EGOSTEER: AN OPEN-SOURCE FULL-STACK SYSTEM TOWARDS STEERABLE DEXTEROUS MANIPULATION FROM EGOCENTRIC VIDEOS

Anonymous authors
Paper under double-blind review

ABSTRACT

The enduring vision of general-purpose robots serving humanity hinges fundamentally on policy *steerability*. However, prevailing paradigms of learning from expert demonstrations demand massive real-world data even on simplified grippers, rendering them prohibitively expensive for high-dimensional, data-scarce dexterous hands. To overcome this bottleneck, we present a full-stack system that scales dexterous VLA pre-training from *egocentric human videos* and enables data-efficient real-robot post-training. It integrates **EgoSmith**, a data pipeline that curates in-the-wild egocentric videos into $9,606$ hours of pre-training data with $8.3\times$ higher throughput and better accuracy than prior SOTA; a unified **Robot Stack** for teleoperation and human-in-the-loop correction tailored for dexterous hands; and **EgoSteer**, a world-model-enhanced VLA operating on a morphology-aligned action space. Human data pre-training equips EgoSteer with language-guided manipulation priors, which are grounded through robot post-training and further refined via DAgger. Empirically, EgoSteer robustly executes free-form instructions across 45 diverse tasks, demonstrating adherence to user intent amid multiple candidate tasks and generalization. The pre-trained model also few-shot adapts to five complex long-horizon tasks, including box folding, on two embodiments with 79% average progress. All system code, datasets, model checkpoints, and an evaluation gallery are publicly available at `ego-steer.github.io`.

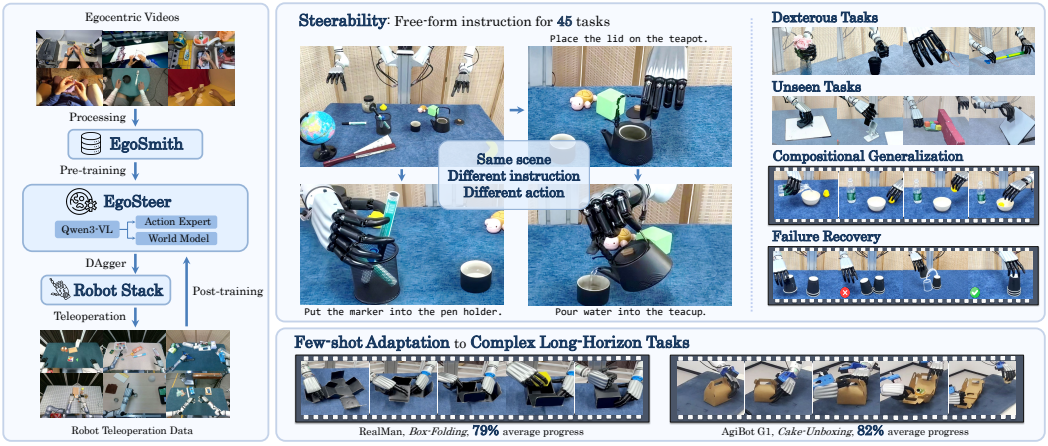


Figure 1: Our **full-stack system** integrates EgoSmith, Robot Stack, and EgoSteer to learn from large-scale egocentric human videos and facilitate data-efficient real-robot post-training, enabling steerable dexterous manipulation across 45 tasks as well as few-shot adaptation to complex, long-horizon tasks.

1 INTRODUCTION

A central goal of embodied intelligence is to enable robots to perform diverse manipulation tasks from open-ended human intent. Despite rapid progress in embodied foundation models (Black et al., 2026; Liu et al., 2025; Luo et al., 2025; Wu et al., 2026a; Brohan et al., 2023; Zitkovich et al., 2023),

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

most systems still require task-specific fine-tuning, while the few that follow free-form language are largely limited to grippers (Intelligence et al., 2026; Ye et al., 2026). Dexterous hands provide a more expressive embodiment that offers greater actuation capacity and fine-grained interaction potential. Yet on such challenging platforms, steerable manipulation remains largely unrealized.

The key bottleneck lies in data and system scalability. While language-guided manipulation typically demands thousands of hours of data even on simplified grippers (Intelligence et al., 2026; Agarwal et al., 2026; Liu et al., 2026a; Wu et al., 2026a), collecting sufficient expert demonstrations directly on high-dimensional, data-scarce dexterous hands is prohibitively expensive. Egocentric human videos (Hoque et al., 2025; Punamiya et al., 2026) offer a scalable alternative, as human hand manipulation contains rich interaction knowledge and is abundant in the wild. However, raw egocentric videos are noisy and lack reliable language and action annotations. Without systematic curation, these unstructured videos yield unstable supervision that can degrade downstream robot policies. Even with curated human data, the system must employ high-capacity models trained with effective objectives, and ground the learned priors to the target robot. Failing to address any of these interdependent components prevents the realization of general language-following manipulation.

In this paper, we close this gap by proposing a full-stack system for steerable dexterous manipulation that scales VLA pre-training from egocentric human videos and enables data-efficient real-robot post-training. Our system begins with **EgoSmith**, a data pipeline that transforms in-the-wild video into clean, fully-annotated pre-training data. Through a tailored four-stage design, EgoSmith aligns RGB-D frames, world-space hand trajectories, camera parameters, and textual instructions, curating a 9.6K-hour pre-training corpus across 12 egocentric datasets. In particular, for 4D motion estimation, the pipeline's primary accuracy and computational bottleneck, we improve upon the open-source SOTA HaWoR (Zhang et al., 2025) to achieve higher precision and a $8.3\times$ throughput speedup. To ground these human-hand interaction priors onto physical embodiments, we design a unified **Robot Stack** for teleoperation, inference, and human-in-the-loop correction. With this stack, we collect 187 hours of high-quality teleoperation data across 193 semantically-diverse tasks for post-training. During deployment, it enables seamless expert intervention for efficient DAgger (Ross et al., 2011) refinement. Finally, we introduce **EgoSteer**, a world-model-enhanced VLA operating on a morphology-aligned action space. By incorporating a world-model expert that predicts action-induced future states in the DINOv3 (Siméoni et al., 2025) latent space, EgoSteer enhances the VLM backbone's action imagination and modality alignment. To facilitate human-robot transfer, EgoSteer employs a unified action space of wrist poses and fingertip keypoints (Cai et al., 2025; Fu et al., 2025), coupled with training-time RTC (Black et al., 2025) to eliminate real-world execution pauses. Data-loading optimizations are further introduced to stabilize training at scale.

Empirically, through large-scale egocentric pre-training, real-robot post-training, and DAgger refinement, EgoSteer robustly follows free-form instructions across 45 tasks with a 75% average success rate. In cluttered scenes with multiple candidate tasks, it reliably executes commanded objectives while generalizing to novel skill compositions and unseen action semantics. Moreover, without explicit training, the policy exhibits notable instruction switching, immediately adapting mid-execution to abrupt command changes requiring drastically different motions. Together, these results confirm that EgoSteer genuinely grounds language guidance rather than memorizing spurious scene-task correlations. In comparative evaluations, EgoSteer consistently outperforms strong baselines such as $\pi_{0.5}$ (Intelligence et al., 2025) and Being-H0.5(Luo et al., 2026), with systematic ablations confirming the necessity of each system design. Additionally, the pre-trained manipulation priors also enable few-shot adaptation to challenging long-horizon tasks, such as box folding and cake unboxing, across two embodiments with 79% average progress rate. Conversely, our from-scratch variant, $\pi_{0.5}$, and sample-efficient baselines including Diffusion Policy (DP) (Chi et al., 2025) and IMLE (Rana et al., 2025) struggle to make substantial progress, underscoring both task difficulty and the efficacy of our curated priors. To summarize, our contributions are five-fold:

- **EgoSmith**, an egocentric data pipeline that curates a 9.6K-hour fully-annotated pre-training corpus across 12 datasets, achieving a $8.3\times$ throughput and better accuracy over prior SOTA.
- **A unified robot stack** for teleoperation, inference, and seamless human-in-the-loop correction, that collects 187 hours of diverse data across 193 dexterous tasks.
- **EgoSteer**, a world-model-enhanced VLA operating on a morphology-aligned action space.
- **Extensive real-robot evaluations** demonstrating that EgoSteer robustly performs free-form language-following across 45 tasks and few-shot adapts to complex long-horizon tasks.
- **Open-source release** of our complete system, curated datasets, and model checkpoints.

2 Related Work

Generalist robot manipulation policies. Building generalist policies has long been a core pursuit in embodied intelligence. Early work often adopts hierarchical architectures, decomposing the vision-language-to-action mapping through intermediate action representations (Zhong et al., 2025), such as trajectories (Gu et al., 2024) and affordances (Huang et al., 2023), to leverage foundation model capabilities. However, these methods are typically constrained by the expressiveness of intermediate representations and compounding errors across modules, restricting their effectiveness to narrow scenarios. Subsequent research has shifted toward scaling diverse datasets for end-to-end training. Focusing on *parallel-jaw grippers*, works such as $\pi_{0.5}$ (Intelligence et al., 2025), $\pi_{0.7}$ (Intelligence et al., 2026), LingBot-VLA (Wu et al., 2026a), and G0.5 (Liu et al., 2026b) scale tens of thousands of hours of robot data, while RDT-2 (Liu et al., 2026a) leverages 10K hours of UMI (Chi et al., 2024) demonstrations, achieving multi-task instruction following. In contrast, steerable manipulation on *dexterous hands* remains largely unexplored, primarily due to the severe scarcity of open-source real-robot data and prohibitive data collection costs. To address this bottleneck, EgoSteer leverages egocentric human data for scalable pre-training and enables data-efficient post-training with fewer than 200 hours of robot data, demonstrating steerability on dexterous hands.

Architecturally, a mainstream paradigm for end-to-end robot policies is vision-language-action (VLA) models, encompassing autoregressive formulations (Kim et al., 2024; Pertsch et al., 2025), flow-matching experts (Black et al., 2026), and hybrid designs that combine both (Driess et al., 2026). More recently, world action models (WAMs) integrate action policies with future video or image prediction. Representative approaches include joint visual-action prediction (Ye et al., 2026), action generation conditioned on visual forecasts (Li et al., 2026c; Zhang et al., 2026b; Hu et al., 2026), decoupled visual and action prediction (Yuan et al., 2026), and unified models that simultaneously optimize diverse world modeling objectives (Bi et al., 2026; Agarwal et al., 2026; Lyu et al., 2026). However, policies reliant on visual generation typically suffer from heavy computational overhead and high inference latency, while unified models face challenges in balancing multiple learning objectives. To circumvent these limitations, EgoSteer adopts a lightweight paradigm by incorporating a world model objective into a flow-based VLA backbone: during training, it predicts action-conditioned future DINO features (Siméoni et al., 2025) to endow the policy with foresight over physical outcomes, thereby enhancing instruction-following fidelity and manipulation precision; during inference, it generates only actions while discarding the visual prediction branch, ensuring minimal computational overhead and high-frequency closed-loop execution.

Scaling with egocentric human data. Egocentric human videos offer a diverse and scalable data source for learning manipulation priors. Although numerous datasets have been released (Hoque et al., 2025; Punamiya et al., 2026; Li et al., 2026d), their annotation comprehensiveness varies widely, with many containing only unannotated monocular RGB streams. While recent pipelines (Li et al., 2025; Luo et al., 2026) attempt to recover missing control modalities, they remain closed-source and constrained by throughput and reconstruction fidelity. To this end, our open-source EgoSmith pipeline improves upon HaWoR to achieve higher reconstruction precision alongside a $8.3\times$ throughput speedup, with rigorous multi-stage quality filtering. In downstream policy learning, existing works have verified that egocentric human data can improve robot performance (Yang et al., 2025; Fu et al., 2025; Li et al., 2026a), reduce real-robot data requirements (Luo et al., 2025), facilitate cross-embodiment transfer (Luo et al., 2026; Zhang et al., 2026a), exhibit scaling behaviors (Zheng et al., 2026), or serve as co-training data to enhance performance on grippers (Intelligence et al., 2026; Wu et al., 2026b). However, its effect on the steerability of dexterous hand models has been underexplored. We close this gap by proposing and open-sourcing a full-stack system that pre-trains a world-model-enhanced VLA on 9.6K hours of curated egocentric data, post-trains it with 187 hours of self-collected teleoperation data, and refines it with DAgger. The resulting model, EgoSteer, faithfully follows language instructions across over 40 tasks, and we demonstrate through ablations that the key designs of our full-stack system are all beneficial to steerability.

3 EgoSmith: Curating Egocentric Videos into Dexterous Priors

While egocentric data are rich in fine-grained interactions and highly scalable for general embodied learning, they are typically monocular RGB videos suffering from camera jitter, frequent occlusions, and a lack of annotations. This section presents **EgoSmith** (Figure 2), an efficient automated pipeline that transforms massive raw videos into fully-annotated training samples to enable effective learning. Full details can be found in Appendix A.

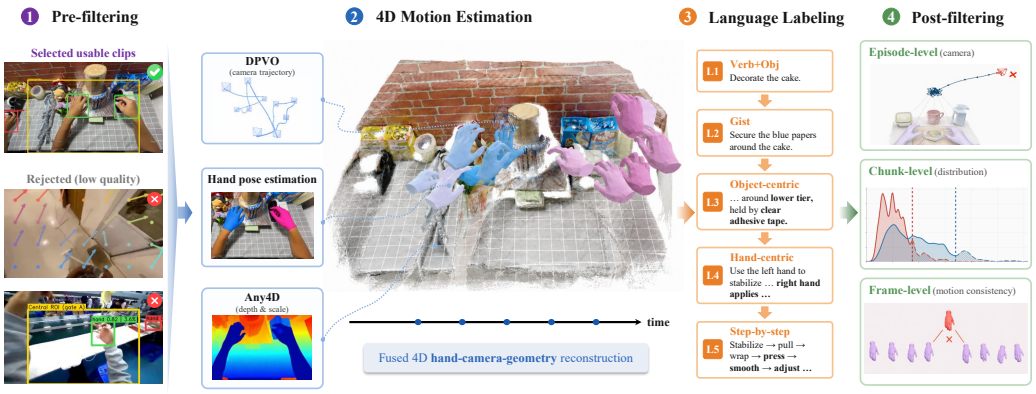


Figure 2: Overview of EgoSmith. Integrating pre-filtering, 4D motion estimation, language labeling, and post-filtering, EgoSmith efficiently curates in-the-wild egocentric videos into clean and annotated training samples.

To achieve raw data cleaning, labeling, and quality control, EgoSmith employs a four-stage pipeline. The **first** stage, **pre-filtering**, uses simple yet effective heuristics to discard locomotion segments, excessive head movements, and hand misidentifications that degrade downstream 4D motion estimation. Specifically, we filter out locomotion and abrupt head motion by detecting excessive camera motion via RANSAC-based (Hartley & Zisserman, 2003) sparse optical flow over a 128-point grid. We then eliminate frames with severe occlusions or bystander interference by applying geometric criteria to the hand counts, scales, and coordinates detected by YOLO (Redmon & Farhadi, 2018; Potamias et al., 2025), preserving only clearly visible egocentric manipulations.

Table 1: Benchmark results of 4D motion estimation. Errors are in mm; time is in seconds per episode.

