joints. Fine-tuning is set to freeze the language model backbone and vision tower, only fine-tuning the projector and diffusion model. The action chunk length is set to 16, and some hyperparameters are set as follows.

Table 12: Hyperparameters for GR00T N1.5.

Name	Value
Steps	10000
Batch Size	32
Learning Rate	$1e^{-4}$
Weight Decay	$1e^{-5}$
Warmup Ratio	0.05
Action Chunks	16

For ACT [66], we choose to follow the lerobot [7] framework. All training is conducted on a single NVIDIA H100 GPU. At this point, we unify the action space with METIS, using the 9D pose of the wrist and the 3D XYZ pose of the fingers as the action space. We use ResNet18 as the vision backbone for training, with the remaining transformer layers trained from scratch. The action chunk length is set to 100, and the hyperparameters are as follows.

For ACT, we adopt the same action space representation as METIS to enable a fair comparison. Instead, existing pretrained VLA models, including OpenVLA-OFT, $\pi_{0.5}$, and GR00T N1.5, are originally trained in joint-based action spaces. To avoid disrupting their learned action hierarchies and pretrained representations, we retain their native joint-level action formulations. All baselines are trained on the same datasets and evaluated under identical experimental conditions.

Table 13: Hyperparameters for ACT.

Name	Value
Steps	600000
Batch Size	8
Learning Rate	$1e^{-5}$
Weight Decay	$1e^{-4}$
Warmup Ratio	0.05
Action Chunks	100
Vision Backbone	ResNet-18

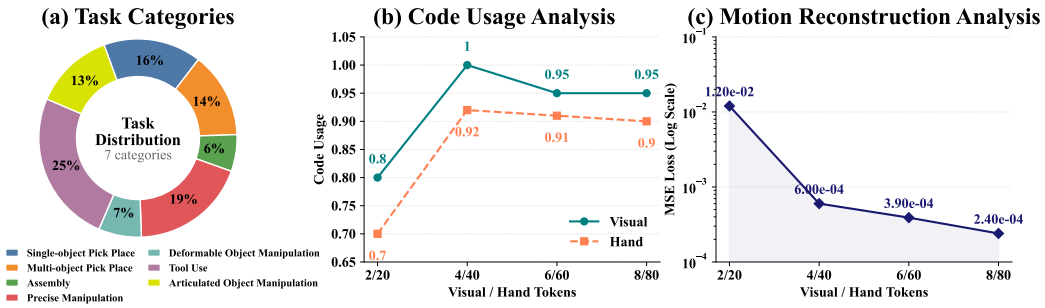
F Cross-Embodiment Settings

In the cross-embodiment experiments, we employ the Sharpa Beta embodiment equipped with a pair of 22-DoF SharpaWave Dexterous Hands. During the data collection phase, we capture first-person view images at 30 Hz along with joint state-action pairs at 60 Hz. The data processing pipeline begins with temporal alignment between the 30 Hz images and 60 Hz joint angles, synchronizing both modalities to 30 Hz. For the joint angle information, we perform forward kinematics (FK) calculations and coordinate frame transformations: the fingertip poses (3-dimensional) are transformed into the wrist coordinate frame, while the wrist poses (9-dimensional) are converted to the camera coordinate frame, ensuring a unified action space. During inference, the model's output fingertip and wrist poses undergo inverse kinematics (IK) computations to convert them back to joint angles for robot control.

