Robotics: Science and Systems 2026
Sydney, Australia, July 13–17, 2026

EgoVerse: An Egocentric Human Dataset for Robot Learning from Around the World

Ryan Punamiya*[1], Simar Kareer*[1], Zeyi Liu[2], Josh Citron[2], Ri-Zhao Qiu[3], Xiongyi Cai[3], Alexey Gavryushin[4], Jiaqi Chen[4], Davide Liconti[4] Lawrence Y. Zhu[1], Patcharapong Aphiwetsa[1], Baoyu Li[1], Aniketh Cheluva[1], Pranav Kuppili[1], Yangcen Liu[1], Dhruv Patel[1], Aidan Gao[1], Hye-Young Chung[1], Ryan Co[1], Renee Zbizika[2], Jeff Jinyun Liu[2], Xiaomeng Xu[2], Haoyu Xiong[5], Geng Chen[3], Sebastiano Oliani[4], Chenyu Yang[4], Xi Wang[4], James Fort[6], Richard Newcombe[6], Josh Gao[7], Jason Chong[7], Garrett Matsuda[8], Aseem Doriwala[8], Marc Pollefeys[4], Robert Katzschmann[4], Xiaolong Wang[3], Shuran Song[2], Judy Hoffman[1], Danfei Xu[1]

[1]Georgia Institute of Technology, [2]Stanford University, [3]University of California San Diego, [4]ETH Zürich
[5]MIT CSAIL, [6]Meta Reality Labs Research, [7]Mecka AI, [8]Scale AI

Fig. 1: **Overview.** EgoVerse is a collaborative framework for scalable human data–driven robot learning. **Capture:** Egocentric demonstrations are collected worldwide using academic, industry, and community-accessible hardware systems, continuously aggregated by a centrally-hosted data management system. **Dataset:** All data are unified into a shared dataset with egocentric video, 3D hand poses, camera motion, and task descriptions across diverse tasks and scenes. **Evaluation:** This work presents a large-scale evaluation study on human-to-robot transfer with shared protocols across multiple labs and robot embodiments.

***Abstract*—Robot learning increasingly depends on large and diverse data, yet robot data collection remains expensive and difficult to scale. Egocentric human data offer a promising alternative by capturing rich manipulation behavior across everyday environments. However, existing human datasets are often limited in scope, difficult to extend, and fragmented across institutions. We introduce EgoVerse, a collaborative platform for human data–driven robot learning that unifies data collection, processing, and access under a shared framework, enabling contributions from individual researchers, academic labs, and industry partners. The current release includes 1,362 hours (80k episodes) of human demonstrations spanning 1,965 tasks, 240 scenes, and 2,087 unique demonstrators, with standardized formats, manipulation-relevant annotations, and tooling for downstream learning. Beyond the dataset, we conduct a large-scale study of human-to-robot transfer with experiments replicated across multiple labs, tasks, and robot embodiments under shared protocols. We find that policy performance generally improves with increased human data, but that effective scaling depends on alignment between human data and robot learning objectives. Together, the dataset, platform, and study establish a foundation for reproducible progress in human data–driven robot learning.**

I. Introduction

Recent progress in robot learning has shown that scaling data is a powerful driver of generalization [5, 26, 31, 39].

Large-scale imitation learning has enabled policies to handle broader task distributions, more visual variation, and longer horizons, echoing trends seen in large vision and language models. However, unlike those domains, robot learning faces a fundamental bottleneck – collecting robot demonstrations requires physical hardware, expert teleoperation, and controlled setups. As a result, expanding robot datasets in scale and diversity remains slow, expensive, and difficult to sustain.

In contrast, egocentric human data offers a promising alternative. Humans naturally perform manipulation and loco-manipulation tasks across diverse environments on a daily basis, generating rich behavioral data at a scale that is infeasible for robots alone. Importantly, human data also provides a unifying abstraction for the community. Instead of coordinating around a specific robot embodiment, researchers can focus on curating diverse, real-world experience data while deferring embodiment decisions downstream. This property has driven growing academic and commercial interest in leveraging egocentric human demonstrations, supported by recent advances in wearable sensors [14, 25, 56] and large-scale data capture systems [11, 15].

Despite the promise of leveraging human data, two major challenges remain. First, effective human-robot transfer remains an open research problem, with unresolved questions around the embodiment gap and scaling behavior. Second, most existing human datasets are one-off, static releases collected for a specific study, making further scaling difficult [25, 29, 41]. Addressing these limitations requires more than collecting a larger dataset: it calls for a continuously growing human data ecosystem that can evolve with new contributors and provide durable insights into human-to-robot transfer.

This paper introduces **EgoVerse**, a large-scale collaborative framework to create an ever-growing dataset of egocentric human demonstrations specifically designed for robot learning, paired with a systematic study of the key factors that enable effective cross-embodiment transfer from diverse human data sources. Our contributions are threefold:

The EgoVerse Dataset. We present a large dataset of egocentric human demonstrations contributed by a consortium of academic groups and industry partners worldwide. The dataset is intentionally composed of **two complementary parts** with distinct purposes. The first, named `EgoVerse-A`, consists of data collected under carefully controlled and standardized protocols, mirrored across participating academic labs. This component is designed to enable *reproducible studies* and *systematic analysis*. The second, named `EgoVerse-I`, focuses on scale, diversity, and richness of annotation, and is sourced from industry partners collecting data in the wild. This industry-driven stream enables EgoVerse to grow beyond the limits of academic institutions and to align research with real-world deployment and industrial interests. Overall, EgoVerse contains 1,362 hours of humassn demonstrations across 240 scenes, and 1,965 tasks, and 2,087 demonstrators. Across both components, the dataset captures manipulation tasks with unprecedented diversity across scenes, objects, and human demonstrators, while maintaining consistency through shared task semantics, standardized protocols, and high-quality manipulation-relevant signals, including 3D hand and head poses, and subtask-level descriptions.

The EgoVerse Ecosystem. EgoVerse is designed to grow over time with contributions from both academic labs and industry partners. To unify this evolving collection, we introduce **EgoDB**, a scalable data management and access system that supports continuous data ingestion from diverse sources, including individual contributors, academic labs, and industry partners. Unlike prior static datasets [3, 16, 25, 36], EgoDB enables the notion of a *living dataset* that grows and evolves with ongoing contributions. The system provides standardized data processing, unified storage formats, controlled data access, visualization tools, and interfaces for downstream learning algorithms. In addition to lab- and partner-operated capture systems, EgoVerse includes a **phone-based human data collection pipeline** that enables lightweight egocentric recording using commodity smartphones. Together, the EgoVerse ecosystem lowers the barrier for groups and individuals with limited resources while also enabling contributors to share data back with the community in a structured and reproducible way.

A Consortium-Scale Study. We conduct the most comprehensive study to date on what matters when learning robot manipulation policies from diverse human embodiment data. Our study is **reproducible by design**: experiments are intentionally replicated across multiple independent labs, tasks, and robot embodiments using shared protocols and evaluation criteria. In particular, the results are executed on **three distinctive robot embodiments** to ensure the main findings are not system-specific. This cross-lab, cross-embodiment setup allows us to identify conclusions that are consistent and robust across settings, as well as effects that systematically differ due to embodiment, sensing, or control variations.

Key Findings. Our study yields several consistent findings across labs, tasks, and robot embodiments. First, co-training robot policies with human data leads to clear and reproducible performance improvements. To our knowledge, this is the first time this effect is validated under a standardized, cross-lab experimental setup spanning multiple robots. Second, the benefits of scaling human data depend critically on the availability of aligned human–robot data, where human and robot data share task semantics and scene context. We find that positive scaling emerges only when such aligned data are included as part of training, suggesting it provides essential grounding for effective human-to-robot transfer. Finally, we find that different forms of human data diversity contribute unevenly to generalization. Increasing demonstrator diversity improves robustness to unseen human embodiments, while scene diversity plays a dominant role in generalization to novel environments, particularly under limited data budgets.

II. Related Work

Datasets of Human Activities. Large-scale human activity datasets such as Something-Something V2 [19], Ego4D [20], HOI4D [36], EgoExo4D [21], and Epic-Kitchens [12] capture rich human behavior across diverse environments. However,

they are not designed for robot learning. These datasets often include tasks beyond current robot capabilities, lack manipulation-relevant annotations such as precise hand poses or object interactions, and contain unstructured activities that are difficult to translate into executable robot demonstrations. In contrast, our approach emphasizes "bounded diversity", focusing on tasks that are feasible for typical bimanual mobile manipulators while preserving natural variation across environments, objects, and demonstrators. More recent datasets [3, 25] introduce manipulation-relevant sensing, including camera calibration and hand pose tracking. While these represent important progress, they are typically released as static, study-specific datasets collected over limited time spans and environments, making them difficult to extend. Moreover, these works do not include systematic evaluation of transfer to robot learning. EgoVerse takes a different approach by treating human data as a continuously growing research resource, and by explicitly validating the dataset through a large-scale, reproducible study designed to ensure robot-learning readiness.

Robot Learning from Human Data. Human data presents two main opportunities for robot learning: abundant unlabeled online videos and curated, labeled demonstrations [1, 29, 46, 52]. Web videos, though plentiful, require pseudo-labeling of actions via inverse dynamics models [6, 13, 55], affordances [2, 44], or point tracking [4, 42, 50] for policy training, forming a basis for some foundation models [8, 37, 38, 54], yet often still necessitating in-domain robot data. Alternatively, labeled human demonstrations can be co-trained with robot data as distinct embodiments for policy learning [22, 29, 32, 35, 41, 59], post-training [7, 30], and world modeling [18, 24]. These works found that this practice enhances robustness and scene understanding. However, such findings remain confined to limited scale and single robot embodiment, leaving critical questions about multiple robot embodiments and varied human data sources largely unexplored. Our work addresses these fundamental gaps through a large-scale human dataset and a carefully-executed consortium-scale study.

Scaling Robot Learning with Massive Data. Recent progress in foundational policy models highlights the benefits of scaling with large datasets. Public efforts such as Open X-Embodiment [39], DROID [31], and Rh20t [17] demonstrate that training on diverse, multi-embodiment data improves generalization across tasks and environments. However, achieving generally capable robots remains fundamentally constrained by data scalability, as robot teleoperation is expensive and labor-intensive. Our work instead examines how human egocentric data can support robot learning at scale and study it as a *first-class data source* alongside robot data.

III. The EgoVerse Dataset: A Human Dataset for Robot Learning from Around the World

A. Human Data Collection Setup

Academic partners used Project Aria glasses as the standard device for `EgoVerse-A`. Industry partners contributed to `EgoVerse-I` with custom-built rigs for scalability and ease of deployment. We also introduce a phone-based capture system that is accessible by the broader community. The setups are illustrated in Fig. 2, with more detail in Appendix.

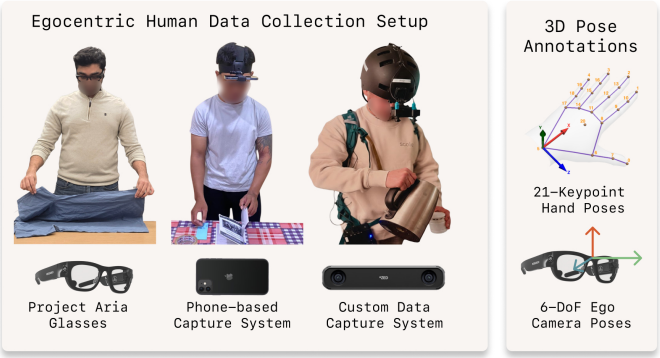

Fig. 2: **Human Data Capture Setup.** (Left) EgoVerse is captured through a variety of hardware systems, including Project Aria glasses (academic labs), a phone-based capture system (accessible by everyone), and custom setups by industry partners. (Right) Regardless of sources, human data is processed into a unified format that contains at minimum egocentric videos, hand keypoints, and camera poses.