Method	RPE ↓	ATE ↓	WA-MPJPE ↓	W-MPJPE ↓	Time ↓
HaWoR	5.63	9.12	34.9	106.7	95.04
EgoSmith (Ours)	**2.60**	**7.57**	**26.7**	**88.2**	**11.45**

Building upon the state-of-the-art method HaWoR, the **second** stage, **4D motion estimation**, reconstructs camera extrinsics, depth, and world-space hand trajectories. Since HaWoR lacks dense depth estimation and relies on DROID-SLAM (Teed & Deng, 2021) for tracking, which is computationally expensive and prone to drift under rapid head movements and drastic scene changes, we propose a more robust and efficient scheme. We leverage DPVO (Teed et al., 2023) for more stable, *metric-free* camera tracking and keyframe depth estimation, and Any4D (Karhade et al., 2025) for frame-wise, *metric-scale* depth prediction. Aligning their scale ratio recovers more accurate metric-scale camera trajectories, which we use to transform camera-frame hand motions into more consistent world-space trajectories. Beyond adopting DPVO, which is far more lightweight than DROID-SLAM, we incorporate batching and asynchronous I/O pipelining to further maximize processing throughput. Using NVIDIA A800 GPUs, we benchmark EgoSmith against HaWoR on a random subset across four datasets with ground-truth annotations: TACO (Liu et al., 2024), OakInk-v2 (Zhan et al., 2024), H2O (Kwon et al., 2021), and EgoVerse (Punamiya et al., 2026). This subset comprises 512 episodes with an average duration of 15.26 seconds at 30 FPS. As shown in Table 1, EgoSmith outperforms HaWoR in both accuracy and efficiency. Lower RPE and ATE indicate that the DPVO-based camera trajectory estimation yields superior local consistency and structural robustness, while lower WA-MPJPE and W-MPJPE show that we effectively mitigate scale distortion and long-term drift, ensuring physically plausible world-space hand tracking over extended sequences. Moreover, EgoSmith reduces the per-episode runtime from 95.04 to 11.45 seconds, achieving an 8.3× speedup to facilitate large-scale processing.

The **third** stage, **language labeling**, performs multi-granularity language annotation, which is crucial for enabling free-form instruction following. We first employ Qwen3.5-VL-Plus (Qwen Team, 2026) to filter out segments lacking meaningful hand-object manipulation. For the remaining clips, the model generates coarse-to-fine, five-level language instructions, ranging from concise verb-object phrases to detailed step-by-step procedures. This hierarchical annotation simultaneously provides task-level semantic grounding and action-level spatiotemporal grounding, enabling downstream models to learn and respond to instructions across varying levels of abstraction.

The **fourth** stage, **post-filtering**, applies complementary quality checks at the episode, chunk, and frame levels. At the episode level, we compute the mean camera translation and rotation changes and

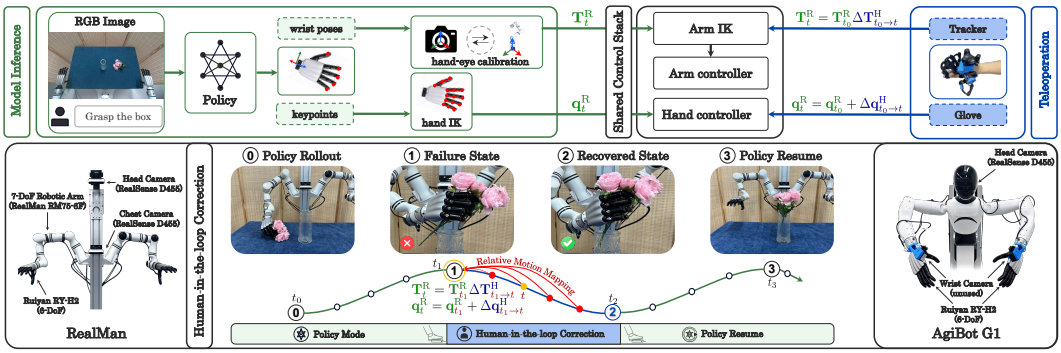


Figure 3: Overview of the Robot Stack. It unifiedly supports teleoperation, policy inference, and human-in-the-loop correction. A relative motion mapping scheme is employed to facilitate seamless transitions during interventions, and the bottom row illustrates the two robotic embodiments utilized in our experiments.

reject episodes that are outliers with respect to their dataset-wide IQR distributions. At the chunk level, we transform wrist positions within a sliding window into the camera coordinate system of each anchor frame and express fingertip positions in the corresponding wrist coordinate systems. We reject episodes whose coordinate extrema fall outside dataset-derived IQR fences or predefined physical bounds. At the frame level, we measure consecutive-frame changes in camera translation and rotation, wrist translation and rotation, and fingertip translation. An episode is rejected if any of these changes exceeds physically plausible limits, thereby removing abrupt reconstruction jumps. Together, these checks remove unreliable episodes caused by discontinuous motion estimates, implausible metric scales, and camera-tracking drift.

With EgoSmith, we curate a large-scale egocentric pre-training corpus across 12 raw datasets (Liu et al., 2024; Kwon et al., 2021; Zhan et al., 2024; Wang et al., 2023; Garcia-Hernando et al., 2018; Banerjee et al., 2025; Damen et al., 2018; Grauman et al., 2022; Punamiya et al., 2026; Hoque et al., 2025; AI, 2025b;a). To filter out highly repetitive videos, we subsample Egocentric-10K (AI, 2025b) and Egocentric-100K (AI, 2025a). For Ego4D (Grauman et al., 2022) and EPIC-KITCHENS (Damen et al., 2018), we apply EgoSmith to their respective VITRA (Li et al., 2025) subsets. Ultimately, this pipeline yields a fully-annotated egocentric dataset comprising 9.60K hours, 2.29M episodes, and 1.04B frames.

4 A Robot Stack for Teleoperation and DAgger Post-Training

While EgoSmith's curated egocentric data provides rich manipulation priors, direct transfer to real robots is hindered by the embodiment gap across vision and control, necessitating real-robot teleoperation to ground these priors onto the target embodiment. This section presents the unified **Robot Stack** (Figure 3), which shares low-level control and dynamics to simultaneously support teleoperation, policy inference, and human-in-the-loop correction. Full details can be found in Appendix B.

For teleoperation, a pair of PsiBot SynGlove-Air gloves and Vive Trackers capture the operator's $SE(3)$ wrist poses and hand joint angles, which respectively drive two robotic arms through inverse kinematics (IK) computed via `mink` (Zakka, 2026) and two 6-DoF robotic hands via joint mapping. During policy inference, the policy publishes wrist pose trajectories in the camera frame and hand keypoints in the wrist frame. These actions share the same arm and hand FK/IK and control nodes with teleoperation, ensuring identical execution dynamics across training and inference.

The primary challenge in enabling human-in-the-loop intervention is preventing sudden state jumps at the handover boundary to ensure a smooth transition. To address this, we propose a *relative motion mapping* scheme. When the operator signals intervention by pressing a foot pedal at step t, the system records the robot end-effector poses $\mathbf{T}_t^{\mathrm{R},i} \in SE(3)$, human wrist poses $\mathbf{T}_t^{\mathrm{H},i} \in SE(3)$, robot hand joint states $\mathbf{q}_t^{\mathrm{R},i} \in \mathbb{R}^6$, and glove states $\mathbf{q}_t^{\mathrm{H},i} \in \mathbb{R}^6$ for each arm/hand index $i \in \{1, 2\}$. Subsequently, at any $t' \geq t$, the operator's relative motions, $\Delta\mathbf{T}_{t\to t'}^{\mathrm{H},i} = (\mathbf{T}_t^{\mathrm{H},i})^{-1}\mathbf{T}_{t'}^{\mathrm{H},i}$, $\Delta\mathbf{q}_{t\to t'}^{\mathrm{H},i} = \mathbf{q}_{t'}^{\mathrm{H},i} - \mathbf{q}_t^{\mathrm{H},i}$, are mapped to the robot, computing the commanded end-effector poses $\mathbf{T}_{t'}^{\mathrm{R},i}$ and hand joint states $\mathbf{q}_{t'}^{\mathrm{R},i}$ as $\mathbf{T}_{t'}^{\mathrm{R},i} = \mathbf{T}_t^{\mathrm{R},i}\Delta\mathbf{T}_{t\to t'}^{\mathrm{H},i}$ and $\mathbf{q}_{t'}^{\mathrm{R},i} = \mathbf{q}_t^{\mathrm{R},i} + \Delta\mathbf{q}_{t\to t'}^{\mathrm{H},i}$. This formulation allows the operator to smoothly take over control by simply mimicking the robot's motion, enabling efficient collection of corrective demonstrations.

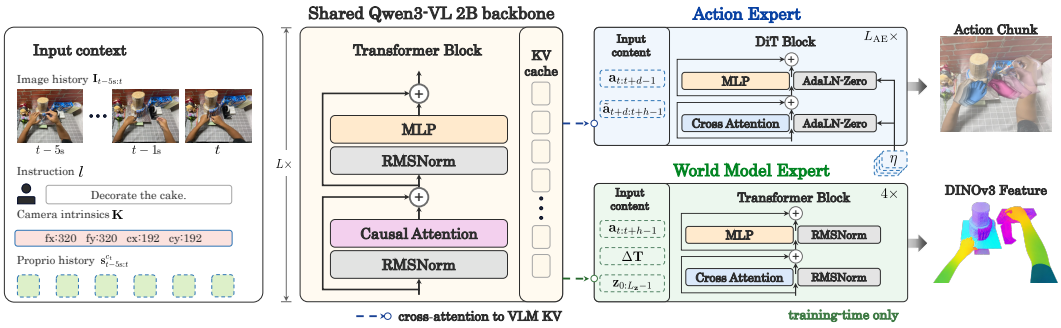


Figure 4: Overview of EgoSteer, a world-model-enhanced VLA model for steerable dexterity. A shared Qwen3-VL backbone extracts KV cache representations from multi-modal inputs. The action expert jointly attends to itself and the backbone to generate action chunks via flow-matching, integrating training-time RTC to eliminate execution pauses. The training-only world model expert predicts future DINOv3 features to improve action accuracy with zero inference overhead.

With unified robot stack, we collect a 187-hour dataset across 193 tabletop tasks spanning seven categories: Pick-and-Place(PnP)-Easy/Medium/Hard, Non-prehensile, Reorient, Bimanual, and Contact-rich. These comprise 56 common tasks covering most core manipulation primitives, alongside 137 long-tail tasks to facilitate human-to-robot transfer and grounding. Multi-level language annotations are generated using Qwen3-VL-Flash (Bai et al., 2025), followed by manual verification. To cover diverse primitives and establish modal alignment, data collection follows a free-form protocol where environments are cluttered, and tablecloths, object instances, and initial configurations are randomized without pre-defined trajectories. This emphasis on natural, human-like execution yields substantial rollout diversity, fostering policy robustness and multi-task generalization.

5 EgoSteer: A World-Model-Enhanced VLA for Steerable Dexterity

To effectively learn language-guided manipulation from human data, teleoperation, and corrections, we propose **EgoSteer**, a flow-based VLA model enhanced by a world-model objective, shown in Figure 4. To ensure robust vision-language understanding while modeling multimodal continuous actions, EgoSteer pairs a Qwen3-VL 2B backbone (Bai et al., 2025) with a DiT-based (Peebles & Xie, 2023) action expert, which jointly attends to itself and backbone to generate action chunks via flow-matching (Black et al., 2026). To facilitate human-robot transfer, we design a unified data format and state-action space across both domains. An episode τ of length N is represented as $\tau = \{l, \mathbf{K}, (\mathbf{I}_t, \mathbf{D}_t, \mathbf{T}_t^{w2c}, \mathbf{s}_t^w, \mathbf{a}_t^w)_{t=0}^{N-1}\}$, where l is the instruction, $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the camera intrinsics, t is the timestep, $\mathbf{I}_t \in \mathbb{R}^{H\times W\times3}$ and $\mathbf{D}_t \in \mathbb{R}^{H\times W\times1}$ are the RGB and depth images, $\mathbf{T}_t^{w2c} \in SE(3)$ is the world-to-camera extrinsics, and bimanual world-frame states and actions $\mathbf{s}_t^w, \mathbf{a}_t^w \in \mathbb{R}^{48}$ comprise the 3D wrist translation, 6D wrist rotation, and 15D fingertip keypoints of both hands. Beyond this unified action space, we align the robot's wrist coordinate frame to match human anatomical proportions, minimizing the domain gap to facilitate knowledge transfer (see Appendix B.1.2 for details). Since depth is unused in our model, the training sample at each timestep t becomes $\{l, \mathbf{K}, \mathbf{I}_{t-k+1:t}, \mathbf{s}^{c_t}_{t-k+1:t}, \mathbf{a}^{c_t}_{t:t+h-1}\}$, where k and h denote the history and prediction lengths, and $\mathbf{s}^{c_t}_{t-k+1:t}$ is the state history transformed into the current camera frame c_t via $\mathbf{T}_t^{w2c}$. The relative action chunk $\mathbf{a}^{c_t}_{t:t+h-1}$ is computed in c_t relative to $\mathbf{s}_t^{c_t}$, where wrist motions are relative $SE(3)$ transforms and finger movements are coordinate displacements. For simplicity, we omit the c_t superscript for $\mathbf{s}$ and $\mathbf{a}$ hereafter. Furthermore, to avoid execution pauses during real-robot inference, we implement training-time Real-Time Chunking (RTC) (Black et al., 2025) in the action expert. Specifically, we feed a clean action prefix $\mathbf{a}_{\text{pre}} = \mathbf{a}_{t:t+d-1}$ of randomly sampled delay d as ground truth and train the expert solely to denoise the subsequent actions $\tilde{\mathbf{a}}_{\text{suf}} = \tilde{\mathbf{a}}_{t+d:t+h-1}$. During deployment, the robot executes the reserved prefix $\mathbf{a}_{\text{pre}}$ during asynchronous VLA inference, transitioning seamlessly to the new chunk $\mathbf{a}_{\text{suf}}$ without execution gaps. Denote our model by π, we train it using Conditional Flow Matching (CFM) (Lipman et al., 2023) by regressing the linear velocity field of the target suffix $\mathbf{a}_{\text{suf}}$ conditioned on the context $\mathbf{C}_t = \{l, \mathbf{K}, \mathbf{I}_{t-k+1:t}, \mathbf{s}_{t-k+1:t}, \mathbf{a}_{\text{pre}}\}$ with $\mathcal{L}_{\text{CFM}}(\pi) = \mathbb{E}_{t,\eta,\boldsymbol{\epsilon}}\left[\|\pi(\tilde{\mathbf{a}}_{\text{suf}}, \eta, \mathbf{C}_t) - (\mathbf{a}_{\text{suf}} - \boldsymbol{\epsilon})\|^2\right]$, where $\eta \in [0,1]$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and

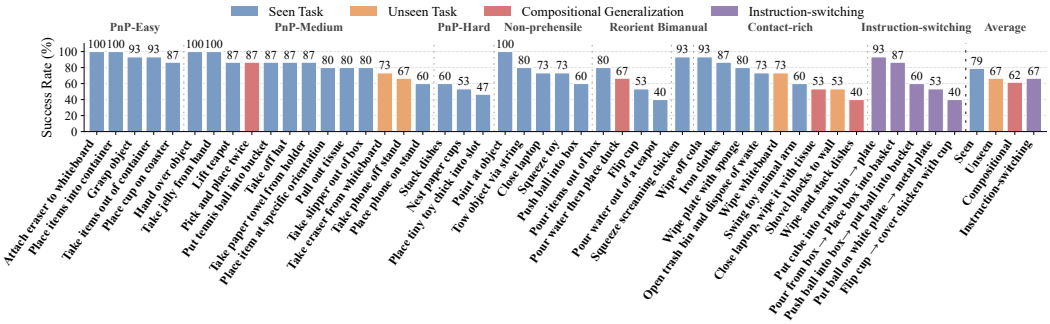


Figure 5: Steerable manipulation performance of EgoSteer across 45 tasks spanning 8 categories. It robustly follows free-form language instructions to achieve an overall success rate of 75%, demonstrating generalization.