G Quantitative Results and Visualization

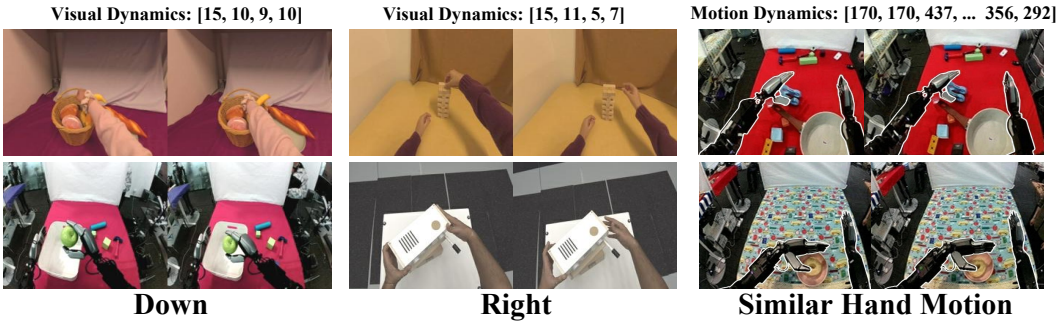
We analyze the effect of motion-aware tokens on code usage and motion reconstruction under different token budgets as shown in Figure 11 (b) and (c). Increasing the number of dynamics tokens improves reconstruction fidelity but also incurs higher inference cost, revealing a clear accuracy–efficiency trade-off. Our 4+40 setting strikes a favorable balance, achieving strong reconstruction accuracy while maintaining efficient code utilization and low latency. Further increasing the token budget yields only marginal gains in motion reconstruction quality.

Figure 11: **(a) Task categories and distribution of the collected dataset. (b) Quantitative analysis of code usage with dynamics tokens. (c) Quantitative analysis of motion reconstruction with dynamics tokens.**



We also visualize learned codewords in Figure 12. Notably, across different embodiments and scene contexts, consistent motion trends (e.g., downward or rightward movements) and similar hand pose are quantized into the same codewords, despite variations in visual appearance. This suggests that motion-aware tokens capture underlying motion semantics rather than surface-level configurations, enabling invariance to embodiment and environmental changes. Such structured tokenization indicates that the learned representation forms a set of reusable motion primitives, which facilitates efficient encoding and generalization across diverse manipulation scenarios.

Figure 12: **Visualization of Codewords.**



H Failure Case Analysis

Through extensive real-world experiments, we identify two primary categories of failure modes that can adversely affect the performance of METIS. The first category of failures arises from the inherent limitations of a **single egocentric viewpoint**. In tasks that require large-range manipulations or precise interactions with small objects, the restricted field of view often causes target objects to move outside the camera's visible region or become partially occluded. Such visibility degradation can lead to inaccurate perception and, consequently, suboptimal or failed action execution. The second failure mode stems from **Inverse Kinematics error**. When controlling high DoF dexterous hands, deriving joint configurations solely from fingertip positions via IK can yield singular or ill-conditioned solutions. Such instability may manifest as unintended finger oscillations or jitter during execution, which in turn degrades manipulation precision and can lead to task failure. Introducing additional kinematic constraints into the IK solver or adopting a learning-based IK predictor may be potential mitigation strategies.

I Limitations

Despite the exceptional performance demonstrated by METIS, several limitations remain. First, our model relies solely on egocentric observations, which may restrict its ability to perceive complete object geometry and fine interaction details. This limitation could be mitigated by incorporating additional wrist-mounted or external cameras. Second, the pretraining process currently excludes large-scale third-person data available online. Extending pretraining to broader multi-view manipulation datasets represents a promising direction for future work.

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: We provide clear contributions and scope of the paper in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We provide discussion of limitations in Appendix I.

   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: [Yes]

Justification: We have provided the theoretical assumption and proof in the method and Appendix.

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: We have fully disclosed all the information needed to reproduce the main experimental results of the paper.

   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: Both the code and the data will be made available.

Guidelines:

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

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: The experimental setting is clearly presented in the paper.

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: [No]

Justification: We report success rates over multiple trials (e.g., 20 trials per task). The reported metrics are deterministic given fixed evaluation protocols.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The paper provides sufficient information on the computer resources.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: We follow the rules of 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: We discuss potential positive societal impacts.

    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 poses no such risks.

    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 the codes and data have been developed under the correct 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: [Yes]

    Justification: We introduce a new dataset (EgoAtlas). We plan to release the dataset, code, and model with comprehensive documentation.

    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: We did not involve with human subjects.

    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 involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A].

Justification: The core contribution in this paper does not involve LLMs as any important, original, or non-standard components.

Guidelines:

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