Project Aria glasses for human data collection. Following prior work [29, 34, 40, 59], we adopt Project Aria glasses (Gen 1) as the standardized capture platform for `EgoVerse-A`. Project Aria glasses are lightweight (75 g) head-worn devices with a wide-FoV RGB camera and two synchronized monochrome scene cameras used for SLAM and hand tracking (Fig. 2). The side cameras maintain visibility of hand motion even when out of the forward-looking RGB view.

Industry partner data collection setup. The `EgoVerse-I` stream aggregates demonstrations collected using custom wearable sensor platforms deployed in diverse indoor environments. These rigs typically feature lightweight head-mounted cameras paired with synchronized sensing modules. Most systems use stereo fisheye RGB cameras to achieve accurate hand pose estimation. The data include ego-view RGB videos and, where available, depth streams, with additional inertial sensing to reconstruct camera motion.

Phone-based capture system. To broaden access to data capture devices, we also make available a setup based on commodity smartphones as part of the EgoVerse ecosystem. This system uses an iPhone mounted on a head strap, with the ultrawide camera recording egocentric RGB video at 1080p and 30 FPS. Captured videos are uploaded to a cloud-based processing pipeline that recovers 6-DoF head pose via visual tracking and estimates 3D hand poses with 21 keypoints per hand. More details on the system are in the Appendix.

B. Human Data Annotation

EgoVerse augments egocentric human demonstrations with structured annotations tailored for robot policy learning. For each frame, we estimate 3D hand pose for both hands using 21 keypoints per hand in the camera frame, paired with a calibrated 6-DoF head pose obtained from visual–inertial SLAM. Academic partners use Project Aria's Machine Perception

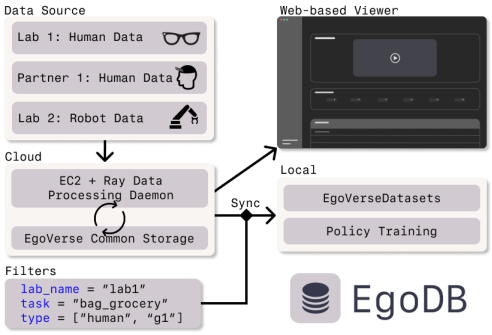

Fig. 3: **EgoDB**. Human and robot data from multiple labs and partners are ingested into a cloud-based processing pipeline, unified in a common storage format, and made accessible through a web-based viewer. Users can sync filtered subsets of the dataset to local machines for downstream policy training.

Service (MPS) for tracking and egomotion, while industry datasets combine partner SLAM, model-based pose estimation, and post-processing to ensure consistent trajectories. In our human-to-robot transfer studies (Sec. IV), these signals serve as proxies for human end-effector motion, enabling cotraining across embodiments. Beyond poses, annotation granularity differs across dataset components. `EgoVerse-A` follows a lightweight per-episode protocol, attaching task descriptions, scene identifiers, primary manipulated objects, and demonstrator metadata to support controlled cross-lab analysis. In contrast, `EgoVerse-I` provides denser annotations, including fine-grained (1–2 s) language descriptions, active-hand indicators, static versus mobile manipulation flags, and additional contextual tags where available.

C. EgoDB: Scalable Data Management System

To support consortium-wide collaboration and long-term dataset growth, we develop EgoDB, a cloud-based system for continuous ingestion, processing, and access of heterogeneous human and robot data (Fig. 3). Data from distributed sources are uploaded to S3-backed storage and converted into a unified, training-ready format shared across the EgoVerse ecosystem. A nightly pipeline performs standardized preprocessing, validation, and indexing to ensure consistent usability for downstream learning. Episode metadata are registered in a centralized SQL database, enabling structured queries over tasks, embodiments, scenes, data sources, and annotation types. EgoDB also provides a web-based interface for browsing demonstrations, inspecting annotations, and tracking dataset growth. For local training, users can synchronize filtered subsets of the dataset via configuration files, enabling reproducible access without manual data management.

D. The `EgoVerse-A` Dataset

Collection Protocol and Dataset Units. To ensure standardized data collection across geographically distributed sites, we adopt a shared protocol organized around *dataset units*. Each unit follows a common instruction format to ensure consistent task execution across labs, and typically consists of approximately 5 minutes of recording, yielding 5–10 demonstrations per task. We enforce basic quality constraints to maintain visibility of hands and manipulated objects, and log key metadata such as demonstrator identity, scene, and object set for traceability. Additional protocol details and quality control procedures are provided in the appendix.

Flagship Tasks. To control task semantics while allowing other axes of variation, we define a set of six *Flagship Tasks* shared across all participating labs.

- *object-in-container*: Pick, place into a container, dump, and repeat continuously for 40 seconds with randomized object–container pairs. Single-arm task.
- *cup-on-saucer*: Reorient a cup from a random initial pose and place it on a saucer. Bimanual task.
- *bag-grocery*: Open a grocery bag and load 1–3 items. Bimanual task.
- *fold-clothes*: Three-fold a T–shirt initialized in random configurations. Bimanual task.
- *scoop-granular*: Scoop granular material (e.g., beans) and transfer it to a container until full. Single-arm task.
- *sort-utensils*: Pick and sort utensils into designated containers. Single-arm task.

These tasks span a range of manipulation regimes, including single-arm and bimanual coordination, fine-grained object placement, and longer-horizon behaviors, while remaining feasible for common robot manipulation platforms.

Structured Axes of Diversity. `EgoVerse-A` is designed to capture diversity while maintaining controlled task semantics and data quality. Human data collection is organized along three primary axes: *task*, *scenario*, and *demonstrator*.

Scenario and object diversity. Each flagship task is performed across 8–12 scenes per lab, with 1–10 dataset units collected per scene to capture within-environment variation. Demonstrations are recorded within a roughly $40\text{cm} \times 60\text{cm}$ workspace, with object positions randomized across trials. Objects are sampled from a fixed set of up to 30 per task within each lab, while independent procurement across sites introduces substantial variation in object geometry, appearance, and material properties.

Demonstrator diversity. Data are collected from 1–8 demonstrators per lab. Despite identical instructions, demonstrators within and across labs exhibit consistent differences in motion patterns, timing, coordination strategies, and hand trajectories. This naturally induces variation in human morphology and egocentric viewpoints due to differences in height, posture, and workspace configuration. Prior work shows that such human variability can significantly influence policy learning and cross-embodiment alignment [29, 35, 40]. Rather than eliminating this variation, we treat it as an inherent property of scalable human data.

Controlled-Diversity Subset. While cross-lab aggregation provides realistic diversity, it also introduces uneven coverage across scenes and demonstrators. To study the effects of diversity under controlled conditions, one lab collected a secondary dataset for *cup-on-saucer* and *fold-clothes* using a fixed pool of 16 demonstrators and 16 scenes. Data were

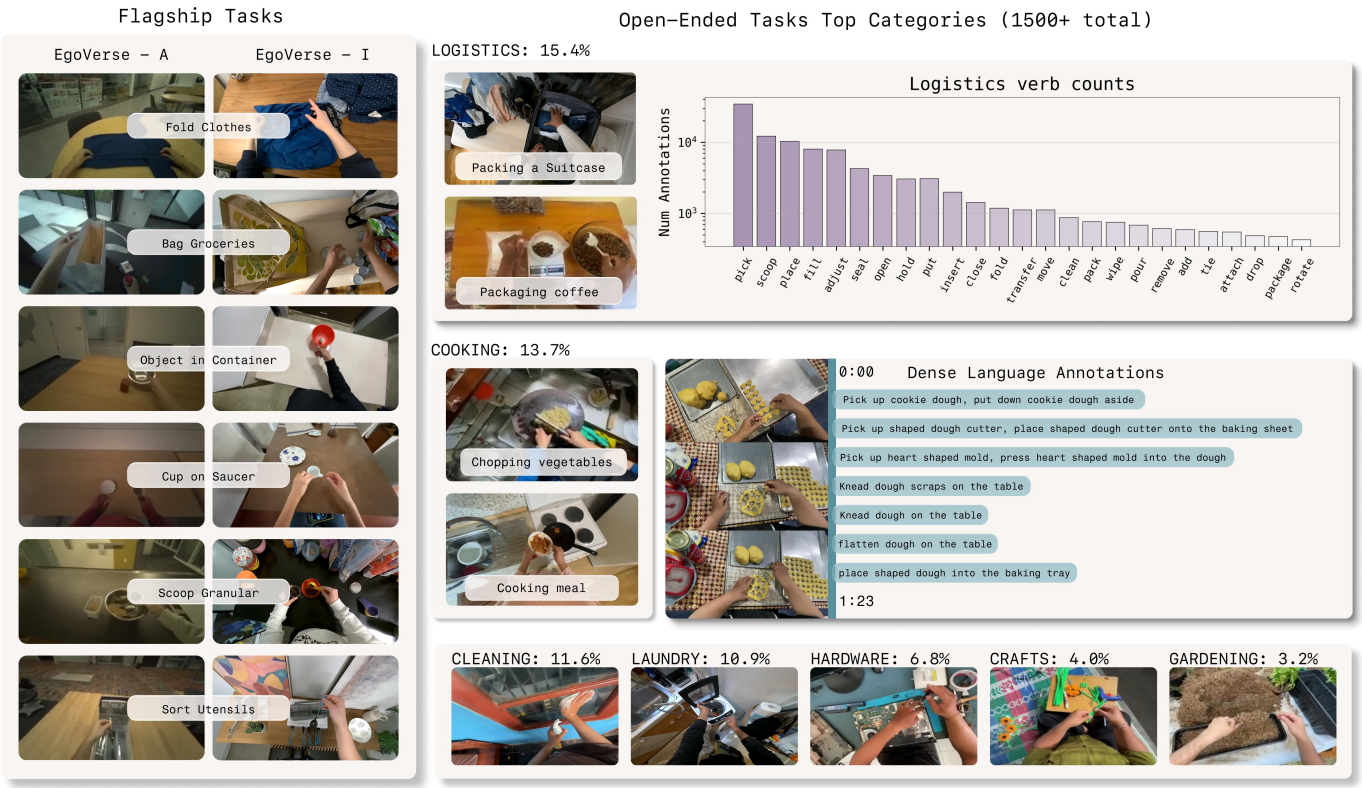

Fig. 4: **Dataset Composition and Diversity.** Left: EgoVerse-A and EgoVerse-I include six shared flagship manipulation tasks collected across diverse scenes and demonstrators. Right: EgoVerse-I contains over 1,500 open-ended tasks spanning everyday activity categories, with representative verb frequency distributions illustrating the diversity of manipulation actions.

allocated via a structured assignment matrix (3.75 minutes to 2 hours per demonstrator–scene pair; $\geq$1 hour per scene) across three experimental regimes:

- **Single-Scene Demonstrator Scaling.** A fixed data budget of 2 hours is varied in a single scene as the number of demonstrators increases from 1 to 16 to study the impact of demonstrator diversity in a single scene.
- **Multi-Scene Interaction Effects.** Scenes 1–8 are used to study demonstrator scaling across diverse environments, where the number of demonstrators is varied from 4–12 with a fixed data budget of 8 hours.
- **Scene Diversity Scaling.** Scenes 1–16 are collected in a fixed demonstrator pool, decoupling scene diversity and data quantity.

By enforcing control over sources of variation, this subset allows scene diversity and demonstrator diversity to be independently scaled and studied (Sec. IV-F).

E. The EgoVerse-I Dataset

EgoVerse-A densely covers the flagship tasks along controlled axes but lacks the task diversity required to train generalist policies. To address this, we introduce EgoVerse-I, the largest action-labeled egocentric human dataset, comprising nearly 1,400 hours of data across nearly 2,000 tasks, 240 scenes, and 2,087 demonstrators (Fig. 4). EgoVerse-I is designed to be maximally useful for robot learning by emphasizing manipulation-heavy task distributions, instructing demonstrators to keep hands visible, and applying manual quality control to retain only manipulation-dense segments. In addition, EgoVerse-I includes dense language annotations (Sec. III-B), making it suitable for training language-conditioned policies such as VLAs. Fig. 4 compares flagship task examples from EgoVerse-A and EgoVerse-I, and illustrates representative language annotations and verb statistics for the open-ended tasks in EgoVerse-I.