$\tilde{\mathbf{a}}_{\text{suf}} = (1-\eta)\boldsymbol{\epsilon} + \eta \mathbf{a}_{\text{suf}}$. To expand the effective batch size and improve loss gradient, we sample four random η per sample.

While VLA excels at vision-language understanding, lacking future imagination limits its action generation accuracy (Ye et al., 2026). To address this, we introduce a world-model expert to predict action-induced future DINOv3 features. The expert takes the ground-truth $\mathbf{a}_{t:t+h-1}$, the relative camera motion $\Delta\mathbf{T} = \mathbf{T}_t^{w2c}(\mathbf{T}_{t+h-1}^{w2c})^{-1}$, and learnable query tokens $\mathbf{z}_{0:L_\mathbf{z}-1}$ of length $L_\mathbf{z}$ as inputs to jointly attend to themselves and the backbone. The upsampled output of $\mathbf{z}$ is supervised against future frame $\mathbf{I}_{t+h-1}$ DINOv3 features via regression loss, which provides a more direct and stable supervision signal than generative loss. For robot setups with an additional chest camera, the backbone inputs both cameras' image histories and intrinsics, while the expert receives their relative motions and regresses both future DINOv3 features by adding distinct camera embeddings to $\mathbf{z}$. To focus optimization on the backbone's representation, this module comprises only four Transformer layers attending to the backbone layers at regular intervals, guiding the gradient to primarily shape the backbone. Crucially, the expert is discarded during inference, avoiding computational overhead.

Since our training samples encompass sparse multi-frame visual histories over extended temporal windows, naive random-access data loading imposes a prohibitive I/O burden. To address this, our infrastructure implements *randomized sequential streaming* based on WebDataset and LeRobot v3 formats. This scheme drastically reduces I/O pressure through sequential shard reading, while preserving training randomness via shuffle buffers, random sample dropping, and randomized shard selection. Additionally, we incorporate Hybrid Sharded Data Parallel (HSDP) (Zhao et al., 2023), mixed-precision training, kernel fusion, and FlexAttention (Dong et al., 2024) to maximize compute throughput. Collectively, these optimizations ensure stable and highly efficient training at scale. Full details of EgoSteer and its training infrastructure can be found in Appendix C.

6 Experiments

We conduct extensive experiments to answer five core research questions: **Q1**. How well does EgoSteer follow free-form instructions to complete various tasks? **Q2**. Does DAgger efficiently and effectively improve performance? **Q3**. How does the pre-training scale affect downstream performance, and how does EgoSteer compare with other VLA baselines? **Q4**. Are egocentric pre-training data quality, the world-model objective, and training-time RTC essential to strong performance? **Q5**. Can large-scale egocentric pre-training enable few-shot adaptation to complex long-horizon tasks?

To thoroughly and rigorously investigate these questions, our evaluations utilize multiple EgoSteer checkpoints across various training configurations and stages. To prevent confusion, we outline their naming convention below. EgoSteer *generalist* models follow a three-stage training pipeline:

1. **Pre-training (PT)** on the egocentric dataset using a single head camera at 384×384 resolution.
2. **Fine-tuning (FT)** on the robot dataset using dual head and chest cameras at 640×480 resolution.
3. **DAgger refinement (DG)** under the identical dual-camera setup.

In contrast, EgoSteer *few-shot single-task* models require only two stages:

1. **Pre-training (PT)** on the same single-camera egocentric data.
2. **Few-shot adaptation (FS)** on a few robot demonstrations with the same single-camera setup.

Accordingly, we designate our checkpoints as **EgoSteer-nK-PT/FT/DG/FS**, where nK represents the hours of egocentric pre-training data (denoting a random subset of the full dataset when $0 <$

Table 2: Various experiments to validate the significance of system components and compare with baselines.

(a) DAgger ablation on four failure-prone tasks.

Method	Avg.
EgoSteer-9.6K-FT	22.5%
EgoSteer-9.6K-DG	**62.5%**

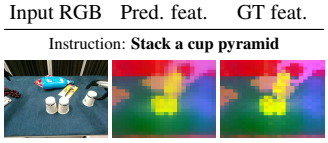
(b) Baseline comparison.

Method	Avg.
$\pi_{0.5}$	22%
Being-H0.5	39%
EgoSteer-9.6K-FT	**74%**

(c) Training ablation.

Method	Avg.
No WM-objective	31%
No training-RTC	39%
Noisy data	33%
EgoSteer-1K-FT	**44%**

(d) World-model visualization.



$n < 9.6$, with $n = 0$ indicating no pre-training), while PT, FT, DG, and FS indicate that the model is trained up to the pre-training, fine-tuning, DAgger, and few-shot stages, respectively. EgoSteer-nK-FT models consistently employ the full 187 hours of real-robot data for fine-tuning. Only the EgoSteer-9.6K-FT model undergoes DAgger refinement; specifically, three DAgger iterations are conducted, collecting 3.7K trajectories across 56 tasks, yielding 8.3 hours of correction data to refine the policy. The resulting model is denoted as **EgoSteer-**9.6**K-DG**, which is also referred to simply as **EgoSteer**. Full experimental details and training configurations are provided in Appendix D.

6.1 Steerable Multi-Task Manipulation and Generalization

Setup. To evaluate the steerability of EgoSteer, we benchmark it across 45 tasks: 32 seen, 4 compositional generalization, 4 unseen, and 5 instruction-switching. *Seen* tasks share action semantics with the fine-tuning dataset; *compositional* tasks require recombining seen primitives into novel sequences; *unseen* tasks introduce novel action semantics absent from fine-tuning; and *instruction-switching* tasks assess the ability to respond to a new instruction during execution (Chen et al., 2026). To ensure challenging evaluation conditions, tasks are performed in randomly cluttered scenes with multiple actionable distractors and commanded via unconstrained, template-free language. We evaluate each task over 15 randomized trials, reporting average success rates in Figure 5.

Results. As shown in Figure 5, EgoSteer achieves 80+% success on 24 tasks and an overall average of 75%. Crucially, across cluttered layouts, the policy strictly adheres to language instructions specifying target objects, hand assignments, and fine-grained motions, even when manipulating flat or small objects. Under identical initial scenes, EgoSteer faithfully carries out distinct tasks according to different instructions (Figure 13). When prompted mid-execution with an abrupt command, EgoSteer can immediately respond and fulfill the new goal (Figure 14). Furthermore, EgoSteer exhibits robust failure recovery, executing multiple retries if a step fails. It also achieves average success rates of 62% on compositional generalization and 67% on unseen tasks, confirming that our full-stack framework effectively harnesses rich priors from egocentric data to unlock robust, generalizable multi-task instruction following with fewer than 200 hours of real-robot demonstrations.

6.2 Efficacy of DAgger Post-Training

Setup. To evaluate the efficacy of DAgger, we compare EgoSteer-9.6K-DG and EgoSteer-9.6K-FT on four failure-prone, dexterous seen tasks, such as "*place phone on stand*". For each task, 10 evaluation trials are conducted using the same settings as in Section 6.1.

Results. As shown in Table 2a, after DAgger iterations totaling 8.3 hours, the average success rate increases from 22.5% to 62.5%. This efficacy stems from the targeted collection of corrective demonstrations addressing deployment failures, achieving a performance leap with minimal data. The refined policy not only exhibits robust failure recovery but also adaptively adjusts actions at critical manipulation bottlenecks. Crucially, these recovery and adjustment capabilities generalize to novel tasks, substantially improving the overall robustness of the DAgger-trained policy.

6.3 Scaling of Pre-Training and Baseline Comparisons

Setup. To investigate the scaling behavior of egocentric data, we compare the pre-training loss curves and robot performance of EgoSteer-0 / 3 / 6 / 9.6K-FT across 10 tasks. Additionally, the baselines $\pi_{0.5}$ and Being-H0.5 are post-trained on our robot data and evaluated across 10 simpler tasks.

Results. As shown by the pre-training loss curves in Figure 6a and the real-robot success and progress rates in Figure 6b, scaling pre-training data drives training loss to lower convergence values while improving real-world execution performance. With expanding pre-training data, the policy exhibits the emergence of failure recovery, enhanced instruction-following, and improved action accuracy, indicating that the model successfully acquires physical common sense for error adjustment and language-guided manipulation priors from increasingly larger datasets curated by EgoSmith.

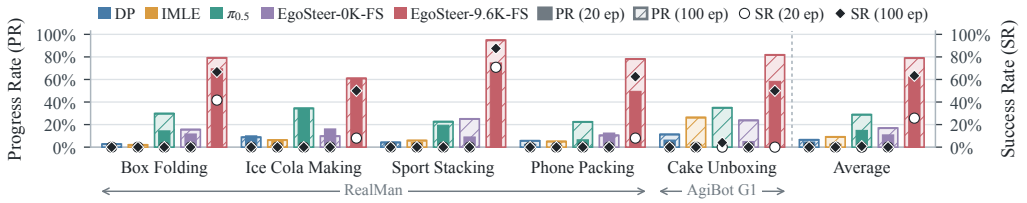


Figure 7: Few-shot performance across five challenging tasks on two embodiments. Ep stands for episodes.

The comparison of EgoSteer-9.6K-FT with the baselines is presented in Table 2b, where EgoSteer consistently outperforms both. Specifically, both baselines lack dedicated mechanisms for knowledge transfer to new dexterous hands, thereby limiting the efficacy of pre-training. Consequently, although they can handle basic PnP actions, they exhibit weak instruction-following, limited generalization, and imprecise execution. These performance gains highlight the advantage of our unified, full-stack system.

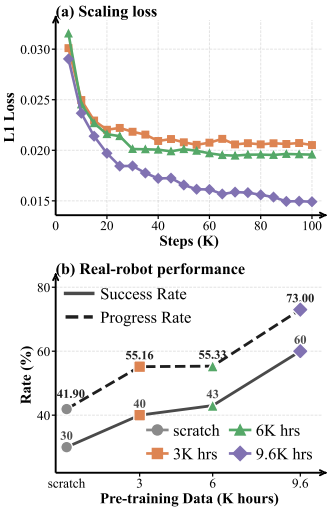


Figure 6: Scaling behavior of pre-training loss and downstream real-robot post-training performance.

6.4 Ablation Studies

Setup. We compare EgoSteer-1K-FT with three ablated variants across 10 seen tasks: first, *No WM-objective*, which omits the world-model expert during pre- and post-training; second, *No training-RTC*, which disables training-RTC during training and inference; and third, *Noisy data*, which utilizes noisy egocentric data unfiltered by EgoSmith pre- and post-filtering.

Results. As shown in Table 2c, removing any core component leads to a performance drop, validating the necessity of each module. Specifically, the *No WM-objective* variant exhibits a marked reduction in fine-grained manipulation accuracy, confirming that enhancing the backbone's action imagination is critical for precise action generation. As visualized in Table 2d, predictions from EgoSteer's world model expert align closely with the ground truth. The *No training-RTC* variant introduces severe action pauses and disrupts execution dynamics, causing contact-rich tasks to fail entirely due to continuous jitter. Finally, the *Noisy data* variant fails to converge effectively, leading to severe degradation in both instruction-following and manipulation precision.

6.5 Few-Shot Adaptation to Complex Long-Horizon Tasks

Setup. To investigate whether manipulation priors from egocentric pre-training promote sample-efficient robot learning, we few-shot adapt EgoSteer-9.6K-PT to five challenging long-horizon tasks, such as "*box folding*", across two embodiments using 20 and 100 demonstrations. These tasks comprise roughly ten fine-grained sub-steps and span approximately one minute. We benchmark EgoSteer-9.6K-FS against our non-pretrained ablation, $\pi_{0.5}$, and strong imitation learning baselines, namely DP and IMLE, over 24 real-world trials per task. Evaluation results and task setups are reported in Figure 7 and illustrated in Figure 15, respectively.

Results. As shown in Figure 7, EgoSteer-9.6K-FS consistently outperforms all baselines. Due to the extreme task complexity, all baselines make negligible progress even with 100 demonstrations. In contrast, using merely 20 demonstrations, EgoSteer-9.6K-FS achieves substantial progress and notable success rates; increasing to 100 demonstrations further yields marked improvements across both metrics. These results confirm that our 9.6K-hour pre-training imparts rich dexterous priors, enabling effective few-shot transfer to novel embodiments and complex manipulation tasks.

7 Conclusion

This paper presents and open-sources a full-stack system for steerable dexterous manipulation that scales egocentric pre-training and enables data-efficient real-robot post-training. It integrates EgoSmith, an efficient egocentric data pipeline; a unified robot stack for teleoperation, inference, and correction; and EgoSteer, a world-model-enhanced VLA with a morphology-aligned action space. EgoSteer demonstrates robust steerability across 45 diverse tasks, exhibiting dexterity, failure recovery, and generalization, while achieving few-shot adaptation to long-horizon tasks on two embodiments. Baseline comparisons and ablations confirm the criticality of system components.

AI USE STATEMENT

We used generative AI tools to assist in code implementation and debugging, literature search and summarization, and manuscript editing to improve language, clarity, and readability. We checked all the AI-assisted code for its correctness, cross-referenced the literature and citations with the original sources, and reviewed all AI-assisted text for factual accuracy and consistency with the methods and experimental results. The authors take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This study uses previously released egocentric datasets, cited in Figure 8a, and involved no new participants. Egocentric recordings may expose bystanders, personal information, and workplace activities. Therefore, we respect the privacy of the original participants and strictly followed the conditions and permitted uses of each dataset. During the study, we did not redistribute restricted clips or attempt to infer sensitive attributes from the recordings.

REPRODUCIBILITY STATEMENT

To ensure full reproducibility, extensive efforts have been made across all aspects of this work:

- **Code and Implementation:** The complete codebase of our full-stack system is open-sourced at our project website (`https://ego-steer.github.io/`). Exhaustive implementation details are documented in the main text and Appendices A to C.
- **Datasets:** The full 9.6K hours of egocentric data processed by EgoSmith and the 187 hours of real-robot data collected via the Robot Stack are fully prepared and will be publicly released upon publication. For review purposes, representative subsets of both datasets and an interactive viewer for the real-robot data are accessible on our project website.
- **Configurations and Checkpoints:** Appendix D details all experimental configurations. Model checkpoints for both EgoSteer-9.6K-PT and EgoSteer-9.6K-DG are provided on the website.
- **Video Recordings:** Extensive evaluation video recordings are hosted on the website to facilitate qualitative inspection and verification.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Build AI. Egocentric-100k, 2025a. URL `https://huggingface.co/datasets/builddotai/Egocentric-100K`.

Build AI. Egocentric-10k, 2025b. URL `https://huggingface.co/datasets/builddotai/Egocentric-10K`.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, et al. Hot3d: Hand and object tracking in 3d from egocentric multi-view videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7061–7071, 2025.

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A unified latent action world model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 35101–35113, 2026.

Kevin Black, Allen Z Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking. *arXiv preprint arXiv:2512.05964*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2026. URL https://arxiv.org/abs/2410.24164.

Jean-Yves Bouguet et al. Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm. *Intel corporation*, 5(1-10):4, 2001.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023. URL https://arxiv.org/abs/2212.06817.

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

Kaiyuan Chen, Shuangyu Xie, Zehan Ma, Pannag R Sanketi, and Ken Goldberg. Robo2vlm: Visual question answering from large-scale in-the-wild robot manipulation datasets. *arXiv preprint arXiv:2505.15517*, 2025.