Fig. 5: UMAP of DINOv3 embeddings.

Visual Diversity Case Study. To compare visual diversity across data sources, we apply UMAP to DINOv3 embeddings from the *fold-clothes* task across three sources: robot data from a single lab, human data in EgoVerse-A, and human data in EgoVerse-I. As shown in Fig. 5, despite shared task semantics, EgoVerse-I substantially expands the visual coverage. This increased diversity is critical for learning policies that generalize beyond lab-specific appearance statistics.

IV. The EgoVerse Study: A Consortium-Scale Examination of Human-to-Robot Transfer

We conduct a large-scale systematic study to examine the factors that enable effective transfer from diverse human embodiment data to robot manipulation policies. Our experimental design prioritizes *reproducibility*: robot experiments are replicated across multiple labs using distinct platforms, controllers, and environments. This multi-lab, multi-embodiment setup enables identification of findings that are consistent across hardware and sensing configurations, ensuring that our conclusions extend beyond system-specific effects. Using shared protocols and evaluation criteria, we study how human data scale and diversity influence human-to-robot transfer.

A. Robot Platforms

Our experiments are carried out on three distinctive robot platforms with varied kinematics, sensing configuration and control interfaces. This allows us to assess which findings are consistent across systems rather than specific to a single robot. (Fig. 6)

`Robot A`. The system comprises of two 6-DoF ARX5 robot arms with parallel jaw grippers. The arms are mounted upright. The main egocentric camera is an Aria Glasses, with two wrist-mounted Intel RealSense D405 cameras.

`Robot B`. Different than Robot A, two ARX5 arms are side-mounted on a custom 3D-printed shoulder structure for human-like workspace [53]. The robot is equipped with a head-mounted Aria glass and wrist-mounted Logitech webcams on each end effector.

`Robot C`. The system is a Unitree G1 robot with 7-DoF arms, each equipped with a 6-DoF Dexterous Inspire Hand. The main egocentric camera is a ZED 2 stereo camera.

B. Human and Robot Data Alignment

Robot Actions Representations. For `Robot A`, base-frame $\mathbb{SE}(3)$ actions are computed from commanded joint angles via forward kinematics, projected into the egocentric camera frame using extrinsics, and represented as 6-DoF Euler poses $(x, y, z, \text{yaw, pitch, roll})$ with a gripper state for each arm, yielding $a^R_{t:t+k} \in \mathbb{R}^{k\times 14}$. `Robot B` follows a similar pipeline, but represents orientation using quaternions $(x, y, z, q_x, q_y, q_z, q_w)$ along with the gripper state, resulting in $a^R_{t:t+k} \in \mathbb{R}^{k\times 16}$. For `Robot C`, the wrist motion is modeled as an absolute $\mathbb{SE}(3)$ pose trajectory in the robot base frame, and dexterous hand control is specified via five keypoint positions relative to the end-effector, which are mapped to joint commands using an inverse kinematics solver.

Human Action Representation. Human egocentric hand tracking is usually with respect to a moving camera frame. Following prior work [29, 41], to unify the reference frames between human and robot for joint policy learning, we opt to construct camera-centered stable reference frames. The raw trajectory of $\mathbb{SE}(3)$ hand poses $[p^H_t,\ p^H_{t+1},\ p^H_{t+2},\ \ldots,\ p^H_{t+k}]$, where each pose p^H_t is in the device frame T^{device}_t. We construct an action $a^H_{t:t+k}$ by projecting the future hand positions in to the t-th device frame. As such, the trajectory is constructed by

$$a^H_{t:t+k} = \left[\left(T^{\text{device}}_t\right)^{-1} T^{\text{device}}_{t+i} \cdot p^H_{t+i} \right]_{i=1}^{k}$$

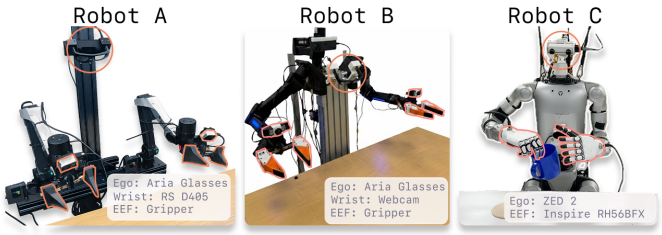

Fig. 6: **Robot Platforms.** We perform evaluation on three distinctive robot platforms across labs with shared protocols.

Aligning Human and Robot Data. Recent cross-embodiment work [29, 40, 48, 59] show that co-training benefits from individually normalizing proprioception and actions. To make the normalization robust to outliers, we employ *quantile* normalization. We map the 1st and 99th percentiles of the feature distribution to the range $[-1, 1]$, following [10, 27]. For a feature tensor x, the normalized output $\hat{x}$ is calculated as $\hat{x} = 2 \cdot \left(\frac{x - q_{0.01}}{q_{0.99} - q_{0.01}}\right) - 1$. To account for varying camera sensors, we perform random image crop and color jittering during training.

C. Learning Architecture and Algorithm

Policy Architecture. To enable joint training across diverse embodiments, we adopt an encoder–decoder architecture with shallow, modality-specific stems [49]. Image observations are processed by a ResNet-18 [23] backbone, while proprioceptive inputs are encoded with an MLP, before being tokenized into a shared space via learned query attention. A shared vision stem processes egocentric RGB observations from both human and robot embodiments, while separate stems handle robot-specific wrist cameras and proprioception. The resulting tokens are concatenated and passed, together with a set of learnable tokens, through a shared transformer encoder f_ϕ. These learnable tokens attend to the multi-modal inputs to extract task-relevant features and condition the action decoders.

Flow Matching Action Decoder. The action decoder π_θ is parameterized by a multi-block transformer decoder trained with a flow matching objective. T learnable tokens which correspond to the action sequence are initialized. The timestep $\tau \sim \text{Beta}(1.5, 1.0)$ is positionally embedded and concatenated to the learnable tokens along the hidden dimension. With alternating self and cross attention blocks, we inject the encoded context from the encoder. The learnable tokens are decoded into the action dimension using a linear layer. Depending on the output action space, we initialize shared or embodiment-specific action decoders.

Training Objective. The encoder f_ϕ and the decoder π_θ are jointly optimized with the BC co-training loss. The BC co-training loss is a popular approach for cross-embodiment policy learning, where a standard Behavior Cloning (BC) loss

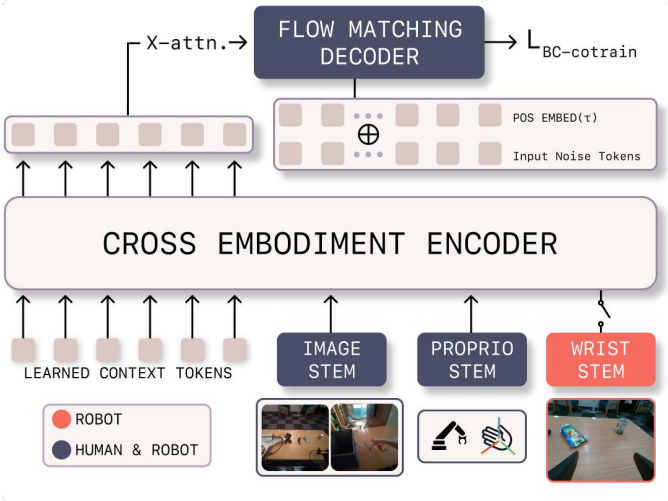

Fig. 7: **Model Architecture.** An illustration of our transformer-based cross-embodiment policy backbone.

is computed on the aggregated human and robot dataset.

$$\mathcal{L}_{\text{BC-cotrain}}(\phi, \theta) = \mathbb{E}_{(o,a)\sim D_H \cup D_R}[\mathcal{L}_{\text{BC}}(\pi_\theta(f_\phi(o)), a)]$$

In practice, per training step, for each embodiment $e \in \{\text{robot}, \text{human}\}$, we compute the conditional flow matching (CFM) loss on a mini-batch of human and robot samples:

$$\mathcal{L}_{\text{BC-cotrain}} = \mathcal{L}_{\text{CFM}}^{\text{robot}} + \mathcal{L}_{\text{CFM}}^{\text{human}}$$

The policy architecture is visualized in Fig. 7. Further details on the policy architecture and algorithm are included in Appendix.

D. Evaluation Setup

Rollout evaluation protocol. Evaluation is performed on four representative Flagship tasks shown in Fig. 8. We evaluate both in-domain (ID) settings, where task layouts match robot training data, and out-of-domain (OOD) settings with unseen objects and environments. For each method, we perform 20 ID and 20 OOD rollouts per task, with randomized initial conditions. Performance is measured using task-specific subtask metrics, including grasps, placements, intermediate manipulations, and full task completion. For ease of comparison, we report a **normalized score** aggregated across rollouts. Full task-specific protocols, rollout counts, scoring definitions, and detailed results are provided in the appendix.

Robot data collection. For each task, approximately 150-300 demonstrations are collected with randomized object placement, orientation and combinations across the workspace. For each task, 4-8 objects are selected from those recorded within `EgoVerse-A` to be used for demonstrations. Task specific details are provided in the appendix.

E. Does Human Data Scale Robot Performance?

We investigate whether human egocentric data can reliably improve robot learning and if performance scales as human data volume increases. Rather than optimizing for a single system, we test reproducibility across multiple robots, labs, and three flagship tasks: *object-in-container* (single-arm), *cup-on-saucer* (fine-grained bimanual), and *bag-grocery* (long-horizon bimanual). We co-train all models with a subset of `EgoVerse-A` data which includes diverse task-specific data and in-domain human data. In-domain human data is collected under the same task definitions as robot teleoperation, but differs in embodiment, sensing, and motion execution. Complete quantitative results are summarized in Fig. 9.

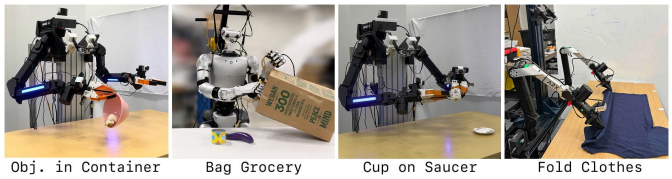

Fig. 8: **Evaluation Tasks.** We conduct evaluation with 4 representative Flagship tasks.

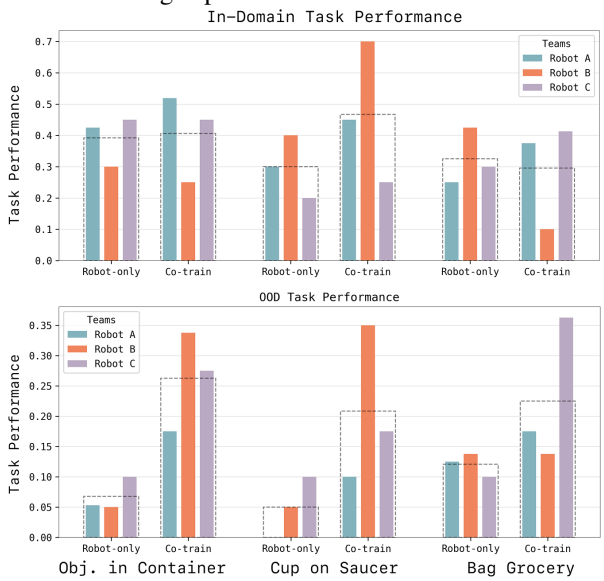

Fig. 9: **Co-training improves transfer.** Joint training with human egocentric data consistently improves in-domain performance and out-of-domain generalization across robots.

Co-training with `EgoVerse-A` improves robot performance and generalization. The presence of `EgoVerse-A` data consistently improves performance in both in-domain and out-of-domain (OOD) settings by up to 30%. It is important to note that the demonstration speed and reset range is different across each manipulation setting. Hence, it is more informative to compare relative improvements from adding human data as opposed to absolute scores. While results are robust across most embodiments and tasks, we observed a performance decrease in the *bag-grocery* task for `Robot B`, whereas `Robot A` and `Robot C` saw improvements. We hypothesize that this divergence stems from `Robot B`'s specific embodiment limitations, which forced robot demonstrations to deviate from the human strategies in `EgoVerse-A`. We include additional analysis of this observation in the Appendix.

Domain-aligned data enables transfer from diverse data. We additionally study whether robot performance scales with

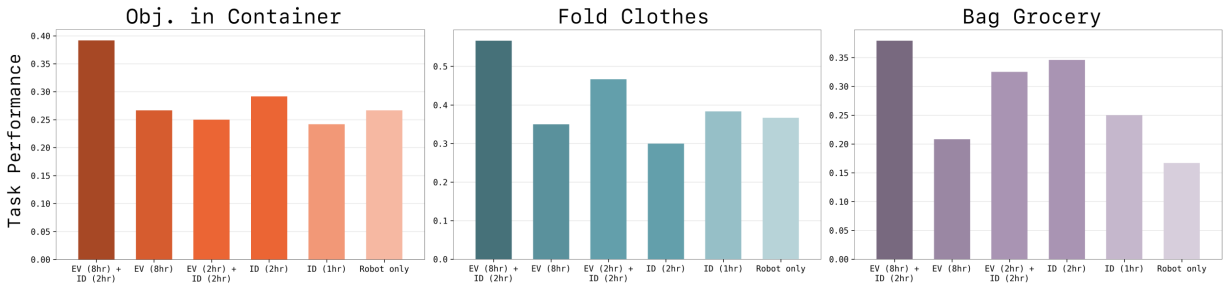

Fig. 10: **Domain-aligned data enables scaling.** We ablate the effect of EgoVerse-A (EV) and aligned human data (ID). A small amount of aligned human data anchors learning and allows performance to improve as diverse human data scale.

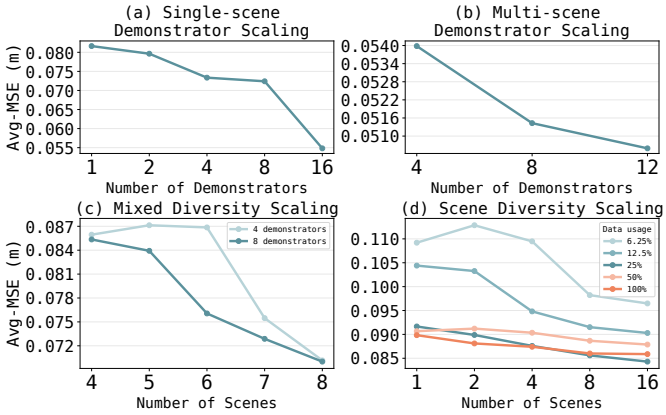

Fig. 11: **Controlled Diversity Results.** (a) Scaling demonstrators in a single scene improves generalization to unseen ones. (b) Demonstrator scaling remains beneficial across eight fixed scenes. (c) Jointly scaling scene and demonstrator diversity yields complementary improvements. (d) Increasing scene diversity improves generalization to unseen scenes across data budgets, with the strongest gains under limited data.

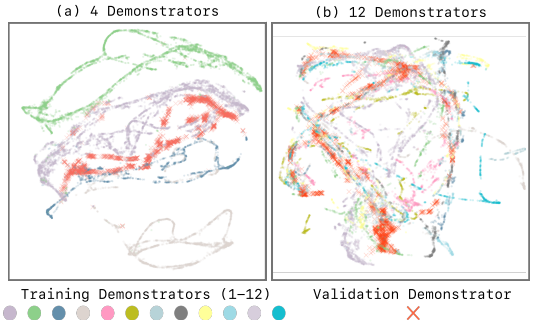

Fig. 12: **Demonstrator Diversity Visualization.** We visualize UMAP embeddings of encoded features for 4 and 12 demonstrators in the multi-scene demonstrator scaling setting, showing greater overlap between training and validation demonstrators with increased demonstrator diversity.

diverse human data for each task from EgoVerse-A. We scale the in-domain human data and diverse human data, while the robot data is kept fixed. For scaling to be effective, the policy must extract transferable structure from diverse human behavior. As shown in Fig. 10, we find that scaling benefits depend critically on the availability of aligned human–robot data. While neither 8h of diverse EgoVerse-A data nor domain-aligned human data alone is sufficient to drive significant performance gains in ID or OOD settings, we observe positive scaling when domain-aligned data "anchors" the learning process. This effect enables the policy to effectively transfer knowledge from diverse sources; for instance, the inclusion of just 2h of domain-aligned data facilitates transfer from 2h of diverse EgoVerse-A data, a trend that scales further as the diverse data volume increases to 8h.

F. How Does Human Data Diversity Affect Generalization?

The experiments above rely on naturally aggregated data from multiple labs, which introduces uncontrolled and potentially imbalanced diversity. Prior work has shown that different sources of diversity in robot datasets, such as environments and viewpoints, can have uneven effects on generalization [31, 26, 43, 58]. Motivated by these findings, we conduct controlled studies to isolate the effects of scene and demonstrator diversity in a human data collection setting described in Sec. III-D.We report primarily the offline Avg-MSE metric in the human-based evaluation setting. While this metric does not directly measure downstream robot performance, it provides a stable signal for comparing generalization across diversity and data scaling regimes [47]. Fig. 11 reports results for the *fold-clothes* task, with additional results in Appendix.

Demonstrator Diversity Scaling. We vary the number of human demonstrators while controlling for task and scene. We study two settings: (1) *Single scene*, where models are trained with $\{1, 2, 4, 8, 16\}$ demonstrators under a fixed 2-hour budget and evaluated on a held-out demonstrator in the same scene; and (2) *Multi scene*, where models are trained with $\{4, 8, 12\}$ demonstrators under a fixed 8-hour budget across 8 scenes and evaluated on held-out demonstrators in the same environments. As shown in Fig. 11(a,b), increasing demonstrator diversity consistently improves generalization to unseen demonstrators. To illustrate this effect, Fig. 12 visualizes UMAP embeddings of encoded features for $\{4, 12\}$ demonstrators, showing increased overlap between training and validation demonstrators with greater demonstrator diversity.

Scene Diversity Scaling. We vary the number of training scenes in $\{1, 2, 4, 8, 16\}$ while holding the demonstrator pool fixed, and evaluate on unseen scenes. As shown in Fig. 11(d), generalization performance improves as scene diversity in-

creases across data budgets. Once data quantity reaches a moderate level, increasing data density yields diminishing returns, whereas expanding scene coverage provides measurable gains. This suggests that beyond a certain scale, environmental diversity plays a more critical role in generalization than additional data collected within individual scenes.

Multi-Scene Interaction Effects. Finally, we jointly scale scene and demonstrator diversity, reflecting realistic EgoVerse data collection. We vary the number of training scenes in $\{4, 5, 6, 7, 8\}$ and demonstrators in $\{4, 8\}$ under a fixed 4-hour budget, and evaluate on unseen scenes and demonstrators. As shown in Fig. 11(c), increasing scene diversity improves generalization under both demonstrator budgets, while the marginal benefit of additional demonstrators decreases as scene coverage grows.

V. Limitations

Our study mainly focused on human-and-robot co-training. Future work should conduct broader algorithmic exploration (e.g., pre-train and fine-tune). Moreover, the scene and demonstrator diversity experiments rely exclusively on offline metrics. While these metrics capture generalization across human demonstrators and environments, additional robot rollouts are necessary to determine whether the observed diversity effects translate to improved generalization in robotic manipulation.

VI. Conclusion

In this work, we introduced EgoVerse, a collaborative framework for scalable human data-driven robot learning. Our paper presents a large-scale human dataset collected across academic and industry partners, with unprecedented diversity in task semantics, scenarios, and demonstrators. Unlike prior static datasets, EgoVerse is designed as a continuously growing resource that supports reproducible study. Beyond the dataset, our consortium-scale evaluation shows that human data improves robot performance, that aligned human–robot data are necessary to anchor effective scaling, and that scene diversity strongly affects generalization under limited data budgets.

VII. Contributions

Project Leads: Ryan Punamiya, Simar Kareer.

Lab Leads (coordinated major components of the project including data collection pipelines, infrastructure, experiments, and cross-team coordination): Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti.

Core Contributors (made substantive contributions to the scientific outcome including data collection, experiments, system development, analysis, and writing): Lawrence Y. Zhu, Patcharapong Aphiwetsa, Baoyu Li, Aniketh Chuleva, Pranav Kuppili, Yangcen Liu, Dhruv Patel, Aidan Gao, Hye-Young Chung, Ryan Co, Renee Zbizika, Jeff Jinyun Liu, Xiaomeng Xu, Haoyu Xiong, Geng Chen, Sebastiano Oliani, Chenyu Yang, Xi Wang.

Industry Partners (contributed engineering support, infrastructure, and dataset collaboration): James Fort, Richard Newcombe, Josh Gao, Jason Chong, Garrett Matsuda, Aseem Doriwala.

Academic PIs (provided project supervision and research direction): Marc Pollefeys, Robert Katzschmann, Xiaolong Wang, Shuran Song, Judy Hoffman, Danfei Xu.

We thank additional collaborators who supported data collection efforts in participating labs.

Georgia Tech Data Collection: Zhenyang Chen, Woo Chul Shin, Shuo Cheng, Liqian Ma, Xinchen Yin, Rohan Bansal, David He, Vaibhav Saxena, Mengying Lin, Nadun Ranawaka

ETH Zürich Data Collection: Wenkai Xuan, Aristotelis Sympetheros, Esteban Padilla Cerdio, Filippos Katsimalis, Robert Jomar Malate.

Industry Support: We thank collaborators at Meta Reality Labs Research, Mecka AI, and Scale AI for data contribution, engineering support, infrastructure assistance, and collaboration on dataset development.

References

[1] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild, 2022. URL https://arxiv.org/abs/2207.09450.

[2] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. *CVPR*, 2023.

[3] Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Fan Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard Newcombe, Robert Wang, et al. Introducing hot3d: An egocentric dataset for 3d hand and object tracking. *arXiv preprint arXiv:2406.09598*, 2024.

[4] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables generalizable robot manipulation. In *European Conference on Computer Vision (ECCV)*, 2024.

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*, 2024.

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

[7] 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.

[8] Boyuan Chen, Tianyuan Zhang, Haoran Geng, Kiwhan Song, William T. Freeman, Jitendra Malik, Russ Tedrake, Vincent Sitzmann, and Yilun Du. Large video planner, 2025. URL http://arxiv.org/abs/2512.15840.

[9] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024.

[10] 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*, 2024.

[11] 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. *arXiv preprint arXiv:2402.10329*, 2024.

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

[13] Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in neural information processing systems*, 36:9156–9172, 2023.