Zhenyang Chen, Alan Tian, Liquan Wang, Benjamin Joffe, Yingyan Celine Lin, Yuxiao Chen, Siddharth Karamcheti, and Danfei Xu. Resteer: Quantifying and refining the steerability of multitask robot policies. *arXiv preprint arXiv:2603.17300*, 2026.

Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

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

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

Juechu Dong, Boyuan Feng, Driss Guessous, Yanbo Liang, and Horace He. Flex attention: A programming model for generating optimized attention kernels. *arXiv preprint arXiv:2412.05496*, 2(3):4, 2024.

Danny Driess, Jost Springenberg, Brian Ichter, Lili Yu, Adrian Li-Bell, Karl Pertsch, Allen Ren, Homer Walke, Quan Vuong, Lucy Xiaoyang Shi, et al. Knowledge insulating vision-language-action models: Train fast, run fast, generalize better. *Advances in Neural Information Processing Systems*, 38:102867–102888, 2026.

Yankai Fu, Ning Chen, Junkai Zhao, Shaozhe Shan, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. Metis: Multi-source egocentric training for integrated dexterous vision-language-action model. *arXiv preprint arXiv:2511.17366*, 2025.

Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3d hand pose annotations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 409–419, 2018.

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

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, Priya Sundaresan, Peng Xu, Hao Su, Karol Hausman, Chelsea Finn, Quan Vuong, and Ted Xiao. RT-trajectory: Robotic task generalization via hindsight trajectory sketches. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F1TKzG8LJO.

Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

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.

Yucheng Hu, Jianke Zhang, Yuanfei Luo, Yanjiang Guo, Xiaoyu Chen, Xinshu Sun, Kun Feng, Qingzhou Lu, Sheng Chen, Yangang Zhang, et al. Bagelvla: Enhancing long-horizon manipulation via interleaved vision-language-action generation. *arXiv preprint arXiv:2602.09849*, 2026.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In Jie Tan, Marc Toussaint, and Kourosh Darvish (eds.), *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 540–562. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/huang23b.html.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025. URL https://arxiv.org/abs/2504.16054.

Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, et al. $\pi_{0.7}$: a steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Yuheng Ji, Huajie Tan, Jiayu Shi, Xiaoshuai Hao, Yuan Zhang, Hengyuan Zhang, Pengwei Wang, Mengdi Zhao, Yao Mu, Pengju An, et al. Robobrain: A unified brain model for robotic manipulation from abstract to concrete. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1724–1734, 2025.

Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. *arXiv preprint arXiv:2512.10935*, 2025.

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, 2024. URL https://arxiv.org/abs/2406.09246.

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*, pp. 10138–10148, 2021.

Baoyu Li, Xinchen Yin, Mengying Lin, Yixin Zhang, and Danfei Xu. Egowam: World action models beyond pixels with in-the-wild egocentric human data. In *Robot World Models*, 2026a.

Hao Li, Ziqin Wang, Zi-han Ding, Shuai Yang, Yilun Chen, Yang Tian, Xiaolin Hu, Tai Wang, Dahua Lin, Feng Zhao, et al. Robointer: A holistic intermediate representation suite towards robotic manipulation. *arXiv preprint arXiv:2602.09973*, 2026b.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026c.

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

Yihang Li, Xuelong Wei, Jingzhou Luo, Yingjing Xiao, Yibo Bai, Guangyuan Zhou, Teng Zou, Chenguang Gui, Jiajun Wen, He Zhang, et al. Egolive: A large-scale egocentric dataset from real-world human tasks. *arXiv preprint arXiv:2604.23570*, 2026d.

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

Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. In *International Conference on Learning Representations*, volume 2025, pp. 29982–30009, 2025.

Songming Liu, Bangguo Li, Kai Ma, Lingxuan Wu, Hengkai Tan, Xiao Ouyang, Hang Su, and Jun Zhu. RDT2: Exploring the scaling limit of UMI data towards zero-shot cross-embodiment generalization. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=ABrWtqsJYp.

Yicheng Liu, Zibin Dong, Baijun Ye, Tianyuan Yuan, Tao Jiang, Anqi Yang, Shicheng Cao, Haonan Liu, Yue Sun, Zihan Guo, et al. G0. 5: One autoregressive stream for robot reasoning and action. *arXiv preprint arXiv:2608.11739*, 2026b.

Yun Liu, Haolin Yang, Xu Si, Ling Liu, Zipeng Li, Yuxiang Zhang, Yebin Liu, and Li Yi. Taco: Benchmarking generalizable bimanual tool-action-object understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21740–21751, 2024.

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, 2025. URL https://arxiv.org/abs/2507.15597.

Hao Luo, Ye Wang, Wanpeng Zhang, Sipeng Zheng, Ziheng Xi, Chaoyi Xu, Haiweng Xu, Haoqi Yuan, Chi Zhang, Yiqing Wang, Yicheng Feng, and Zongqing Lu. Being-h0.5: Scaling human-centric robot learning for cross-embodiment generalization, 2026. URL https://arxiv.org/abs/2601.12993.

Jiangran Lyu, Kai Liu, Xuheng Zhang, Haoran Liao, Yusen Feng, Wenxuan Zhu, Tingrui Shen, Jiayi Chen, Jiazhao Zhang, Yifei Dong, et al. Lda-1b: Scaling latent dynamics action model via universal embodied data ingestion. *arXiv preprint arXiv:2602.12215*, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

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.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. Wilor: End-to-end 3d hand localization and reconstruction in-the-wild. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12242–12254, 2025.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y Zhu, et al. Egoverse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Krishan Rana, Robert Lee, David Pershouse, and Niko Suenderhauf. Imle policy: Fast and sample efficient visuomotor policy learning via implicit maximum likelihood estimation. *arXiv preprint arXiv:2502.12371*, 2025.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. URL http://arxiv.org/abs/1804.02767.

Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)*, 36(6), November 2017.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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

Yingbo Tang, Lingfeng Zhang, Shuyi Zhang, Yinuo Zhao, and Xiaoshuai Hao. Roboafford: A dataset and benchmark for enhancing object and spatial affordance learning in robot manipulation. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 12706–12713, 2025.

Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry. *Advances in Neural Information Processing Systems*, 36:39033–39051, 2023.

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*, pp. 20270–20281, 2023.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17868–17879, 2024.

Luis Wiedmann, Orr Zohar, Amir Mahla, Xiaohan Wang, Rui Li, Thibaud Frere, Leandro von Werra, Aritra Roy Gosthipaty, and Andrés Marafioti. Finevision: Open data is all you need. *arXiv preprint arXiv:2510.17269*, 2025.

Wei Wu, Fan Lu, Yunnan Wang, Shuai Yang, Shi Liu, Fangjing Wang, Qian Zhu, He Sun, Yong Wang, Shuailei Ma, et al. A pragmatic vla foundation model. *arXiv preprint arXiv:2601.18692*, 2026a.

Wei Wu, Fangjing Wang, Fan Lu, He Sun, Shi Liu, Yunnan Wang, Yibin Yan, Yong Wang, Shuailei Ma, Xinyang Wang, et al. From foundation to application: Improving vla models in practice. *arXiv preprint arXiv:2607.06403*, 2026b.

Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, Xuxin Cheng, Ri-Zhao Qiu, Hongxu Yin, Sifei Liu, Song Han, Yao Lu, and Xiaolong Wang. Egovla: Learning vision-language-action models from egocentric human videos, 2025. URL https://arxiv.org/abs/2507.12440.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

Tianyuan Yuan, Zibin Dong, Yicheng Liu, and Hang Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. Robopoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

Kevin Zakka. Mink: Python inverse kinematics based on MuJoCo, February 2026. URL https://github.com/kevinzakka/mink.

Xinyu Zhan, Lixin Yang, Yifei Zhao, Kangrui Mao, Hanlin Xu, Zenan Lin, Kailin Li, and Cewu Lu. Oakink2: A dataset of bimanual hands-object manipulation in complex task completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 445–456, 2024.

Gu Zhang, Qicheng Xu, Haozhe Zhang, Jianhan Ma, Long He, Yiming Bao, Zeyu Ping, Zhecheng Yuan, Chenhao Lu, Chengbo Yuan, et al. Unidex: A robot foundation suite for universal dexterous hand control from egocentric human videos. *arXiv preprint arXiv:2603.22264*, 2026a.

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

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, et al. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026b.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Ruijie Zheng, Dantong Niu, Yuqi Xie, Jing Wang, Mengda Xu, Yunfan Jiang, Fernando Castañeda, Fengyuan Hu, You Liang Tan, Letian Fu, Trevor Darrell, Furong Huang, Yuke Zhu, Danfei Xu, and Linxi Fan. Egoscale: Scaling dexterous manipulation with diverse egocentric human data, 2026. URL https://arxiv.org/abs/2602.16710.

Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, Zhiquan Qi, Yitao Liang, Yuanpei Chen, and Yaodong Yang. A survey on vision-language-action models: An action tokenization perspective, 2025. URL https://arxiv.org/abs/2507.01925.

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, et al. Roborefer: Towards spatial referring with reasoning in vision-language models for robotics. *Advances in Neural Information Processing Systems*, 38:28404–28481, 2026.

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*, pp. 2165–2183. PMLR, 2023.

Appendix

Table of Contents

A Details of EgoSmith **17**

A.1 Implementation Details. 17

A.1.1 Pre-Filtering Heuristics 17

A.1.2 4D Motion Estimation 17

A.1.3 Language Labeling Prompt 18

A.1.4 Post-Filtering Criteria 19

A.2 Curated Dataset Statistics 20

B Details of the Robot Stack **22**

B.1 Implementation Details. 22

B.1.1 Hardware Setup 22

B.1.2 Aligning Robot Data with Egocentric Human Data 22

B.1.3 Hand-Eye Verification and Re-Calibration 22

B.1.4 Language Labeling Prompt 22

B.2 Teleoperation Data Collection 24

C Details of EgoSteer **27**

C.1 VLM Co-Training Data 27

C.2 Implementation Details. 27

D Experimental Details **30**

A DETAILS OF EGOSMITH

This section presents the implementation details of EgoSmith (Appendix A.1) and provides detailed statistics of the curated 9.6K-hour pre-training corpus derived from 12 egocentric human video datasets (Appendix A.2).

A.1 IMPLEMENTATION DETAILS

A.1.1 PRE-FILTERING HEURISTICS

The pre-filtering stage employs frame-wise heuristics to rapidly discard low-quality segments, such as those containing locomotion, excessive head movement, hand absence, occlusion, or others' hand interference. These five scenarios are handled by two specialized gates: a camera gate for motion-related issues, and a hand gate for visibility anomalies. A contiguous segment is pruned only if it contains at least three consecutive invalid frames; isolated failures are retained as they exert negligible impact on subsequent reconstruction.

The camera gate estimates ego-motion using sparse optical flow. For each frame, a 128-point grid is tracked back 15 frames earlier via pyramidal Lucas–Kanade (Bouguet et al., 2001). We fit a similarity transform using RANSAC (Hartley & Zisserman, 2003); the frame passes this gate if the translation of the estimated transform is within 10% of the image's longer dimension.

The hand gate uses YOLO (Redmon & Farhadi, 2018; Potamias et al., 2025) to detect hands, retaining a bounding box as valid only if it satisfies three criteria: (a) confidence ≥ 0.30 to reject false positives; (b) area within $[2\%, 50\%]$ of the image, where the 50% upper bound excludes hands abnormally close to the lens, and the 2% lower bound is calibrated on Egocentric-10K/100K (AI, 2025b;a) subsets, where we manually labeled small bounding boxes to find a threshold that filters out most other people's hands while retaining the operator's; and (c) spatially intersecting with the lower-central region (normalized $[0.075, 0.925]$ horizontally, $[0.075, 1.0]$ vertically). The gate requires ≥ 2 valid detections per frame, which naturally filters out hand absence, occlusion, or cases where only other people's hands are present, while preserving clear bimanual manipulation.

Together, the two gates yield video segments characterized by stable camera motion and clearly visible bimanual interactions.

A.1.2 4D MOTION ESTIMATION

In stage 2, we reconstruct hand motions in a unified, metric world-space coordinate system. Within this pipeline, we adopt the ViT module from HaWoR as an off-the-shelf camera-frame hand reconstructor. Specifically, hands are detected and cropped across frames, and the ViT processes these inputs in temporal windows to regress the frame-wise MANO (Romero et al., 2017) pose parameters $\boldsymbol{\theta}_t \in \mathbb{R}^{51}$, shape parameters $\boldsymbol{\beta}_t \in \mathbb{R}^{10}$, and the camera-relative root translation $\mathbf{t}_t \in \mathbb{R}^3$. Although this camera-space hand reconstruction is highly reliable, placing these reconstructions in world space with accurate physical dimensions requires a robust, metric camera trajectory. This is a primary limitation of the original HaWoR pipeline, as its dependency on DROID-SLAM can accumulate drift under egocentric conditions with rapid head movements or textureless environments. To address this, we design a robust pipeline to recover a metric, temporally consistent camera trajectory in world space, onto which the hand reconstructions are subsequently mapped.

We replace DROID-SLAM with DPVO (Teed et al., 2023) to estimate the camera trajectory. DPVO is more robust in long-range egocentric scenarios and incurs much lower computational cost, outputting up-to-scale camera poses $\hat{\mathbf{T}}_t = (\mathbf{R}_t, \hat{\mathbf{p}}_t) \in SE(3)$, where the hat notation $\hat{\cdot}$ denotes up-to-scale quantities. We then anchor the trajectory to the physical scale using metric depth estimates from Any4D (Karhade et al., 2025) as a reference. To ensure temporal coherence across the entire sequence, we perform a cross-chunk alignment on the local Any4D depth windows, yielding a temporally consistent metric depth sequence.

To recover the metric scale factor s for DPVO trajectory, we compute the median ratio of the aligned Any4D depth $\mathbf{D}_t^{\text{Any4D}}$ to the DPVO depth $\hat{\mathbf{D}}_t^{\text{DPVO}}$ over the background pixels across all keyframes:

$$s = \text{median}_{t,(u,v)\in\mathcal{B}_t} \frac{\mathbf{D}_t^{\text{Any4D}}(u,v)}{\hat{\mathbf{D}}_t^{\text{DPVO}}(u,v)}, \tag{1}$$

where $\mathcal{B}_t$ is the valid background region in frame t, defined by excluding the hand regions projected from our reconstructed 3D hand mesh. We then calibrate the entire trajectory via $\mathbf{p}_t = s\hat{\mathbf{p}}_t$, yielding a metric camera trajectory.

Given the metric camera poses, we transform the camera-space hand vertices and joints, denoted generally as 3D coordinates $\mathbf{x}_t^{\text{cam}} \in \mathbb{R}^3$, into world space via $\mathbf{x}_t^{\text{world}} = \mathbf{R}_t^\top(\mathbf{x}_t^{\text{cam}} - \mathbf{p}_t)$. The output for each video segment consists of the world-space bimanual states and actions, frame-wise camera intrinsics and extrinsics $(\mathbf{K}_t, \mathbf{T}_t)$, MANO parameters, and the Any4D metric scene depth.

Concerning efficiency, stage 2 represents the primary computational bottleneck of our entire pipeline, with most of its overhead arising from the two core components of 4D motion estimation: camera trajectory estimation and hand motion reconstruction. For camera tracking, replacing the dense DROID-SLAM with the substantially more lightweight DPVO significantly reduces the associated computational cost. Hand reconstruction, meanwhile, involves repeated ViT inference over temporal windows, for which the original HaWoR processes only a single 16-frame window at a time. We therefore group multiple windows into a single batch for parallel forward passes and asynchronously overlap CPU-based frame decoding and cropping with GPU-based model inference to minimize GPU idling. By improving the efficiency of both major components, these optimizations collectively accelerate the complete 4D motion estimation stage, achieves an overall speedup of approximately 8.3× compared to HaWoR.