[14] Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, Cheng Peng, Chris Sweeney, Cole Wilson, Dan Barnes, Daniel DeTone, David Caruso, Derek Valleroy, Dinesh Ginjupalli, Duncan Frost, Edward Miller, Elias Mueggler, Evgeniy Oleinik, Fan Zhang, Guruprasad Somasundaram, Gustavo Solaira, Harry Lanaras, Henry Howard-Jenkins, Huixuan Tang, Hyo Jin Kim, Jaime Rivera, Ji Luo, Jing Dong, Julian Straub, Kevin Bailey, Kevin Eckenhoff, Lingni Ma, Luis Pesqueira, Mark Schwesinger, Maurizio Monge, Nan Yang, Nick Charron, Nikhil Raina, Omkar Parkhi, Peter Borschowa, Pierre Moulon, Prince Gupta, Raul Mur-Artal, Robbie Pennington, Sachin Kulkarni, Sagar Miglani, Santosh Gondi, Saransh Solanki, Sean Diener, Shangyi Cheng, Simon Green, Steve Saarinen, Suvam Patra, Tassos Mourikis, Thomas Whelan, Tripti Singh, Vasileios Balntas, Vijay Baiyya, Wilson Dreewes, Xiaqing Pan, Yang Lou, Yipu Zhao, Yusuf Mansour, Yuyang Zou, Zhaoyang Lv, Zijian Wang, Mingfei Yan, Carl Ren, Renzo De Nardi, and Richard Newcombe. Project aria: A new tool for egocentric multi-modal ai research, 2023. URL https://arxiv.org/abs/2308.13561.

[15] Haritheja Etukuru, Norihito Naka, Zijin Hu, Seungjae Lee, Julian Mehu, Aaron Edsinger, Chris Paxton, Soumith Chintala, Lerrel Pinto, and Nur Muhammad Mahi Shafiullah. Robot utility models: General policies for zero-shot deployment in new environments. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8275–8283. IEEE, 2025.

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

[17] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot. *arXiv preprint arXiv:2307.00595*, 2023.

[18] Raktim Gautam Goswami, Amir Bar, David Fan, Tsung-Yen Yang, Gaoyue Zhou, Prashanth Krishnamurthy, Michael Rabbat, Farshad Khorrami, and Yann LeCun. World models can leverage human videos for dexterous manipulation, 2025. URL https://arxiv.org/abs/2512.13644.

[19] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

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

[21] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024.

[22] Irmak Guzey, Haozhi Qi, Julen Urain, Changhao Wang, Jessica Yin, Krishna Bodduluri, Mike Lambeta, Lerrel Pinto, Akshara Rai, Jitendra Malik, Tingfan Wu, Akash Sharma, and Homanga Bharadhwaj. Dexterity from smart lenses: Multi-fingered robot manipulation with in-the-wild human demonstrations, 2025. URL https://arxiv.org/abs/2511.16661.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385.

[24] Zihao He, Bo Ai, Tongzhou Mu, Yulin Liu, Weikang Wan, Jiawei Fu, Yilun Du, Henrik I. Christensen, and Hao Su. Scaling cross-embodiment world models for dexterous manipulation, 2025. URL https://arxiv.org/abs/2511.01177.

[25] Ryan Hoque, Peide Huang, David J Yoon, Mouli Siva-

purapu, and Jian Zhang. Egodex: Learning dexterous manipulation from large-scale egocentric video. *arXiv preprint arXiv:2505.11709*, 2025.

[26] Yingdong Hu, Fanqi Lin, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. *arXiv preprint arXiv:2410.18647*, 2024.

[27] 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.

[28] Frederik Ebert Jedrzej Orbik. Oculus reader: Robotic teleoperation interface, 2021. URL https://github.com/rail-berkeley/oculus_reader. Accessed: YYYY-MM-DD.

[29] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. Egomimic: Scaling imitation learning via egocentric video, 2024. URL https://arxiv.org/abs/2410.24221.

[30] Simar Kareer, Karl Pertsch, James Darpinian, Judy Hoffman, Danfei Xu, Sergey Levine, Chelsea Finn, and Suraj Nair. Emergence of human to robot transfer in vision-language-action models, 2025. URL https://arxiv.org/abs/2512.22414.

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

[32] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos, 2025. URL https://arxiv.org/abs/2503.00779.

[33] Jialong Li, Xuxin Cheng, Tianshu Huang, Shiqi Yang, Ri-Zhao Qiu, and Xiaolong Wang. Amo: Adaptive motion optimization for hyper-dexterous humanoid whole-body control, 2025. URL https://arxiv.org/abs/2505.03738.

[34] Vincent Liu, Ademi Adeniji, Haotian Zhan, Siddhant Haldar, Raunaq Bhirangi, Pieter Abbeel, and Lerrel Pinto. Egozero: Robot learning from smart glasses. *arXiv preprint arXiv:2505.20290*, 2025.

[35] Yangcen Liu, Woo Chul Shin, Yunhai Han, Zhenyang Chen, Harish Ravichandar, and Danfei Xu. Immimic: Cross-domain imitation from human videos via mapping and interpolation, 2025. URL https://arxiv.org/abs/2509.10952.

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

[37] 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 pre-training from large-scale human videos. *arXiv preprint arXiv:2507.15597*, 2025.

[38] NVIDIA, :, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. Gr00t n1: An open foundation model for generalist humanoid robots, 2025. URL https://arxiv.org/abs/2503.14734.

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

[40] Ryan Punamiya, Dhruv Patel, Patcharapong Aphiwetsa, Pranav Kuppili, Lawrence Y Zhu, Simar Kareer, Judy Hoffman, and Danfei Xu. Egobridge: Domain adaptation for generalizable imitation from egocentric human data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[41] Ri-Zhao Qiu, Shiqi Yang, Xuxin Cheng, Chaitanya Chawla, Jialong Li, Tairan He, Ge Yan, David J. Yoon, Ryan Hoque, Lars Paulsen, Ge Yang, Jian Zhang, Sha Yi, Guanya Shi, and Xiaolong Wang. Humanoid policy ˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[42] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning, 2025. URL https://arxiv.org/abs/2501.06994.

[43] Vaibhav Saxena, Matthew Bronars, Nadun Ranawaka Arachchige, Kuancheng Wang, Woo Chul Shin, Soroush Nasiriany, Ajay Mandlekar, and Danfei Xu. What matters in learning from large-scale datasets for robot manipulation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[44] Junyao Shi, Zhuolun Zhao, Tianyou Wang, Ian Pedroza, Amy Luo, Jie Wang, Jason Ma, and Dinesh Jayaraman. Zeromimic: Distilling robotic manipulation skills from

web videos. In *International Conference on Robotics and Automation (ICRA)*, 2025.

[45] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

[46] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos. *arXiv preprint arXiv:1912.04443*, 2019.

[47] Generalist AI Team. Gen-0: Embodied foundation models that scale with physical interaction. *Generalist AI Blog*, 2025. https://generalistai.com/blog/nov-04-2025-GEN-0.

[48] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

[49] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers, 2024. URL https://arxiv.org/abs/2409.20537.

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

[51] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators, 2023.

[52] Haoyu Xiong, Quanzhou Li, Yun-Chun Chen, Homanga Bharadhwaj, Samarth Sinha, and Animesh Garg. Learning by watching: Physical imitation of manipulation skills from human videos. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7827–7834, 2021. doi: 10.1109/IROS51168.2021.9636080.

[53] Haoyu Xiong, Xiaomeng Xu, Jimmy Wu, Yifan Hou, Jeannette Bohg, and Shuran Song. Vision in action: Learning active perception from human demonstrations. *arXiv preprint arXiv:2506.15666*, 2025.

[54] Ruihan Yang, Qinxi Yu, Yecheng Wu, Rui Yan, Borui Li, An-Chieh Cheng, Xueyan Zou, Yunhao Fang, 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.

[55] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Sejune Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos, 2024. URL https://arxiv.org/abs/2410.11758.

[56] Jessica Yin, Haozhi Qi, Youngsun Wi, Sayantan Kundu, Mike Lambeta, William Yang, Changhao Wang, Tingfan Wu, Jitendra Malik, and Tess Hellebrekers. Osmo: Open-source tactile glove for human-to-robot skill transfer. *arXiv preprint arXiv:2512.08920*, 2025.

[57] Kevin Zakka. Mink: Python inverse kinematics based on MuJoCo, December 2025. URL https://github.com/kevinzakka/mink.

[58] Lihan Zha, Apurva Badithela, Michael Zhang, Justin Lidard, Jeremy Bao, Emily Zhou, David Snyder, Allen Z. Ren, Dhruv Shah, and Anirudha Majumdar. Guiding data collection via factored scaling curves, 2025. URL https://arxiv.org/abs/2505.07728.

[59] Lawrence Y. Zhu, Pranav Kuppili, Ryan Punamiya, Patcharapong Aphiwetsa, Dhruv Patel, Simar Kareer, Sehoon Ha, and Danfei Xu. Emma: Scaling mobile manipulation via egocentric human data. *IEEE Robotics and Automation Letters*, 11(3):3087–3094, 2026. doi: 10.1109/LRA.2026.3653320.

VIII. Appendix

A. Table of Contents

Appendix	Section
A	Table of Contents
B	Extended Discussion
C	EgoVerse Data Composition
D	Human Data Collection Setup Detail
E	Data Capture Hardware Detail
F	EgoDB Detail
G	Data Alignment and Post Processing
H	Robot Data Collection Setup Detail
I	Policy Architecture and Learning Detail
J	Robot Experiment Results
K	Controlled Diversity Experiments
L	Latent Space Visualization

TABLE I: Appendix table of contents with section links.

B. Extended Discussion

This work explores one concrete instantiation of learning from egocentric human data via joint human–robot co-training, but more importantly, it opens a broader design space for future research enabled by EgoVerse. A natural next direction is to systematically study different training paradigms beyond co-training, including large-scale pretraining on heterogeneous human data followed by targeted fine-tuning on small amounts of robot or aligned human–robot data. Understanding when pretraining yields transferable structure, and how embodiment-specific design decisions impact transfer, remains an open question that EgoVerse is well positioned to support.

Another promising avenue is leveraging the full breadth of unaligned, in-the-wild human data in `EgoVerse-I`. While our study focuses on more controlled `EgoVerse-A`, `EgoVerse-I` contains rich, open-ended manipulation data with dense language annotations. This creates opportunities for language-conditioned or goal-conditioned policies that can learn from broader task distributions without requiring strict alignment at training time. Developing methods that can exploit such weakly aligned or unaligned data, while still grounding execution through limited aligned supervision, is a key step toward scalable human-to-robot transfer.

Finally, the consortium-scale nature of EgoVerse enables deeper analysis of embodiment effects that go beyond aggregate performance trends. Future work can examine how specific embodiment factors, such as kinematic structure (e.g., gripper vs. dexterous hands), sensing configuration, or control parameterization, interact with different types of human data. Such analyses can inform principled data selection, curriculum design, and model architectures as datasets continue to grow. Taken together, EgoVerse is not only a dataset but a shared experimental substrate for studying how representation learning, embodiment, and data scale jointly shape the next generation of generalizable robot policies.

Part	Percent	# Hours	# Episodes	# Tasks
`EgoVerse-A` (all partners)	5.5%	75	2,385	6
`EgoVerse-I` partner A	76.1%	1,035	72,993	1,898
`EgoVerse-I` partner B	18.4%	250	3,128	45

TABLE II: Dataset composition across EgoVerse components.

Category	Percent	# Hours
Logistics	15.4%	209
Cooking	13.7%	186
Cleaning	11.6%	158
Laundry	10.9%	148
Hardware	6.8%	92
Crafts	4.0%	54
Gardening	3.2%	44

TABLE III: Task category distribution.

C. EgoVerse Data Composition

The amount of data contributed by each partner for `EgoVerse-A` and `EgoVerse-I` is summarized in Table II. While providing a per-task breakdown of hours across the full EgoVerse dataset is infeasible, we instead report aggregate statistics over semantic task categories and their corresponding dataset sizes in Table III. In addition, we provide a frequency distribution for the top 20 "verbs" across each category in Table IV.

D. Human Data Collection Setup Detail

`EgoVerse-A` is designed to capture diversity while maintaining controlled task semantics and data quality.