For completeness, we provide detailed definitions of the four metrics used in Table 1 to jointly evaluate camera-trajectory and world-space hand-motion accuracy (all errors are reported in mm):

- **Relative Pose Error (RPE)**: Measures local camera tracking drift over a fixed temporal interval without global trajectory alignment, making it sensitive to local motion errors and metric-scale inaccuracies.
- **Absolute Trajectory Error (ATE)**: Measures global camera trajectory error after aligning the estimation to the ground truth with a single $\mathrm{Sim}(3)$ transformation, thereby evaluating trajectory structure and long-term drift while factoring out global scale and coordinate-frame differences.
- **World-Aligned Mean Per Joint Position Error (WA-MPJPE)**: Measures world-space hand joint error after aligning the predicted and ground-truth hand trajectories with a single $\mathrm{Sim}(3)$ transformation over each 100-frame segment. Unlike per-frame alignment, this preserves temporal trajectory consistency and penalizes relative drift within the segment.
- **World Mean Per Joint Position Error (W-MPJPE)**: Measures world-space hand joint error after rigidly aligning only the first frame of each 100-frame segment with an $\mathrm{SE}(3)$ transformation. It therefore preserves the predicted metric scale and accumulates subsequent scale, translation, rotation, and temporal drift errors, providing the strictest evaluation of world-space hand tracking.

A.1.3 Language Labeling Prompt

Below, we present the prompt template designed for Qwen3.5-VL-Plus (Qwen Team, 2026) to generate multi-granularity language annotations for egocentric human videos.

Egocentric Human Video Language Labeling Prompt

```
# Video Annotation Task: General Egocentric Hand Action Description

## Objective
Provide a comprehensive 5-level language description for pre-cropped egocentric videos of
     human hand-object interaction. These videos may show daily activities, tool use,
    object handling, cooking, cleaning, repair, assembly, organization, inspection, or
    other practical manipulation tasks.

## Context
- **Perspective:** Ego-view focused on the hands and manipulated objects.
- **Purpose:** Training data for robotic systems.
- **Core Focus:** The visible hand action, the manipulated object, contact points, grasp
    type, spatial relationships, and the physical sequence needed to complete the task.

---

## Task Instructions

Analyze the video clip and perform the following steps:
```

```
### Step 1: Content Filtering (Hand-Action Status Check)
- **Mark "status": "Invalid"** if the video shows:
    - Walking, camera transition, or scene scanning with no meaningful hand-object
     interaction.
    - Passive observation with no active manipulation.
    - Hands resting, hanging, or only briefly entering the frame without acting on an
     object.
    - Non-manipulation activities such as talking, reading, waiting, or looking around.
    - A task where the main hand action cannot be identified because of severe occlusion,
     blur, or ambiguity.
- **Mark "status": "Valid"** if the video shows active hand-object manipulation, such as
     picking, placing, opening, closing, pouring, wiping, folding, pressing, turning,
     cutting, fastening, arranging, inserting, removing, scanning a barcode/card/object,
     or operating an object.
- If the main action is identifiable but partially occluded, mark `"status":"Valid"`.

### Step 2: Multi-Level Hand Action Instructions (For "Valid" only)

- **Level 1 (Verb + Object):** Core task. Max 5 words. Example: "Open the drawer."
- **Level 2 (Gist):** Concise summary of the hand action. Max 15 words.
- **Level 3 (Object-Centric):** Describe the manipulated object, relevant parts, state,
     and spatial features. Max 30 words.
- **Level 4 (Hand-Centric):** Specify left/right hand roles, grasp style, contact points,
      and coordination. Max 50 words.
- **Level 5 (Dense Sequence):** Step-by-step physical breakdown. Max 100 words.
    - Use spatial anchors such as "near the left edge," "above the bowl," "inside the
     slot," or "against the surface."
    - Describe motion trajectories such as "lift upward," "slide forward," "rotate
     clockwise," "press downward," or "pull toward the body."
    - Describe outcome state such as "fully seated," "opened," "aligned," "placed flat,"
     "wiped clean," or "released."

---

## Strict Formatting & Quality Requirements

- **Verb-first imperative:** Start with an action verb. **NO subjects**.
- **Definite object references:** Use "the" for visible objects; avoid "a" or "an" in the
      generated instructions.
- **No transitional words:** Omit "then", "next", "afterwards".
- **Action Precision:** Use specific physical verbs such as "Grip," "Lift," "Place," "
     Open," "Close," "Pour," "Fold," "Wipe," "Press," "Turn," "Insert," "Remove," "Align
     ," "Slide," "Scoop," "Cut," "Peel," "Tighten," or "Release" when possible.
- **Avoid over-specialization:** Do not infer hidden intent, brand names, object
     identities, or materials unless clearly visible.

---

## Output Format
Return a single JSON object. No markdown code fences. No extra text.
For `"status":"Invalid"`, return empty strings for all five levels.

{
  "status": "Valid/Invalid",
  "language_instructions": {
    "level1": "<verb and object>",
    "level2": "<concise summary>",
    "level3": "<object-focused description>",
    "level4": "<hand/object interaction details>",
    "level5": "<dense physical step-by-step>"
  }
}
```

A.1.4 POST-FILTERING CRITERIA

In this stage, we perform quality control on the reconstructed outputs to filter out reconstruction anomalies and problematic segments, ensuring overall data quality. This evaluation is conducted from coarse to fine across three granularities: entire episodes, chunk windows, and adjacent frames.

At the episode level, we characterize overall camera motion using the mean frame-to-frame changes in the translation and rotation components of the camera extrinsics. Specifically, for each episode, we average the magnitudes of the consecutive-frame translation changes and rotation-matrix differences. We then compare these two episode-level motion statistics against their respective distributions over the episodes in the current dataset processing batch. Because typical camera mo-

tion varies across capture devices, scenes, and manipulation styles, we use dataset-dependent IQR fences rather than fixed thresholds. An episode is rejected if either statistic falls outside $[Q_1 - 2.5\,\mathrm{IQR},\ Q_3 + 2.5\,\mathrm{IQR}]$. This criterion targets episodes with atypical camera trajectories caused by tracking failures or unusually large head and body motion.

Chunk-level checks evaluate whether hands fall within physically reasonable spatial boundaries in a standardized egocentric coordinate frame. Directly comparing absolute hand coordinates is problematic, since they are coupled with camera and body movements. Specifically, within a sliding window spanning approximately the past 5 seconds and the future 30 frames, we transform all hand states and actions into the current camera frame. Under this canonical system, wrist positions are defined relative to the camera, and finger joints relative to the wrist. Within this coordinate system, we evaluate both distributional outliers and absolute physical limits. First, outliers in wrist translation/rotation and finger positions are identified using the same IQR criterion based on the respective dataset's distribution. Second, we enforce a universal physical ceiling of 1.5 meters on each coordinate axis for the hands, as a human hand cannot physically reach further than this distance from the head. If any sliding chunk window within an episode violates either the IQR outlier threshold or the 1.5-meter physical limit, the entire episode is discarded.

Frame-level checks identify sudden jumps between adjacent frames. We compute the frame-to-frame changes in camera translation and rotation, wrist translation and rotation, and finger translation. Unlike the previous levels, we do not rely on dataset-specific distributions here, as the physical speed of human hands and heads has a universal limit. Any movement exceeding this limit is attributed to reconstruction artifacts rather than valid motion. We therefore apply fixed physical thresholds: camera translation ≤ 0.20 m/frame, wrist and finger translation ≤ 0.30 m/frame, camera rotation $\leq 28^\circ$/frame, and wrist rotation $\leq 41^\circ$/frame. An episode is discarded if any of its frames violate these thresholds.

Collectively, these three levels of checks filter out problematic segments caused by head tracking drift, inaccurate motion reconstruction, and motion discontinuities.

A.2 Curated Dataset Statistics

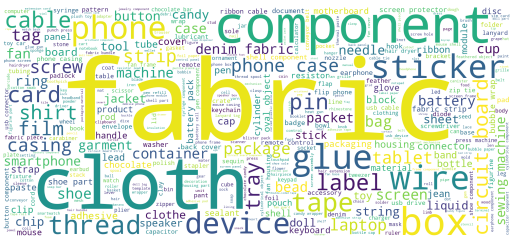
The final pre-training corpus is constructed by utilizing EgoSmith to process 12 raw egocentric datasets, ultimately yielding a standardized, 9.6K-hour dataset. This curated corpus comprises world-space bimanual states and actions, frame-wise camera intrinsics and extrinsics, metric scene depth, and coarse-to-fine language annotations. In this section, we detail the scale, source composition, and semantic diversity of this dataset. The primary objective is not merely scaling up dataset volume; instead, the core advantage lies in data quality. Every sample is richly annotated and filtered through a rigorous quality-control pipeline, guaranteeing high-quality, modality-aligned manipulation knowledge to assist the model in learning steerable dexterous manipulation. The dataset further offers broad coverage of manipulation tasks, objects, and multi-granularity language descriptions.

Scale and Source Composition. Figure 8a presents the duration contribution and proportion of each source dataset, while also indicating whether each annotation is natively provided or reconstructed by our pipeline. The source datasets are highly complementary, covering a wide range of devices, scenes, and manipulation styles. Although a small number of large-scale collections dominate the total duration, the breadth of sources contributes substantial diversity.

Task and Semantic Diversity. To characterize the manipulation span of our dataset, we extract (verb, object) tuples from the L1 verb–object annotations and analyze the distributions of action verbs and manipulated objects. The dataset covers 8,969 distinct object nouns and 623 action verbs (Figures 8b and 8c). Figure 8d illustrates the top 50 most frequent "verb + object" atomic tasks. Common tasks concentrate on fundamental manipulation skills, aligning with the natural distribution of daily hand–object interactions. At the same time, the distribution exhibits a prominent long tail: a large number of low-frequency yet semantically diverse tasks and objects provides broad coverage, preventing the dataset from being biased toward a few dominant actions. This combination of common fundamental skills, long-tail task diversity, and multi-granularity language annotations provides downstream models with abundant training samples, broad task coverage, and rich dexterous manipulation knowledge.

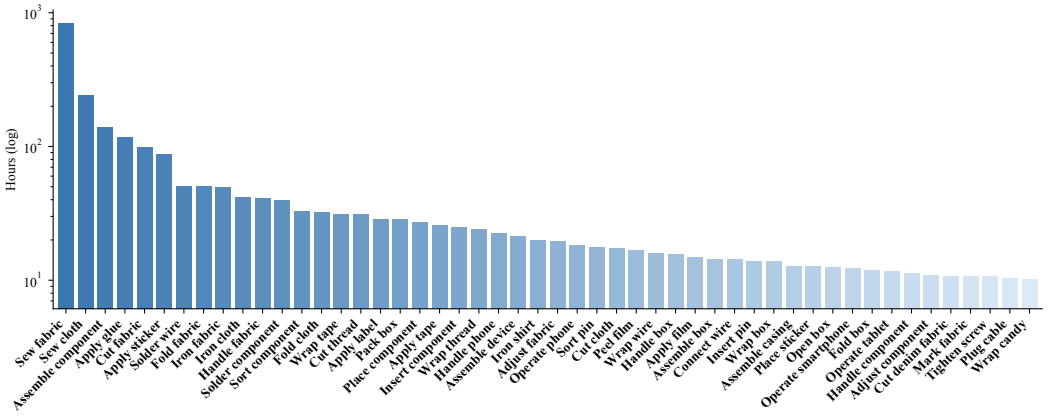
Dataset	Hours	Percentage (%)	Episodes	Hand	Depth	Camera	Language
Egocentric-100K (AI, 2025a)	8,049	83.8	1,795,731	✓	✓	✓	✓
EgoVerse (Punamiya et al., 2026)	690	7.2	35,175		✓		✓
EgoDex (Hoque et al., 2025)	370	3.9	147,588		✓		✓
Egocentric-10K (AI, 2025b)	288	3.0	194,915	✓	✓	✓	✓
Ego4D (Grauman et al., 2022)	138	1.4	74,505	✓	✓	✓	✓
Epic-Kitchens (Damen et al., 2018)	49	0.5	26,454	✓	✓	✓	✓
HoloAssist (Wang et al., 2023)	11.5	0.1	11,426		✓		✓
HOT3D (Banerjee et al., 2025)	4.5	0.05	1,105		✓		
TACO (Liu et al., 2024)	3.0	0.03	1,558				✓
OakInk-v2 (Zhan et al., 2024)	1.7	0.02	891		✓		
H2O (Kwon et al., 2021)	1.0	0.01	935				
FPHA (Garcia-Hernando et al., 2018)	0.5	0.01	578	✓	✓		✓
Total	**9,606**	100	**2,290,861**				

(a) Dataset composition and annotation sources



(b) Word cloud of manipulated objects

(c) Word cloud of action verbs



(d) Top-50 "verb + object" atomic tasks

Figure 8: Curated dataset statistics. (a) Source composition of the 12 egocentric datasets, detailing duration, percentage, episode count, and per-annotation origin. Checkmarks denote annotations generated by our pipeline, while blank entries indicate natively provided annotations. For FPHA, the checkmark under *Hand* specifically denotes the additionally annotated second hand. (b)–(d) Task and semantic diversity: word clouds of the manipulated objects and action verbs, and the most frequent verb–object atomic tasks, exhibiting long-tailed coverage over diverse manipulation skills.

B Details of the Robot Stack

This section details the implementation of the Robot Stack (Appendix B.1), alongside the collection protocols and statistics of the 187-hour real-robot dataset (Appendix B.2).

B.1 Implementation Details

B.1.1 Hardware Setup

The two physical embodiments utilized in our work are illustrated in Figure 3. The primary platform, referred to as the RealMan embodiment, consists of two 7-DoF RealMan RM75-6F robotic arms and two 6-DoF Ruiyan RY-H2 dexterous hands. It is equipped with one head-mounted and one chest-mounted Intel RealSense D455 camera, providing dual egocentric viewpoints to ensure a comprehensive visual field. The AgiBot G1 embodiment is adapted from an AgiBot G1 humanoid robot by replacing its default end-effectors with two 6-DoF Ruiyan RY-H2 hands. It features one head-mounted Intel RealSense D455 camera and two wrist cameras, the latter of which are unused. During experiments, the robot's neck, waist, and mobile base are kept fixed. Both platforms conduct tabletop manipulation on an operational table. The RealMan platform serves as the primary embodiment, on which the entire 187-hour dataset was collected. All experiments are conducted on this setup, with the sole exception of the few-shot adaptation experiment performed on the AgiBot G1.

For both platforms, the human wrist tracker, arm kinematic solvers, and joint control ROS 2 nodes operate at 100 Hz, while the glove, hand solvers, and control ROS 2 nodes run at 80 Hz. This high-frequency loop ensures near-zero latency, enabling intuitive bimanual teleoperation and the collection of fine-grained manipulation demonstrations. The cameras capture frames at 30 Hz. Data is recorded at these native frequencies and subsequently resampled to 30 Hz for training.

B.1.2 Aligning Robot Data with Egocentric Human Data

Robot data collection aims to ground the human manipulation priors learned from large-scale pre-training onto the target physical embodiment. To preserve and transfer this pre-trained knowledge, minimizing the domain gap between robot and human data is essential. Because the robotic palm is slightly longer than a human palm, we translate the robot's wrist coordinate frame axially forward. This alignment ensures that the scale from the wrist to the fingertips matches human anatomical proportions. When converting raw robot data into training data, this coordinate transformation is applied alongside hand-eye calibration to map hand actions into the camera coordinate frame. Correspondingly, during real-time policy inference, the control stack executes the inverse transformation and hand-eye mapping to project predicted actions back to the robot's physical coordinate space.

B.1.3 Hand-Eye Verification and Re-Calibration

In practice, hand-eye calibration is prone to drift due to mechanical maintenance, wear, or collisions, with such discrepancies sometimes only identified *post*-acquisition. To ensure data quality, we introduce an offline verification and automated recalibration pipeline. RGB-D images from the camera are projected into a 3D point cloud, and the robot's 3D mesh is rendered in the same space based on the hand-eye calibration, utilizing their spatial overlap for validation. Upon detecting misalignment, our pipeline runs FoundationPose (Wen et al., 2024) on the point cloud to estimate the robot hand's 6D pose based on its URDF. Through FK, the calibration matrix is back-calculated for each frame. Averaging these matrices across frames and removing outliers robustly recovers the calibration parameters, preventing data degradation.

B.1.4 Language Labeling Prompt

Below, we present the prompt template designed for Qwen3-VL-Flash (Bai et al., 2025) to generate multi-granularity language annotations for teleoperated demonstrations.

> **Robot Data Language Labeling Prompt**
>
> ```
> **CRITICAL VISUAL CONTEXT & PRIOR (READ CAREFULLY):**
> ```

```
You are observing two synchronized egocentric videos (Head View: top, Chest View: bottom)
      of an agent performing a manipulation task. This output will be used as language
     instructions for robotic training.
1. **The Agent's Hand:** The moving entity is the agent's bare end-effector. It generally
      has a grey base and black fingers.
2. **COLORED TIPS WARNING:** The tips/pads of the fingers often have GREEN, ORANGE, or
     RED tape/markers on them. THESE ARE PART OF THE FINGERS. They are NOT separate tools
     .
3. **EMPTY-HANDED PRIOR:** The agent is operating empty-handed. NEVER describe the agent
     as holding or using a 'green-tipped tool', 'hot knife', 'pliers', or any handheld
     instrument.
4. **ABSOLUTE GROUND TRUTH (TASK ALIGNMENT):** The specific task is **[{task_name}]**.
     This task name is your absolute ground truth for interpreting WHAT is being
     manipulated (Objects) and HOW it is being manipulated (Verbs). You MUST use the
     exact nouns implied by the task name.
5. **GRAMMAR:** Your description must be in **simple present tense**. Write in fluent
     English, avoid awkward phrasing.

**OBJECTIVE:**
Describe the agent's actions (focusing strictly on hand-object interactions) in **simple
     present tense** by integrating information from both views into a single, unified
     description at three levels of detail. **Since this is for robot training, you MUST
     completely ignore all task-irrelevant items.**

"**CONSTRAINTS:**
1. **Unified Description:** Provide ONE consolidated set of descriptions.
2. **Levels of Detail:**
   - **Level 1 (Gist):** A concise summary of the main action. Should always be a verb+
     noun phrase (i.e Ring a bell) (<20 words).
   - **Level 2 (Descriptive):** Main action + features and spatial layout of the **
     ACTIVELY MANIPULATED OBJECTS ONLY** (<40 words). **DO NOT list or describe any
     stationary background clutter.** In your description, list which hand is performing
     the action.
   - **Level 3 (Sequential):** The step-by-step temporal flow of key functional actions
     (<70 words). Include essential phases (reach -> manipulate -> release) but OMIT
     trivial micro-adjustments or hovering. List which hand is performing each action.
3. **Zero Subjects (Strict):** **Start every single sentence directly with a verb** (e.g
     ., 'Reach for...', 'Grasp...'). DO NOT use subjects (e.g., 'The person', 'The robot
     ', 'The hand', 'It').
4. **Strict Focus on Interaction (NO CLUTTER):** Focus ONLY on the objects being actively
      touched, moved, or interacted with (and their immediate targets/receptacles). **
     Completely IGNORE all irrelevant background items** (e.g., wipes, boxes, tubes,
     stands that are not part of the task). Never write phrases like 'other items remain
     unchanged' or 'in the background'.
5. **Vocabulary Restrictions:**
   - DO NOT output words like 'robot', 'mechanical arm', 'gripper', 'human', or 'finger'.
   - DO NOT output colors of the agent's hand/tips.
6. **Task Vocabulary (Verbs & Nouns):** Your choice of verbs AND target nouns must
     strictly align with the task **[{task_name}]**.
7. **Action Logic & Validation:** Focus on the actual state changes of the objects.
     Verify actual contact using both views.
8. **Object Disambiguation:** Use spatial descriptors (e.g., 'the topmost card') ONLY for
      task-relevant items to distinguish them from each other.
9. **Tense:** Use **Simple Present** tense (e.g., 'reach', 'grasp', 'slide').
10. **Spatial Description:** Use the camera frame as the reference.
   - Use **'upper', 'middle', 'lower'** to describe distance of objects on table. For
     example, 'the upper left of the table' refers to the far side of the table, and 'the
      lower left' refers to the near side.
   - Use 'left' 'right' to describe horizontal relationships\n"
   - Use 'on', 'on top of', 'above', 'below' etc. to describe vertical relationships.


**OUTPUT FORMAT:**
1. [Level 1 Description]
2. [Level 2 Description]
3. [Level 3 Description]


**EXAMPLE (If Task Name is 'Draw_cards'):**
1. Draw playing cards.
2. Slide the top cards from a central deck with left hand to draw them to the lower part
     of the table.
3. Reach toward the central deck with left hand, press down on the topmost card, and
     slide it backward. Return to the deck, press on the next card, and slide it backward
      to complete the draw."
```

B.2 TELEOPERATION DATA COLLECTION

Although pre-training on egocentric human videos equips the model with rich dexterous manipulation priors, direct physical deployment is prevented by the embodiment gap across visual appearance, dynamics, and kinematics. Furthermore, because these pre-training labels are generated by automated depth and hand pose estimation models, their accuracy is constrained by current model capabilities, potentially introducing systematic biases into the pre-trained policy. Therefore, collecting real-robot teleoperation data aims to correct and ground these priors onto the target embodiment with high sample efficiency. Additionally, due to kinematic constraints on robotic degrees of freedom, the robot cannot perfectly replicate all fine-grained human hand movements. To achieve steerable manipulation, we must collect a highly diverse range of free-form tasks to maximize the coverage of basic manipulation primitives, guiding the model to efficiently transfer cross-domain knowledge and fostering robust compositional generalization and multi-task instruction following.

Based on these considerations, within the kinematic limits of RealMan's degrees of freedom, we design 193 semantically distinct dexterous manipulation tasks. For each task, approximately 300 randomized, diverse demonstration trajectories, totaling around 1 hour, are collected under cluttered scenarios, constructing a high-quality real-robot dataset of 187 hours and 55K trajectories.

These 193 tasks are classified into two major categories:

- **Common Tasks** (56 tasks): Everyday manipulations that are readily achievable within the current robot hardware configuration and sensor limits, yielding high teleoperation success rates.
- **Long-Tail Tasks** (137 tasks): Infrequent and physically challenging manipulations, such as contact-sensitive operations lacking tactile feedback, with lower collection success rates, primarily designed to ensure comprehensive semantic coverage of the dexterous manipulation space.

Based on motion and physical interactions, all tasks are further categorized into seven classes:

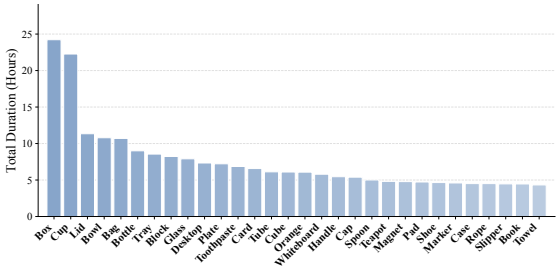
- **PnP-Easy**: Single-step tabletop pick-and-place where objects are easily graspable and the placement space is open.
- **PnP-Medium**: Non-planar or 3D spatial pick-and-place that requires precise operations related to containers, demanding higher control accuracy and spatial perception, such as "*put tennis ball into ball holder*".
- **PnP-Hard**: Multi-step or high-precision pick-and-place sequences, such as "*stack paper cups*".
- **Non-prehensile**: Non-grasping actions, including pushing, pulling, and pressing.
- **Reorient**: Actions involving rotation and reorientation, such as "*pour water*" or "*flip paper cups*".
- **Bimanual**: Bimanual tasks requiring high synchronization and spatial coordination between arms and hands, such as "*plug cable into charger*".
- **Contact-rich**: Operations involving frequent and complex physical contact interactions, requiring physical understanding, such as "*wipe whiteboard*".

Throughout data collection, strict requirements are enforced on the randomness, diversity, and quality of each trajectory. Specifically, the tabletop features cluttered, unstructured scenes rather than pre-arranged, simplified environments, avoiding task identification from visual inputs alone. This design forces the model to deeply align language instructions with physical actions, encouraging it to learn general task semantics and execution goals rather than memorizing demonstration trajectories. Furthermore, given the highly randomized object configurations, operators are instructed to perform teleoperation in a natural, human-like manner. Consequently, different demonstrations of the same task exhibit substantial variations in execution trajectories, thereby covering a broader distribution of the mapping from human priors to the robot's action space.

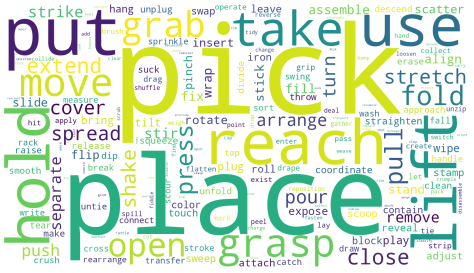
As illustrated in Figure 9, we analyze the 187-hour real-robot dataset across several key dimensions, including the verb and noun word clouds with their top 30 frequency distributions, the task duration breakdown across seven manipulation categories, and the detailed duration statistics for the 56 common and 137 long-tail tasks. These statistics reveal a diverse vocabulary of actions and manipulated objects, alongside a balanced distribution across both manipulation categories and task durations. Encompassing rich semantic concepts and manipulation primitives, this dataset effectively grounds pre-trained human manipulation priors onto the RealMan embodiment. Additionally, Figure 10 showcases the dual-view image sequences and language annotations of a teleoperation trajectory, while Figure 11 visualizes sample tasks across each category.



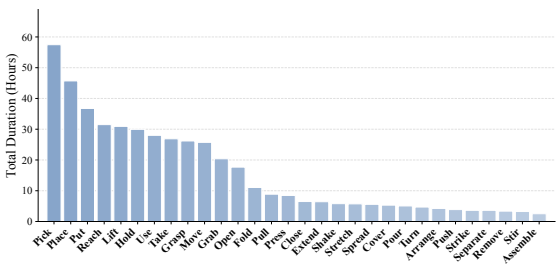
(a) Word cloud of nouns in annotation



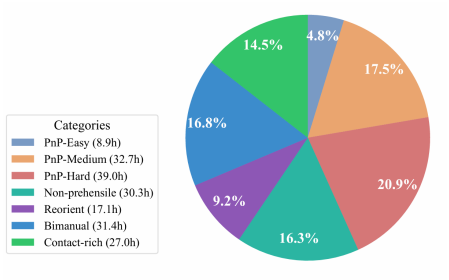
(b) Top 30 noun frequencies



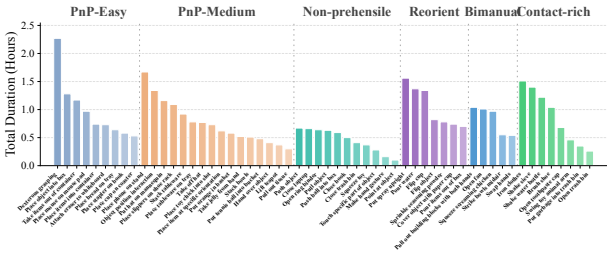
(c) Word cloud of verbs in annotation



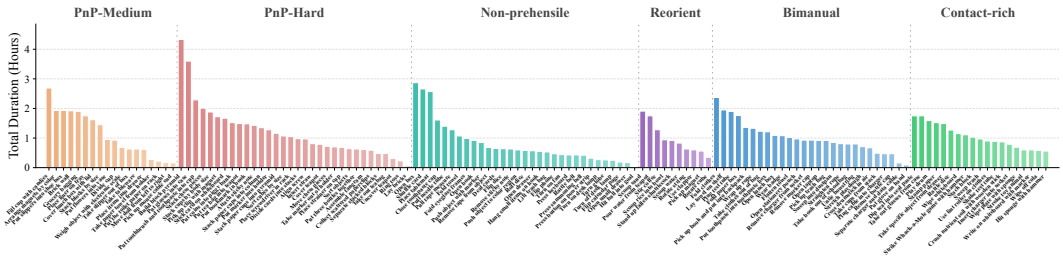
(d) Top 30 verb frequencies



(e) Task duration by seven categories



(f) Common task duration statistics



(g) Long-tail task duration statistics

Figure 9: Teleoperation dataset statistics. (1) Figure 9a - Figure 9d: Word clouds and top-30 frequency distributions of nouns and verbs in language annotations. (2) Figure 9e: Task duration breakdown by seven manipulation categories (PnP-Easy/Medium/Hard, non-prehensile, reorient, bimanual, contact-rich). (3) Figure 9f and Figure 9g: Detailed duration statistics for common tasks (56 tasks) and long-tail tasks (137 tasks).

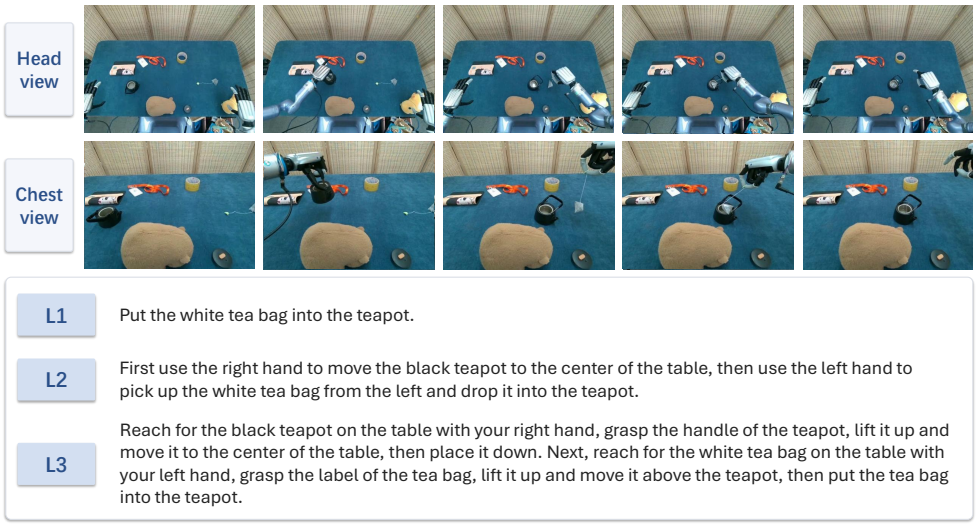


Figure 10: Dual-view image sequence and three-level language annotations of a representative trajectory. The frames illustrate a cluttered, randomized setup where the operator executes the task in a natural, human-like manner. The hierarchical language annotations describe the manipulation process from coarse to fine, assisting the model in aligning free-form instructions with physical actions.