Scenario and object diversity. Each flagship task is performed across 8–12 unique scenes per lab, with 1–10 dataset units collected per scene to capture within-environment variation. Within each scene, a roughly $40\text{cm} \times 60\text{cm}$ workspace is used, and collectors are explicitly encouraged to randomize object positions across demonstrations. Within each lab, objects are sampled from a fixed set of up to 30 objects per task. As objects are procured independently across participating sites, the dataset exhibits broad variation in real-world object geometry, appearance, and material properties.

Demonstrator diversity. Data are collected from 1–8 demonstrators per lab. Even under identical instructions and tutorial videos, demonstrators exhibit distinct motion habits, timing, coordination strategies, and hand trajectories. The dataset therefore captures substantial variation in human morphology and egocentric viewpoints, arising naturally from differences in collector height, posture, and workspace configuration (e.g., seated versus standing).

Prior work has shown that such human motion variability can significantly affect policy learning behavior and cross-embodiment alignment [29, 40, 35]. Rather than eliminating this variation, we treat it as an inherent component of human data that must be managed as datasets scale. While all participating labs follow the same instruction protocol and task definitions, each lab contributes a distinct distribution of environments, objects, and human behaviors.

Logistics	Freq.	Cooking	Freq.	Cleaning	Freq.	Laundry	Freq.	Hardware	Freq.	Crafts	Freq.
pick	34,524	pick	16,197	scrub	20,320	pick	6,998	pick	16,387	pick	7,577
scoop	12,164	place	6,678	pick	12,297	fold	6,190	place	5,476	place	2,517
place	10,322	cut	5,975	clean	7,713	iron	5,342	adjust	5,263	adjust	1,835
fill	7,991	scoop	4,722	wipe	6,863	adjust	3,628	remove	3,305	fold	1,059
adjust	7,802	adjust	4,496	dip	5,951	place	3,325	unscrew	3,222	cut	896
seal	4,249	fill	2,654	place	5,480	smooth	2,429	hold	2,104	hold	589
open	3,392	hold	2,153	adjust	5,225	straighten	1,161	tighten	2,052	move	467
put	3,064	slice	1,974	hold	3,358	flip	763	clean	1,564	apply	434
hold	3,041	remove	1,687	remove	2,862	smoothen	751	test	1,253	attach	391
insert	1,996	press	1,646	wash	2,301	hold	560	put	1,196	press	367
close	1,426	put	1,215	rinse	1,901	clean	487	scrape	1,068	align	352
fold	1,184	move	1,088	scrape	1,877	grab	463	fill	1,047	put	347
transfer	1,118	open	1,066	brush	1,752	spread	439	solder	908	wrap	316
move	1,117	seal	1,060	thread	1,647	move	372	move	786	shake	269
clean	868	trim	964	put	850	button	349	install	647	tie	239
pack	760	flatten	803	pull	668	lift	321	insert	620	smooth	235
wipe	749	chop	729	tie	607	unfold	314	apply	585	shape	234
pour	686	transfer	673	polish	576	insert	300	screw	574	twist	216
remove	611	fold	632	move	558	press	295	open	569	remove	214
add	586	knead	624	rotate	524	trim	289	secure	568	insert	200

TABLE IV: Top 20 verbs per category with frequencies. Each row corresponds to the same rank across six task categories.

`EgoVerse-I` is designed to capture a wide variety of demonstration-style tasks and behaviors across extremely diverse settings and objects. To ensure that the data is maximally useful for robot learning, we reduce noise as much as possible by enforcing that the hands remain visible in the scene whenever feasible and that demonstrators follow a consistent strategy for each sub-task. We additionally instruct demonstrators to be decisive in their actions, avoiding unnecessary hesitation or corrective motions, so that the resulting trajectories exhibit clear intent and well-defined temporal structure.

E. Human Data Capture Detail

Project Aria Glasses. We use the Project Aria Gen 1 glasses for egocentric human data capture in `EgoVerse-A`. The Gen 1 device weighs approximately 75 g, designed to be worn comfortably for hours without significantly altering natural human behavior. The hardware integrates five cameras: one forward-facing global-shutter RGB cameras for egocentric scene capture, two side-facing monocolor camera for SLAM and hand tracking, and two inward-facing cameras for eye tracking. In EgoVerse, we use the primary forward-facing RGB camera as the main visual stream for learning, providing a stable first-person view closely aligned with human manipulation intent and task execution. The device also includes a tightly synchronized IMU, enabling accurate visual–inertial sensing. All raw sensor streams are processed through Meta's MPS (Machine Perception Services) pipeline, which performs calibration, temporal alignment, and visual–inertial odometry to produce metrically consistent camera poses and synchronized RGB streams in a world-referenced frame.

Phone-based Human Data Capture System. To broaden access to large-scale human data collection beyond specialized wearable devices, we introduce a phone-based egocentric capture system built on commodity smartphones (Fig. 13). The setup mounts a smartphone on a lightweight head strap and records egocentric RGB video using the ultrawide camera at 1080p and 30 FPS. This configuration preserves a wide field of view over the workspace while remaining inexpensive, easy to deploy, and comfortable for extended use. Recorded videos are uploaded to a cloud processing system that estimates 6-DoF head motion via visual tracking and recovers 3D hand pose with 21 keypoints per hand. The resulting signals are temporally synchronized and converted into the same canonical representation used throughout EgoVerse, including egocentric video, camera motion, and hand trajectories. By matching the output format of more instrumented capture systems, this phone-based pipeline allows data from resource-constrained contributors to be directly integrated into the broader dataset without special handling or separate learning recipe.

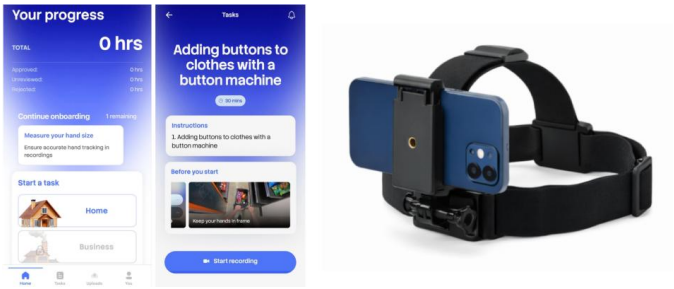

Fig. 13: **Phone-based Data Collection System.** (Left) Screenshot of the accompanying app for the iPhone-based human data collection system. (Right) The setup consist of an off-the-shelf head strap phone mount and an iPhone.

Custom Capture Hardware. EgoVerse is meant to be compatible with a variety of human data sources, including highly customized capture systems designed for large-scale industrial data collection. As a concrete example, one industry partner contributes data collected using a custom stereoscopic head-mounted rig built from a pair of fisheye RGB cameras

configured in a coplanar, collinear stereo setup with a fixed 6 cm baseline. The system records synchronized RGB video at 1920×1200 resolution and 30 FPS, paired with depth streams derived from stereo reconstruction and tightly time-aligned inertial measurements from an onboard IMU. Multi-sensor SLAM integrates RGB, depth, and inertial signals recovers metrically consistent 6-DoF head pose trajectories. Hand pose is estimated using vision-based models enriched with stereo depth to recover anatomically plausible 3D hand motion, from which 21 keypoints per hand are extracted and temporally smoothed. Although this rig is substantially more instrumented than the phone-based setups, all outputs are converted into the same EgoVerse canonical format, allowing data collected at industrial scale to be seamlessly unified with other sources for downstream robot learning.

F. EgoDB Detail

EgoDB comprises of four main components, the data collection and uploading, the SQL database, S3 based automated processing and EgoVerseDataset for training.

1) Data Collection and Uploading: `EgoVerse-A` data collected using the Aria glasses in the form of *.vrs* files and robot data in lab-specific formats are uploaded using a unified uploading script. At upload time, data collectors are asked to annotate `operator`, `lab`, `task`, `embodiment`, `robot_name`, `scene`, `objects`, and `is_eval` according to the schema summarized in Table V. The raw files are hashed using UTC timestamps and uploaded to the S3 bucket alongside *.json* files containing the annotated metadata. An hourly daemon running on a lightweight `t3.xlarge` AWS EC2 instance checks the bucket for new files and metadata and updates the SQL database accordingly.

2) SQL Database: The SQL database is a Postgres SQL table with rows that correspond to the schema in Table V and enables easy filtering. Each row of the SQL table is a single file.

3) Ray Processing Daemon: `EgoVerse-A` and Robot data are processed and have their metadata updated by nightly Ray processing daemons. The daemon consists of 3 Ray clusters.

Project Aria Data. For the `EgoVerse-A` data, Cluster A responsible for running MPS (Machine Perception Services) runs on a single head node (*t3a.2xlarge*). It syncs batches of files without corresponding MPS folders and makes nightly parallelized API calls to the MPS platform. After MPS is completed, it syncs back the processed MPS folders containing the data to the S3 bucket. Cluster B is responsible for converting the MPS output into the training ready processed format. This cluster has a *t3a.2xlarge* head node and *r6a.2xlarge* worker nodes. The head node checks the SQL table for the entries without a `processed_path` and with corresponding MPS folders and schedules jobs on worker nodes to be processed.

Robot Data. For the robot data, Cluster C is responsible for converting the raw robot data into the training ready processed format. This cluster has a *t3a.2xlarge* and *c5.18xlarge* worker nodes. The head node checks the SQL table for the entries without a `processed_path` and with corresponding MPS folders and schedules jobs on worker nodes to be processed.

Field	Description
`episode_hash`	Unique identifier for the episode, derived from UTC timestamps at upload time.
`operator`	Identifier for the human operator or demonstrator.
`lab`	Data collection site or partner lab.
`task`	Canonical task name (e.g., *bag-grocery*, *fold-clothes*).
`embodiment`	Embodiment type (human, robot platform, etc.).
`robot_name`	Robot platform identifier, if applicable.
`num_frames`	Number of frames in the processed trajectory (updated post-processing).
`task_description`	Free-form natural language description of the task.
`scene`	Scene or environment identifier.
`objects`	Serialized list of objects involved in the episode.
`processed_path`	S3 path to processed data artifacts (updated post-processing).
`processing_error`	Error message logged during automated processing, if any.
`mp4_path`	S3 path to rendered visualization video, if available.
`is_deleted`	Flag indicating whether the episode has been removed or deprecated.
`is_eval`	Flag indicating whether the episode is from policy evaluation.
`eval_score`	Scalar evaluation score, if applicable.
`eval_success`	Binary indicator of task success during evaluation.

TABLE V: Schema of the EgoDB episode table. Fields in italics are updateable during automated processing.

The worker nodes are additionally responsible for producing an mp4 of the processed data, and return a processed path, an error if applicable and number of frames in the file to the head node. Once the worker nodes process the files, the corresponding head nodes update the following sections of the SQL table, `processed_path`, `mp4_path`, `processing_error` and `num_frames` for those files.

4) EgoVerseDataset: We provide a unified dataset interface, `EgoVerseDataset`, for loading EgoVerse data directly from S3 into training-ready PyTorch datasets. The dataset enables scalable, filtered access to large collections of processed episodes across embodiments, tasks, and labs.

`EgoVerseDataset` resolves valid episodes by querying the SQL database using user-specified filters (e.g., task, lab, scene, robot name) and retrieves only episodes with a populated `processed_path`. Matching episodes are synchronized from S3 to a local cache directory, with safeguards to avoid re-downloading episodes that are already present locally.

The dataset downloading is parallelized using the *s5cmd* library which is a Python parallelized syncing tool for AWS S3. Each episode is loaded as an independent dataset instance, and dataset construction is parallelized across episodes to reduce load time. Episodes whose embodiment metadata does not match the requested embodiment are automatically skipped. The resulting collection of episodes is split into training and validation subsets according to a configurable validation ratio, or alternatively subsampled using a percent-based split.

The dataset supports multiple operating modes, including `train`, `valid`, `total`, and `percent`, enabling consistent dataset construction for both standard training and scaling experiments. All loaded episodes are exposed through a unified interface, allowing downstream training code to treat the aggregated data as a single iterable source.