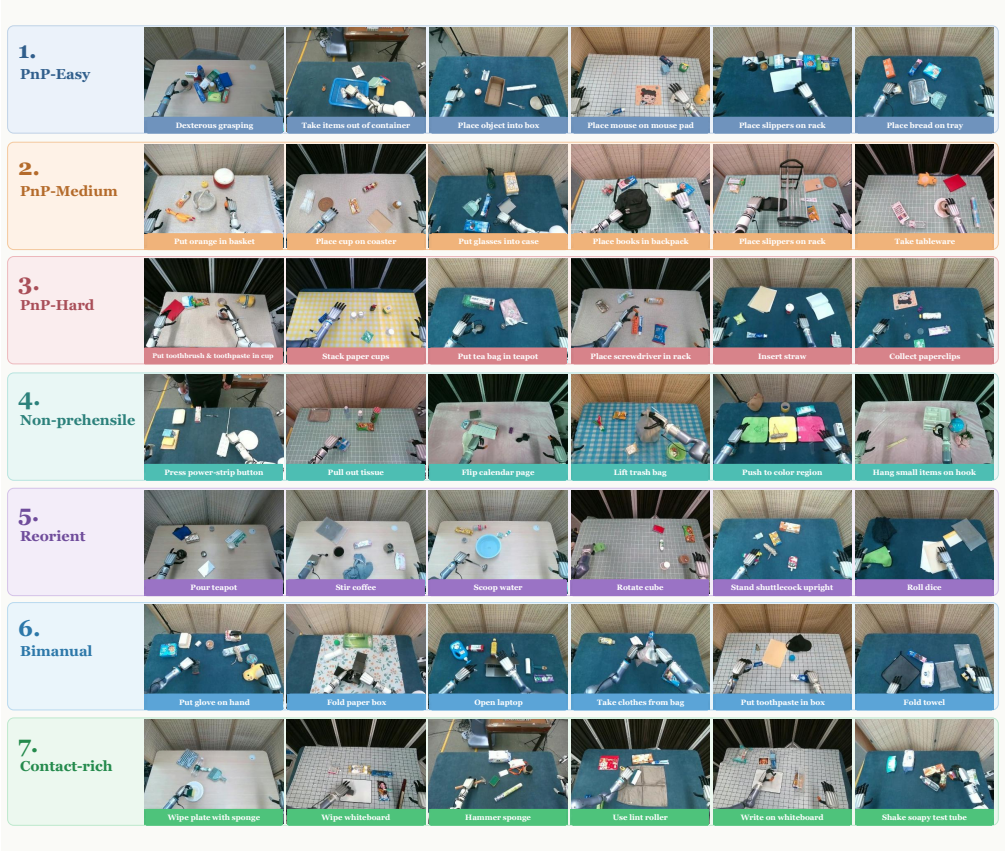


Figure 11: Representative task examples across seven manipulation categories. These diverse tasks are executed in cluttered, randomized scenarios using natural, human-like manipulations, covering a wide range of action semantics and manipulation primitives to facilitate the efficient grounding of pre-trained egocentric human priors onto the physical robot.

Table 3: VLM datasets used for EgoSteer co-training. This table details the sample sizes and the multi-modal and embodied knowledge domains covered by each dataset. This co-training mixture integrates general vision-language knowledge with interaction understanding and spatial-geometric priors, preserving the model's inherent world knowledge while facilitating its ability to follow free-form instructions to generalize across diverse manipulation tasks.

Dataset	Samples	Captioning	Visual Question Answering	Multiple-Choice Question	Pointing	Bounding Box	Affordance	Trajectory	Spatial	Planning
FineVision (Wiedmann et al., 2025)	3.5M	✓	✓	✓		✓				
RefSpatial (Zhou et al., 2026)	2.5M		✓	✓	✓				✓	
RoboInter-VQA (Li et al., 2026b)	1.6M		✓	✓	✓	✓		✓	✓	✓
RoboPoint (Yuan et al., 2024)	1.3M		✓		✓	✓			✓	
RoboAfford (Tang et al., 2025)	765K		✓		✓	✓	✓			
Robo2VLM (Chen et al., 2025)	678K		✓	✓						
ShareRobot (Ji et al., 2025)	13K		✓			✓	✓	✓		✓

C Details of EgoSteer

This section first introduces the VLM datasets utilized for co-training with VLA data, *i.e.* egocentric human videos and real-robot data, to preserve EgoSteer's vision-language knowledge and ensure generalization (Appendix C.1). We then present implementation details of EgoSteer (Appendix C.2).

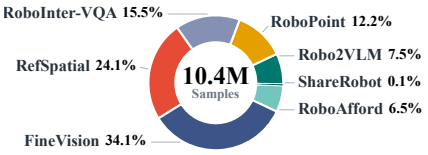


Figure 12: Statistical distribution of samples across VLM co-training datasets.

C.1 VLM Co-Training Data

To preserve general vision-language reasoning capabilities while simultaneously cultivating robust robotic task comprehension, a 10.4M-sample VLM co-training mixture is curated across seven datasets, ranging from open-world perception to embodied interaction grounding, to co-train EgoSteer. This mixture comprises four categories of data:

- **General VLM Pre-Training**: Incorporates FineVision (Wiedmann et al., 2025) to prevent the catastrophic forgetting of open-world semantic concepts and preserve general visual-language reasoning.
- **Spatial Grounding**: Utilizes RefSpatial (Zhou et al., 2026) and RoboPoint (Yuan et al., 2024) for multi-step spatial referring, 2D visual grounding, and precise coordinate localization.
- **Embodied QA**: Integrates RoboInter-VQA (Li et al., 2026b), Robo2VLM (Chen et al., 2025), and ShareRobot (Ji et al., 2025) to support embodied question answering and temporal reasoning. This preserves the model's capabilities in high-level task planning and causal scene understanding.
- **Affordance Perception**: Adopts RoboAfford (Tang et al., 2025) for fine-grained manipulation affordance prediction and spatial interaction grounding.

The sample distribution of these datasets is illustrated in Figure 12, with their covered multimodal, embodied knowledge domains detailed in Table 3. We standardize these datasets to comply with the conversational input format of Qwen3-VL. Specifically, we normalize 2D bounding box and point coordinates to Qwen3-VL's native $[0, 1000]$ scale and adopt its standard representation formats. To ensure training stability, we only utilize single-image samples and exclude those with context lengths exceeding our maximum budget. By co-training on this VLM mixture, EgoSteer preserves open-world vision-language understanding and reasoning while enhancing its comprehension of robotics-specific tasks, thereby assisting the policy in following open-ended instructions and generalizing to novel manipulation scenarios.

C.2 Implementation Details

Backbone Input Scheme. EgoSteer treats the image observation history as a temporal video sequence to leverage the native video-processing capabilities of the Qwen3-VL-2B backbone. In practice, this history is downsampled to 6 frames at 1 FPS, covering a 5 s window, with proprioceptive states sampled at the corresponding timestamps. Language instructions and camera intrinsics are formatted as textual inputs, while the proprioceptive state history is encoded by a two-layer MLP and injected as continuous tokens. Because this proprioceptive history correlates strongly with tar-

get actions, the model is susceptible to shortcut learning, tending to ignore visual inputs and task instructions in favor of proprioception. To mitigate this, each frame of the proprioceptive history is replaced by a learnable mask token with a 75% probability during training, forcing the model to attend to the full multimodal context. Additionally, when utilizing dual-camera inputs during real-robot post-training, the chest-camera sequence is randomly dropped with a 50% probability to prevent over-reliance on chest-view observations.

Action Expert. The action expert is built on a DiT architecture, comprising 14 layers with a hidden dimension of 1024 and an intermediate size of 2816. It features 8 attention heads, each with a head dimension $d_{\text{head}} = 128$, totaling approximately 300 M parameters.

The action expert operates on continuous action chunks of length $h = 32$ at a frequency of 30 Hz. Consistent with the VLM backbone, we apply Interleaved MRoPE for positional encoding. During pre-training, we set the delay $d = 0$ to disable training-time RTC, maximizing action supervision signals and learning rich human manipulation priors. During real-robot post-training, the simulated delay d is uniformly sampled as $d \sim \mathcal{U}([0, 5])$ to accommodate varying inference latencies during deployment. For the flow matching timestep $\eta \in [0, 1]$, the prefix action $\mathbf{a}_{\text{pre}}$ is assigned $\eta = 1$, representing no noise and excluding it from the loss computation. Conversely, the target suffix action $\mathbf{a}_{\text{suf}}$ has its timestep sampled from the probability distribution $P(\eta) = \text{Beta}(\frac{s-\eta}{s}; 1.5, 1)$ with $s = 0.999$ (Black et al., 2026). This timestep η is sinusoidally encoded, mapped through a two-layer MLP, and injected into the action expert via AdaLN-Zero.

Layer ℓ of the action expert employs a joint attention mechanism, attending to both itself and the VLM backbone's key-value cache from layer $f(\ell) = 2\ell$ after a linear projection. Specifically, let the query, key, and value of the m-th attention head in layer ℓ of the action expert be $\mathbf{Q}^{\text{AE}}_{\ell,m}, \mathbf{K}^{\text{AE}}_{\ell,m}, \mathbf{V}^{\text{AE}}_{\ell,m} \in \mathbb{R}^{h \times d_{\text{head}}}$, and the corresponding key and value of the m-th head in layer $f(\ell)$ of the backbone be $\mathbf{K}^{\text{B}}_{f(\ell),m}, \mathbf{V}^{\text{B}}_{f(\ell),m} \in \mathbb{R}^{N_{\text{B}} \times d_{\text{head}}}$, where N_{B} denotes the backbone's input sequence length. The joint attention at layer ℓ for the m-th head is mathematically formulated as:

$$\text{Softmax}\left(\frac{1}{\sqrt{d_{\text{head}}}}\mathbf{Q}^{\text{AE}}_{\ell,m}\left(\text{concat}[\mathbf{K}^{\text{B}}_{f(\ell),m}\mathbf{W}^{\text{K}}_{\ell}, \mathbf{K}^{\text{AE}}_{\ell,m}]\right)^{\text{T}}\right)\text{concat}[\mathbf{V}^{\text{B}}_{f(\ell),m}\mathbf{W}^{\text{V}}_{\ell}, \mathbf{V}^{\text{AE}}_{\ell,m}], \tag{2}$$

where $\mathbf{W}^{\text{K}}_{\ell}, \mathbf{W}^{\text{V}}_{\ell} \in \mathbb{R}^{d_{\text{head}} \times d_{\text{head}}}$ represent learnable projection matrices. This linear projection on the backbone's key and value representations is designed to align the semantic spaces of the backbone and the action expert.

During real-world deployment, the simulated delay is set to $d = 4$ to cover physical inference latency. To achieve efficient closed-loop control, only the first 12 steps of the predicted 32-step action chunk are retained. Subtracting the 4 prefix steps used for latency conditioning, the robot actually executes 8 new action steps per inference cycle. This high-frequency, asynchronous execution enables highly responsive control, rendering the system robust to dynamic manipulation tasks.

World Model Expert. The world model expert is a lightweight Transformer comprising 4 layers, with its single-layer architecture identical to Qwen3's text layer. It has a hidden dimension of 1024, an intermediate size of 4096, and 8 attention heads with a head dimension $d_{\text{head}} = 128$, totaling approximately 70 M parameters. At layer ℓ, the world-model expert employs a joint attention mechanism, attending to both itself and the VLM backbone's key-value cache from layer $f(\ell) = 7\ell$ after a linear projection, consistent with the formulation of the action expert.

For inputs, the relative camera motion $\Delta\mathbf{T} \in SE(3)$ is flattened into a 16-dimensional vector and encoded into a single continuous token via a two-layer MLP. Let the ground-truth DINOv3 features of the future frame $\mathbf{I}_{t+h-1}$ be $\mathbf{Z} \in \mathbb{R}^{H_v \times W_v \times C_{\text{DINO}}}$. We utilize DINOv3 (ViT-L/16) (Siméoni et al., 2025) for feature extraction with an input resolution of 384×384, yielding a spatial resolution of $H_v = W_v = 24$ and a feature dimension of $C_{\text{DINO}} = 1024$. To align with the spatial token-merge format of the backbone, the sequence length $L_{\mathbf{z}}$ of the input query vector $\mathbf{z}$ is configured to match the merged spatial resolution: $L_{\mathbf{z}} = H'_v \times W'_v = \frac{H_v}{2} \times \frac{W_v}{2} = 144$. The world model expert outputs $\hat{\mathbf{Y}} \in \mathbb{R}^{H'_v \times W'_v \times d_{\text{WM}}}$ with a channel dimension $d_{\text{WM}} = 1024$, which is subsequently mapped back to the original DINOv3 spatial resolution through a 2×2 linear upsampling projection layer, yielding the reconstructed feature map $\hat{\mathbf{Z}} \in \mathbb{R}^{H_v \times W_v \times C_{\text{DINO}}}$. The world model objective is optimized via a

mean squared error (MSE) loss:

$$\mathcal{L}_{\text{WM}} = \frac{1}{H_v \cdot W_v \cdot C_{\text{DINO}}} \sum_{u=1}^{H_v} \sum_{v=1}^{W_v} \left\| \mathbf{Z}_{u,v} - \hat{\mathbf{Z}}_{u,v} \right\|_2^2. \tag{3}$$

Compared to direct image prediction in pixel space, regressing DINOv3 features preserves rich semantic information, naturally filtering out lighting variations and background noise to provide more stable gradient guidance for the VLM backbone. Furthermore, the world model expert adopts Interleaved MRoPE positional encoding, consistent with the VLM backbone, which enhances spatial-temporal awareness of multimodal sequences.

Attention Pattern. The Qwen3-VL backbone employs causal attention. Both the action expert and the world-model expert jointly attend to their entire respective sequences and the entire backbone sequence. Crucially, the action expert and the world-model expert do not attend to each other.

Data Processing. Due to significant variations in scale and quality across the 12 egocentric pre-training datasets, a heuristic sampling weight scheme is employed to balance different data sources. Specifically, each dataset i is assigned a subjective quality score $w_i \in [1, 10]$ based on data quality. To mitigate scale discrepancies, this score is scaled by the square root of the total frame count n_i, yielding the final sampling weight $W_i = w_i \cdot n_i^{0.5}$. For data augmentation, ColorJitter is applied to the input images. Furthermore, all action dimensions, except for wrist rotations, are normalized to the range $[-1, 1]$ using their 1st and 99th percentiles estimated from a randomly sampled subset.

Joint Optimization Objective. To balance the primary flow-matching task and the two auxiliary targets, the total training loss $\mathcal{L}_{\text{total}}$ is defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CFM}} + \mathcal{L}_{\text{WM}} + 0.05\mathcal{L}_{\text{VLM}}, \tag{4}$$

where $\mathcal{L}_{\text{CFM}}$ is the action flow-matching loss, $\mathcal{L}_{\text{WM}}$ is the world-model feature regression loss, and $\mathcal{L}_{\text{VLM}}$ is the autoregressive next-token prediction loss of the VLM. The loss weights are set to align the numerical scales of the three loss terms.

Training Infrastructure. Both the pre-training corpus curated by EgoSmith and the real-robot dataset are stored sequentially as individual episodes in the WebDataset format. During data loading, training batches are randomly sampled from a large shuffle buffer of size $16,384$. This buffer is continuously filled by drawing samples from each dataset proportionally to its pre-defined weight. For each dataset, the data stream is maintained by randomly selecting and streaming WebDataset shards, caching samples via a sliding window, and applying a 20% random retention probability. By combining randomized shard reading, sample dropping, and large-buffer shuffling, this scheme ensures training randomness while leveraging the sequential streaming of WebDataset, thereby drastically reducing I/O pressure. In addition to WebDataset, we also support the LeRobot v3 format. Furthermore, in multi-node distributed training, we explicitly manage Python's garbage collection to effectively mitigate training speed fluctuations and synchronization jitter across the cluster, ensuring high efficiency and stability.

D EXPERIMENTAL DETAILS

This section provides supplementary details for the experimental evaluations presented in Section 6, including the model training hyperparameters, per-task success rates, and additional illustrations. In particular, Table 4 lists the hyperparameter configurations for both the pre-training and post-training phases in the main experiments of EgoSteer in Section 6.1. Table 5 presents the per-task success rates on the four evaluation tasks in Section 6.2 to demonstrate the performance gains from DAgger refinement. For the scaling analysis in Section 6.3, Table 6 lists the training hyperparameters of EgoSteer-3 / 6 / 9.6K-FT models and the baseline trained from scratch, with their corresponding task-specific success rates reported in Table 7. Table 8 summarizes the post-training configurations of $\pi_{0.5}$ and Being-H0.5 used in our baseline comparisons in Section 6.3. Additionally, Table 9 compares the detailed success rates of our method against $\pi_{0.5}$ and Being-H0.5 across ten tasks. Table 11 specifies the training configurations of the ablation experiment from Section 6.4, including the EgoSteer-1K-FT model and its three ablated variants, *No WM-objective*, *No training-RTC*, and *Noisy data*. Their respective performance comparison is provided in Table 10. Table 12 specifies the hyperparameters for EgoSteer-9.6K-FS on the five challenging, long-horizon tasks evaluated in Section 6.5. Finally, illustrations of EgoSteer's language-following and mid-execution instruction-switching behaviors and few-shot experiment setups are shown in Figures 13 to 15.

Table 4: Training configurations for both the pre-training and post-training phases in the main experiments of EgoSteer in Section 6.1. During pre-training, the VLM backbone is frozen for the first 5,000 steps, during which AE and WM are warmed up for 2,000 steps; once it is unfrozen, VLM is warmed up for 2,000 steps.

Hyperparameter	Pre-Training	Post-Training
Camera setup	Head	Head & Chest
Resolution	384×384	640×480
GPUs	128 A800	96 A800
Gradient accumulation	2	1
Global batch size	4608	384
Training steps	175K	60K
Learning rate (VLM / AE / WM)	1×10^{-4} / 3×10^{-4} / 3×10^{-4}	1×10^{-5} / 3×10^{-5} / 3×10^{-5}
Freeze-VLM steps	5000	0
Warmup steps	2000	2000
Training time	164 h	29 h

Table 5: Success rates across four highly dexterous and failure-prone manipulation tasks, comparing EgoSteer-9.6K-DG against EgoSteer-9.6K-FT in Section 6.2. Each task is evaluated over 10 randomized trials. Bold values highlight the best performance.

Task	EgoSteer-9.6K-DG	EgoSteer-9.6K-FT
Stack tableware	**80.0%**	50.0%
Close laptop	**70.0%**	10.0%
Place phone on stand	**50.0%**	0.0%
Flip cup	**50.0%**	30.0%
Average	**62.5%**	22.5%

Table 6: Training configurations for the pre-training scaling study in Section 6.3.

	Pre-Training			Post-Training			
Hyperparameter	EgoSteer-3K-PT	EgoSteer-6K-PT	EgoSteer-9.6K-PT	EgoSteer-0K-FT	EgoSteer-3K-FT	EgoSteer-6K-FT	EgoSteer-9.6K-FT
Camera setup	Head	Head	Head	Head & Chest	Head & Chest	Head & Chest	Head & Chest
Resolution	384×384	384×384	384×384	640×480	640×480	640×480	640×480
GPUs	64 A800	64 A800	128 A800	64 A800	32 A800	64 A800	64 A800
Gradient accumulation	2	4	2	2	4	2	2
Global batch size	2304	4608	4608	512	512	512	512
Training steps	100K	100K	160K	60K	60K	60K	60K
Learning rate (VLM / AE / WM)	1×10^{-4} / 3×10^{-4} / 3×10^{-4}			1×10^{-5} / 3×10^{-5} / 3×10^{-5}			

Table 7: Per-task success rates for the pre-training scaling study in Section 6.3. Each task is evaluated over 10 randomized trials. Bold values highlight the best performance.

Task	EgoSteer-0K-FT	EgoSteer-3K-FT	EgoSteer-6K-FT	EgoSteer-9.6K-FT
Grasp object	80%	80%	70%	**100%**
Hand over object	70%	80%	80%	**100%**
Place items into container	40%	80%	90%	**100%**
Point at object	20%	40%	60%	**70%**
Place toy chick into slot	30%	0%	20%	**40%**
Pull out tissue	60%	20%	50%	**80%**
Push ball into box	0%	0%	10%	**20%**
Put garbage into trash bin	0%	**60%**	30%	30%
Stack paper cups	0%	20%	10%	**40%**
Stack tableware	0%	**20%**	10%	**20%**
Average	30%	40%	43%	**60%**

Table 8: Post-training configurations of $\pi_{0.5}$ and Being-H0.5. All methods are post-trained on the same real-robot dataset for 60K optimizer updates and evaluated using their final checkpoints. Both baselines retain their officially recommended input resolution. For $\pi_{0.5}$, we adopt EgoSteer's 48-dimensional wrist-pose and fingertip representation. For Being-H0.5, we retain the officially recommended action representation.

Hyperparameter	$\pi_{0.5}$	Being-H0.5
Pre-training checkpoint	`pi05_droid`	Official released base checkpoint
Camera setup	Head & Chest	Head & Chest
Resolution	224×224	224×224
Action representation	Wrist poses + fingertip positions	EEF poses + dexterous hand joints
Gradient accumulation	1	1
Global batch size	128	672
Training steps	60K	60K
Peak learning rate	5×10^{-5}	1×10^{-4}
Freeze-VLM steps	0	0
Warmup steps	1,500	3,000
Evaluation checkpoint	Final at 60K steps	Final at 60K steps

Table 9: Per-task comparison between EgoSteer-9.6K-FT and two VLA baselines, Being-H0.5 and $\pi_{0.5}$, in Section 6.3. All methods are fine-tuned on our real-robot dataset and evaluated on the same 10 tasks. Each task uses 10 randomized trials. Bold values highlight the best performance.

Task	EgoSteer-9.6K-FT	Being-H0.5	$\pi_{0.5}$
Grasp object	**100%**	80%	80%
Hand over object	**100%**	60%	20%
Place items into container	**100%**	50%	0%
Pour items out of box	**50%**	30%	0%
Place bread on tray	70%	**80%**	40%
Pull out tissue	**80%**	10%	30%
Place item at specific orientation	**30%**	**30%**	**30%**
Attach eraser to whiteboard	**90%**	50%	20%
Put garbage into trash bin	**30%**	0%	0%
Put tennis ball into bucket	**90%**	0%	0%
Average	**74%**	39%	22%

Table 10: Per-task success rates for the ablation study in Section 6.4. EgoSteer-1K-FT is compared with ablated variants. Each task uses 10 randomized trials. Bold values highlight the best performance.

Task	EgoSteer-1K-FT	No WM-objective	No training-RTC	Noisy data
Grasp object	**60%**	40%	30%	**60%**
Hand over object	**40%**	20%	30%	**40%**
Place items into container	50%	30%	50%	**70%**
Pour items out of box	**60%**	50%	10%	0%
Place bread on tray	**70%**	10%	60%	50%
Pull out tissue	20%	30%	**60%**	10%
Place item at specific orientation	**20%**	10%	10%	**20%**
Attach eraser to whiteboard	50%	**80%**	50%	40%
Put garbage into trash bin	**30%**	0%	0%	0%
Put tennis ball into bucket	40%	40%	**90%**	40%
Average	**44%**	31%	39%	33%

Table 11: Training hyperparameters for the ablation study in Section 6.4. All variants are pre-trained on 1K hours of egocentric data and fine-tuned on the same real-robot dataset, without DAgger refinement. Pre-training steps are selected at the lowest evaluation L_1 loss for each pre-training run.

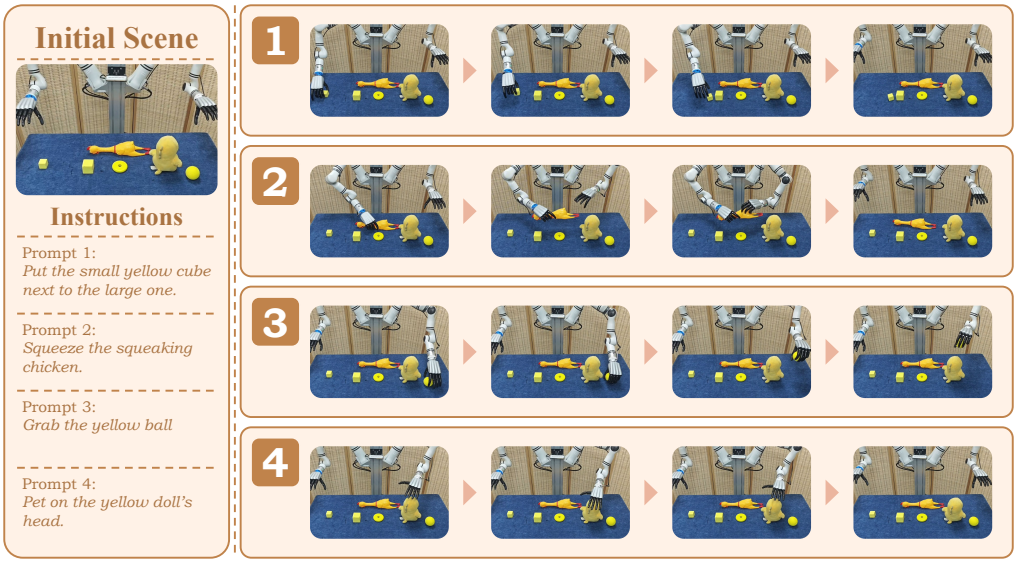
	Pre-Training				Fine-Tuning			
Hyperparameter	EgoSteer-1K-PT	No WM-objective	No training-RTC	Noisy data	EgoSteer-1K-FT	No WM-objective	No training-RTC	Noisy data
Camera setup	Head	Head	Head	Head	Head	Head	Head	Head
Resolution	384×384	384×384	384×384	384×384	384×384	384×384	384×384	384×384
GPUs	64 A800	64 A800	64 A800	64 A800	64 A800	64 A800	64 A800	64 A800
Global batch size	1152	1152	1152	1152	1152	1152	1152	1152
Training steps	30K	80K	30K	20K	60K	60K	60K	60K
Learning rate (VLM / AE / WM)	1×10^{-4} / 3×10^{-4} / 3×10^{-4}				1×10^{-5} / 3×10^{-5} / 3×10^{-5}			

Table 12: Few-shot fine-tuning hyperparameters of EgoSteer-9.6K-FS for the five long-horizon dexterous tasks in Section 6.5.

Hyperparameter	Shared value	
Pre-training checkpoint	EgoSteer-9.6K-PT at 170K steps	
Camera setup	Head	
Resolution	384×384	
GPUs	8 A800	
Gradient accumulation	1	
Global batch size	144	
Learning rate (VLM / AE / WM)	1×10^{-5} / 3×10^{-5} / 3×10^{-5}	
Task	Fine-tuning steps	
	20 demonstrations	100 demonstrations
Box-Folding	1.6×10^4	2.4×10^4
Cake-Unboxing	3.5×10^4	2.3×10^4
Cup-Stacking	1.3×10^4	1.9×10^4
Phone-Packing	1.6×10^4	2.0×10^4
Cola-Making	1.8×10^4	2.6×10^4

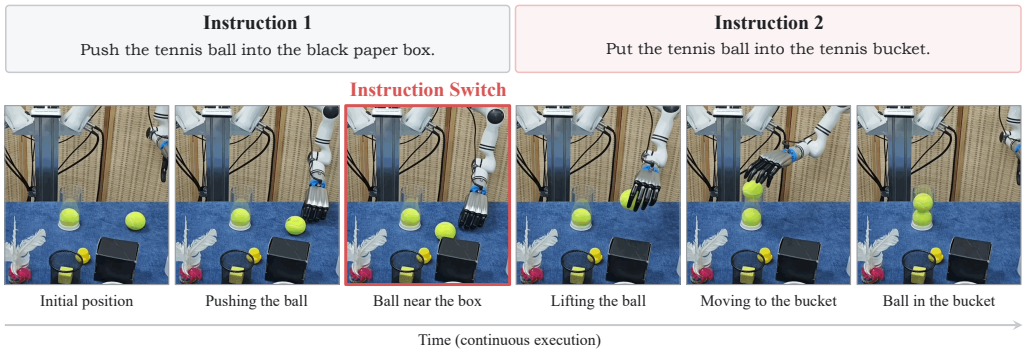


(a) Steerability in an office scene.

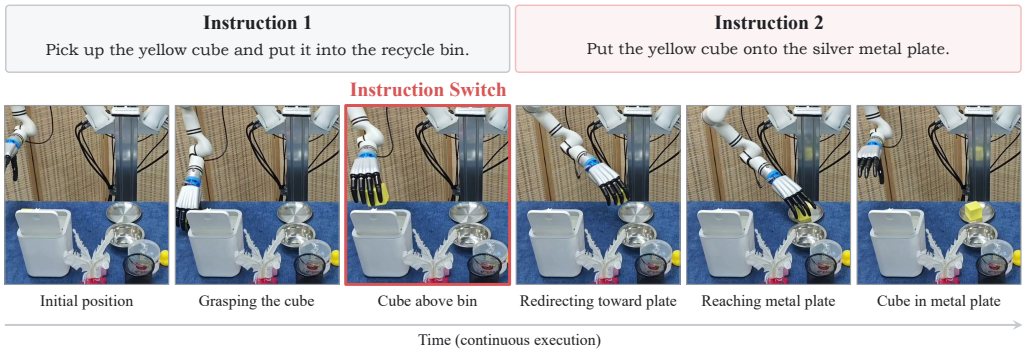


(b) Steerability among objects with similar colors in the same scene.

Figure 13: Illustrations of the steerability of EgoSteer. It is capable of executing different actions given different language instructions under the same initial scene configuration.

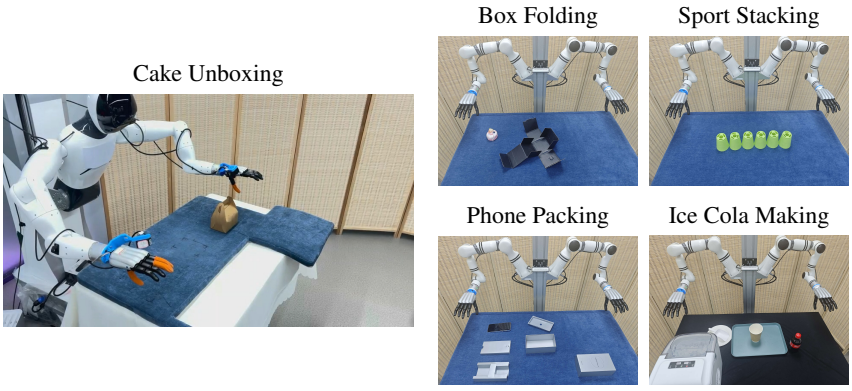


(a) The robot initially pushes the tennis ball toward the box. After the instruction switch, it redirects the ball to the bucket.



(b) The robot initially moves the yellow cube toward the recycle bin. After the instruction switch, it redirects the cube to the metal plate.

Figure 14: Illustrations of EgoSteer handling instruction switching during continuous execution. The instruction is switched at an intermediate timestep, after which EgoSteer adjusts its subsequent behavior to follow the updated instruction. This indicates that EgoSteer is genuinely following language guidance rather than memorizing spurious scene-task correlation. The red box indicates the switching moment.



(a) AgiBot G1. **(b)** RealMan.

Figure 15: Experimental setups for the five single-task evaluations. The photographs show the robot embodiments and representative tabletop object arrangements.