5) Accessing Data from S3.: Given a set of metadata filters, data are resolved from the SQL database, synchronized from S3, and instantiated as PyTorch dataset objects. Code block 1 shows a simplified example illustrating this process.

Listing 1: Simplified example illustrating SQL-based episode resolution, S3 synchronization using `s5cmd`, and instantiation of training datasets.

```
# 1. Query SQL table to resolve processed episodes
filters = {
    "robot_name": "robot_a",
    "lab": "lab_a",
    "task": "task_x",
    "is_deleted": False,
}
rows = query_sql_table(filters)
# rows: [(processed_path, episode_hash), ...]

# 2. Download processed data from S3
for processed_path, episode_hash in rows:
    local_dir = f"/tmp/egoverse/{episode_hash}/"
    s3_src = f"{processed_path}/*"
    run_command(["s5cmd", "sync", s3_src, local_dir])

# 3. Instantiate dataset objects
datasets = []
for _, episode_hash in rows:
    dataset = SingleEgoVerseDataset(
        root=f"/tmp/egoverse/{episode_hash}",
        mode="train",
    )
    datasets.append(dataset)

# 4. Combine datasets for training
train_dataset = MultiEgoVerseDataset(datasets)
```

G. Data Alignment and Post Processing

Due to inherent differences in execution speed between humans and robot teleoperation, we apply post-processing to temporally align action trajectories across embodiments. For human demonstrations, we extract a 1-second window of actions and resample it to a sequence of length $T = 100$. For robot data, we extract a 1.5-second window and similarly resample it to length 100. This ensures that all trajectories represent a comparable phase of task execution despite differences in control frequency and execution speed. We apply linear interpolation for the 3D positions in the actions and spherical linear interpolation (SLERP) for quaternion and euler angle rotation representations.

H. Robot Data Collection Setup Detail

1) Hardware Setup:

a) `Robot A`: We employ a VR teleoperation system using the Meta Oculus 3 headset and Oculus Pro controllers based on the RAIL Lab Oculus Reader [28]. We use inverse kinematics on the commanded robot base frame end-effector using the Mink IK Solver [57] to obtain joint angles. The joint angles are executed by the ARX5 Joint Space Controller. The system is implemented using a multi-threaded Python setup, while the low-level hardware communication is handled via a CAN-based interface. The robot platform consists of two off-the-shelf ARX5 robot arms mounted as shown in Fig. 6 using off-the-shelf Vention aluminum beams with 3D-printed connectors.

b) `Robot B`: We employ a customized 3D-printed GELLO [51] device for teleoperation, with a one-to-one joint mapping between the GELLO interface and the ARX robot. The robot platform consists of two off-the-shelf ARX5 robot arms mounted left and right on a rigid 4040 aluminum frame with 3D-printed connectors as shown in Fig. 6. The gripper opening width is controlled via a trigger mounted at the distal end of each GELLO arm. The system is implemented using ROS 2 for message passing and inter-module communication. Low-level hardware communication is handled via a CAN-based interface, while motion execution is managed by the standard ARX5 Cartesian Controller.

c) `Robot C`: We use a Unitree G1 robot equipped with Inspire 6dof hands for experiments. The robot is equipped with a 3 DoF actuated head [33]. An Apple Vision Pro [9] for teleoperation, which maps the left and right wrist poses to the robot left wrist and right wrist, and use fingertip keypoints for retargeting. The head motors track the rotation fo the teleoperator's head via IK.

The controller runs on the Jetson Orin NX which is built in to the G1 robot. We implemented a python interface that communicates with the Unitree C++ backend via LCM. For inference, we deploy the model on a remote server, and use websocket for synchronized states and images communication.

2) Robot Data Composition: The per-task amount of robot demonstrations per robot and per task is summarized in Table VI.

I. Policy Architecture and Learning Detail

All learning hyperparameters are specified in Table VII

1) Cross Embodiment Encoder and Stems:

Task	# Demos — # Hours		
	Robot A	Robot B	Robot C
object-in-container	100 — 1.2	200 — 2.7	240 — 3.0
bag-grocery	300 — 5.1	150 — 1.67	139 — 1.8
cup-on-saucer	360 — 3.3	183 — 1.0	111 — 1.2
fold-clothes	300 — 3.0	–	–

TABLE VI: Robot dataset composition across tasks and platforms, reported as number of demonstrations and total hours.

a) ***Vision Inputs****:* Given an RGB frame $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, we apply ImageNet normalization an d then pass it through a ResNet-18 encoder truncated *before* the global-pool layer to obtain a $7 \times 7 \times 512$ feature map. We flatten the last convolutions and project them to d_{proj} with a single linear layer. We cross-attend L learnable query tokens of dimension d to the projected features using a single multi-head cross attention block with D_{stem} heads.

b) ***Proprioceptive Inputs****:* The proprioceptive observation vector $\mathbf{q} \in \mathbb{R}^{d_q}$ (joint angles, end effector pose etc.) is quantile normalized and passed through a single linear layer of hidden dimension d_{proj}. A single multi-head cross attention block with D_{stem} heads attends L query tokens with hidden dimension d to the projected proprioceptive data.

c) ***Encoder****:* Input observation data from m stems produces $m \times L$ query tokens. These tokens are concatenated along the sequence dimension. A set of learnable context tokens M are prepended to the token sequence. The new sequence of $M + m \cdot L$ is the input to the cross embodiment encoder. The cross embodiment encoder f_ϕ is a multi-block transformer encoder without masking. It consists of D_{enc} heads, N_{enc} blocks, and an embedding dimension of d. Each head has an attention dimension of d/D_{enc}.

2) Flow Matching Decoder.: The decoder is parameterized by a multi-block diffusion transformer with N_{dec} layers, D_{dec} attention heads, and embedding dimension d_{dec}. The M context tokens produced by the encoder are used as the conditioning sequence for the flow-matching decoder.

A noise token sequence of shape $\mathbb{R}^{T \times d_{\text{dec}}/2}$ is combined with a learnable positional embedding. A sine–cosine embedding of the continuous time variable $\tau \sim \text{Beta}(1.5, 1.0)$ is expanded to $\mathbb{R}^{T \times d_{\text{dec}}/2}$ and concatenated along the hidden dimension, yielding a token sequence of dimension $\mathbb{R}^{T \times d_{\text{dec}}}$. During training, the noise token sequence is constructed by sampling i.i.d. Gaussian noise $a_0 \sim \mathcal{N}(0, I)$ with shape $\mathbb{R}^{T \times d_a}$ and sampling a continuous timestep $\tau \in (0, 1]$. The decoder input is formed by linear interpolation between noise and the ground-truth action sequence $a_1 = \mathbf{a}$,

$$x_\tau = \tau a_0 + (1 - \tau) a_1,$$

which is then projected into the decoder token space.

At inference time, the noise token sequence is initialized as pure Gaussian noise $x_{\tau=1} \sim \mathcal{N}(0, I)$. The decoder is applied iteratively while integrating the learned velocity field from $\tau = 1$ to $\tau = 0$ using fixed-step Euler updates, producing the final action sequence at the end of integration. 10 fixed step Euler updates (inference steps) are used during evaluation.

The resulting token sequence is processed by a stack of transformer blocks with alternating self-attention and cross-attention. In the cross-attention blocks, the action tokens attend to the M conditioning tokens from the encoder. A final linear projection maps the hidden tokens to the predicted action sequence in $\mathbb{R}^{T \times d_a}$.

3) Co-training with Flow Matching.: As discussed earlier, the total co-training loss is defined as

$$\mathcal{L}_{\text{BC-cotrain}} = \mathcal{L}_{\text{CFM}}^{\text{robot}} + \mathcal{L}_{\text{CFM}}^{\text{human}}.$$

For a given embodiment e, we sample a timestep $\tau \sim \text{Beta}(1.5, 1.0)$ and minimize the error in the predicted vector field:

$$\mathcal{L}_{\text{CFM}}^{e} = \mathbb{E}_{\tau, a_0, a_1, s} \Big[\big\| \pi_\theta(x_\tau, \tau, f_\phi(s)) - (a_0 - a_1) \big\|^2 \Big],$$

where $x_\tau = \tau a_0 + (1 - \tau) a_1$ denotes the linear probability path between Gaussian noise a_0 and the target action a_1.

4) Training Details.: We train the model for 150,000 optimization steps with a global batch size of 32–64 and learning rate of 1×10^{-4}. Since experiments are conducted across multiple labs and platforms, the available compute resources and total training time vary.

All model hyperparameters are summarized in Table VII.

J. Robot Experiment Results

1) Training Mixture Details: We summarize training data mixtures for the various results reported in the paper below.

Flagship Co-train (EV(8hr) + ID(2hr)): For each task, we use a fixed co-training setup that combines 8 hours of `EgoVerse-A` (EV) human data with 2 hours of in-domain (ID) human data, together with task-matched robot demonstrations. The in-domain human data are collected using the same objects, task configuration, and scene as the robot dataset for the particular task. The `EgoVerse-A` data are aggregated across four collection sites for that task. Human samples are uniformly drawn from the union of EV (8hr) and ID (2hr) data. Robot data are sampled uniformly from the available demonstrations for the task and platform, as summarized in Table VI, using frame-level sampling.

Scaling Law Experiments: This experiment is to ablate the effect of each human data component: In Domain Human (ID) and `EgoVerse-A` Human (EV). The main result is discussed in Sec. IV-E. Here we explain the data composition for each training setup.

- *EV(8hr)*: Flagship Co-train training mixture without the in-domain human data.
- *EV(2hr) + ID(2hr)*: Flagship Co-train model training mixture with only 1 site's `EgoVerse-A` data.
- *ID(2hr)*: Flagship Co-train model training mixture without the EV(8hr) data.
- *ID(1hr)*: Flagship Co-train model training mixture without the EV(8hr) data and in-domain data with demonstrations subsampled to 1 hour.

Hyperparameter	Value
Observation Stems	
Visual Feature Embeddings	ResNet-18
m (number of stems)	4 (1 vision main camera, 2 vision wrist camera + 1 proprioception)
L (query tokens per stem)	16
D_{stem}	8
d_{proj}	256
Cross-Embodiment Encoder	
M (context tokens)	64
N_{enc}	16
D_{enc}	8
d	256
Encoder positional embedding	sine-cosine
Encoder normalization	Pre-LN
Flow Matching Decoder	
N_{dec}	6
D_{dec}	4
d_{dec}	128
d_a	task-specific
Time embedding	sine–cosine
τ distribution	$\text{Beta}(1.5, 1.0)$
Probability path interpolation	linear
Training	
Optimizer	AdamW
Learning rate	1×10^{-4}
Weight decay	1×10^{-4}
Training steps	150,000
Global batch size per embodiment	32–64
Human : robot batch ratio	1 : 1
Inference	
ODE solver	Euler
Integration steps	10
Integration interval	$\tau : 1 \to 0$
Initial noise	$\mathcal{N}(0, I)$

TABLE VII: Model hyperparameters. Symbols follow the notation introduced in Appendix VIII-J.

- *Robot only*: Flagship Co-train model training mixture without any human data.

For both the Flagship Co-training and the Scaling Law experiments, we use a global batch size of 32 with a fixed 1:1 human-to-robot ratio, i.e., 16 human samples and 16 robot samples per batch. This co-training recipe is used for all flagship experiments; variations are described separately in the scaling-law experiments.

2) Rollout Evaluation Protocol: We provide more detail of standardized evaluation protocol shared across different labs (robots) below. Images of the training and evaluation objects are in Fig 14.

a) ***object-in-container****:* The scene contains one object and one container randomly place with the object being closer to the robot vertically than the container.

In-domain: Randomly sample 20 positions across the workspace grid for object and container. For the first 10 positions, use 1 object-container combination from training set and for the next 10 use another object-container combination.

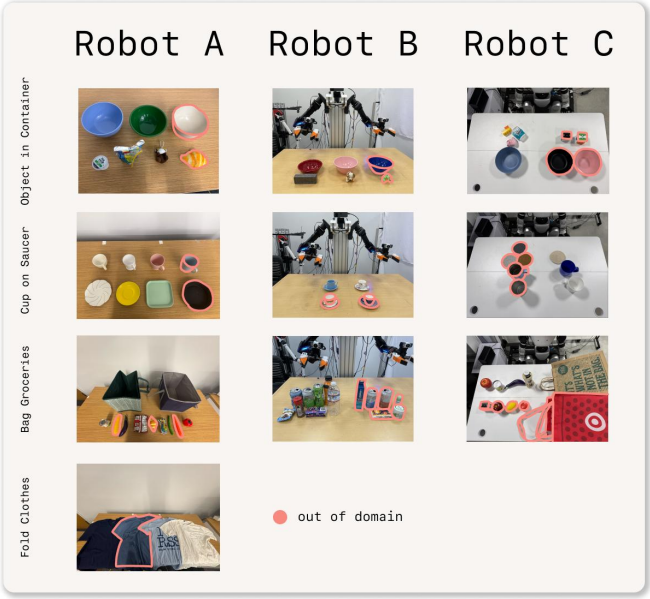

Fig. 14: Objects used for training and evaluation across tasks and robot platforms.

Perform one rollout for each position.

Out-of-domain: Sample an object-container combination from the human dataset that is not present in the robot dataset. For the first 10 rollouts, randomly sample 10 positions on the in-domain table. For the second 10 rollouts, randomly sample 10 positions on an unseen table. Perform one rollout for each position. Run each rollout for 40 seconds.

Termination: Rollouts are terminated early if the policy becomes stuck, exhibits unsafe behavior, or is unable to continue execution.

Scoring: Score the policy 1pt for each successful object placement in container and 1pt for each successful emptying of the container. Report total points obtained within 40 seconds.

b) ***cup-on-saucer****:* The scene starts with a cup placed to either the right or left of the workspace and a saucer placed on the opposite side.

In-domain: For each case (cup on left and cup on right), randomly sample 10 positions across the workspace grid, yielding 20 total rollouts. At each position, randomly select a cup–saucer combination from the seen object set and apply a random planar rotation sampled uniformly from $\pm 30°$. Perform one rollout per position.

Out-of-domain: For each case (cup on left and cup on right), randomly sample 5 positions on the in-domain table and 5 positions on an unseen table, yielding 10 total rollouts per case. At each position, randomly select a cup–saucer combination from the seen object set and apply a random planar rotation sampled uniformly from $\pm 30°$. Perform one rollout per position.

Termination: Rollouts are terminated early if the policy becomes stuck, exhibits unsafe behavior, or is unable to continue execution.

Scoring: Assign 1 point for successfully rotating and picking up the cup, 1 point for a successful handover, and 1 point for placing the cup on the saucer. Additionally, report the total task success rate (SR), defined as the percentage of rollouts in which the cup is correctly placed on the saucer.

c) ***bag-grocery****:* The scene contains a bag and three grocery objects. The bag is placed to the left of the robot, and the three objects are placed to the right.

In-domain: Randomly sample 20 positions across the workspace grid for the bag and the three objects. For each position, sample a set of 3 objects from the training set and a random permutation (ordering) of those objects. Perform one rollout per position.

Out-of-domain: Randomly sample 10 positions across the workspace grid. For the first 5 positions, sample one random 3-object set; for the next 5 positions, sample a second random 3-object set. For each position, randomize the permutation (ordering) of the objects. Repeat the same protocol on an unseen scene.

Termination: Rollouts are terminated early if the policy becomes stuck, exhibits unsafe behavior, or is unable to continue execution.

Scoring: Assign 1 point for successfully opening the bag and 1 point for each object placed in the bag (3 points), for a maximum of 4 points per rollout. A placement is considered successful only if the objects are placed in left-to-right order inside the bag.

d) ***fold-clothes****:* The scene contains a t-shirt placed horizontally, with the collar oriented to the right of the robot.

In-domain: Randomly select 2 t-shirts from the training set. Randomly sample 10 positions across the workspace grid and apply a random planar rotation from $\{0°, 30°, -30°\}$. For each sampled position, perform one rollout for each of the 2 selected t-shirts (2 rollouts per position).

Out-of-domain: Randomly select 1 t-shirt that appears only in the human dataset (and is not present in the robot dataset). Randomly sample 10 positions across the workspace grid and apply a random planar rotation from $\{0°, 30°, -30°\}$. Perform one rollout per position. Repeat the same protocol on an unseen scene.

Termination: Rollouts are terminated early if the policy becomes stuck, exhibits unsafe behavior, or is unable to continue execution.

Scoring: Assign 1 point for successfully completing the bottom-sleeves folding stage, 1 point for successfully completing the top-sleeves folding stage, and 1 point for folding the t-shirt in half, for a maximum of 3 points per rollout. Full completion of all three stages denotes task success; report task success rate (SR) as the percentage of rollouts that achieve full completion.

Score Normalization We compute normalized score for *bag-grocery* and *object-in-container* to standardize comparisons across different execution speeds across labs. We report success rates for fold clothes and cup on saucer. We calculate normalized score as total points scored divided by maximum possible points.

Fig. 15: Differences in task execution strategies for the *bag-grocery* task across embodiments.

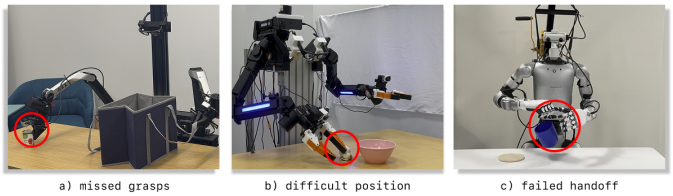

Fig. 16: Common failure modes for each task

3) Additional Analysis: As discussed in Sec. IV-E, `Robot B` exhibits a systematic strategy mismatch between human and robot demonstrations, which we hypothesize contributes to the observed degradation in co-training performance. This mismatch is illustrated in Fig. 15. In both the `EgoVerse-A` human demonstrations and the `Robot A` robot demonstrations, the bag is first opened using two hands (or grippers), after which grocery items are inserted. In contrast, the `Robot B` demonstrations employ a different strategy, where one gripper is used to prop the bag open while the other performs item insertion. This divergence leads to inconsistent behavior distributions between human and robot data, potentially weakening cross-embodiment alignment during co-training.

4) Task Failure Modes: For *object-in-container* and *bag-grocery*, we saw difficulty with picking primitives in certain parts of the workspace. While our co-trained policies generally exhibited more robust grasping primitives, there is still room for improvement. For the *cup-on-saucer* task, the object handover was difficult, especially for Robot C with a dexterous hand. Common failure modes are shown in Fig 16.

K. Controlled Diversity Experiments

We provide detailed experimental protocols and data compositions for the controlled diversity studies. While Sec. IV-F presents results on *fold-clothes*, this section presents the corresponding results for *cup-on-saucer* and a precise training-validation setting used across all four scaling regimes.

Data Composition and Evaluation Protocols. Experiments are conducted using the controlled-diversity subset described in Sec. III-D, consisting of 16 demonstrators and 16 scenes for each task. Following the same policy architecture and training procedure described in Sec. VIII-J, we vary the training data by constructing different demonstrator–scene pair combinations and evaluate human open-loop action prediction accuracy using the offline Avg-MSE metric: given a predicted action sequence $\hat{\mathbf{a}}_{1:T} \in \mathbb{R}^{T\times D}$ and ground-truth sequence $\mathbf{a}_{1:T} \in \mathbb{R}^{T\times D}$, we compute the mean-squared error at each

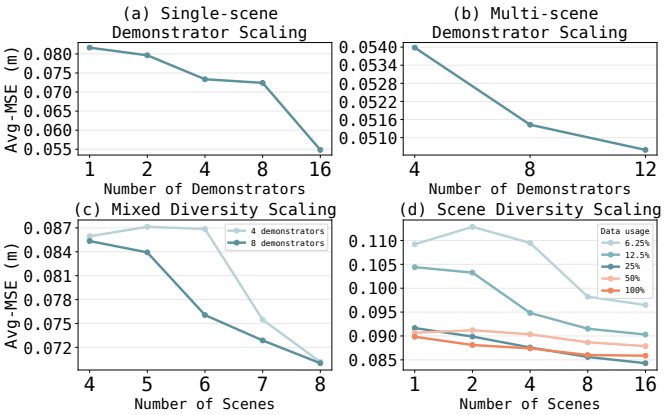

Fig. 17: **Controlled diversity results on fold-clothes.**

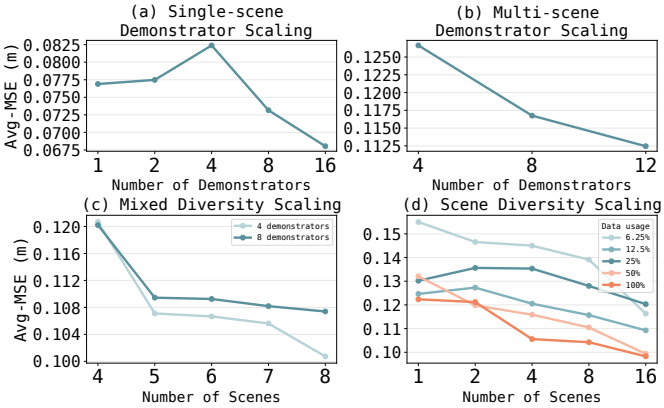

Fig. 18: **Controlled diversity results on cup-on-saucer.**

timestep and average over the sequence and action dimensions:

$$\text{Avg-MSE}(\hat{\mathbf{a}}_{1:T}, \mathbf{a}_{1:T}) = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{D}\left\|\hat{\mathbf{a}}_t - \mathbf{a}_t\right\|_2^2,$$

where we report this value averaged across validation episodes. We provide the detailed training data budgets for single-scene demonstrator scaling, multi-scene demonstrator scaling, mixed diversity scaling, and scene scaling in Table VIII, Table IX, Table X, and Table XI, respectively.

Analysis of Controlled Diversity Experiments. We present the results for *fold-clothes* and *cup-on-saucer* in Fig. 17 and Fig. 18, respectively.

1) Single-scene Demonstrator Scaling: This experiment studies whether adding motion diversity from increasing the number of demonstrators, given a fixed data budget of 2 hours, improves generalization towards unseen demonstrator at the same scene. As shown in Figs. 17(a), 18(a), increasing the number of demonstrators improves performance across both tasks. For *fold-clothes*, Avg-MSE decreases monotonically with demonstrator count, with larger gains at higher diversity levels, indicating that demonstrator variation meaningfully enhances coverage of state-action distribution. For *cup-on-saucer*, performance is slightly non-monotonic at low counts but improves at larger scales, suggesting that sufficient demonstrator diversity is needed to yield stable gains.

2) Multi-scene Demonstrator Scaling: In this study, we extend demonstrator scaling from a fixed single scene to eight scenes to examine whether the scaling effect persists in a multi-scene setting, aligning more closely with our real-world data collection setup. It is evaluated on unseen demonstrators within the same scenes. Given a fixed training data budget of 8 hours, the multi-scene demonstrator scaling results (Figs. 17(b), 18(b)) show consistent and clear improvements in generalization to unseen demonstrators as the number of demonstrators increases from 4 to 12 across both tasks.

3) Scene Diversity Scaling: Next, we assess how scene diversity and per-scene data allocation affect scene generalization, evaluated on unseen scenes data collected from other labs. In both tasks, increasing the number of scenes consistently reduces Avg-MSE, demonstrating that scene diversity improves generalization, as shown in Figs. 17(d), 18(d). In *fold-clothes*, the trend is monotonic and especially pronounced under low data budgets, indicating that broader environmental coverage effectively compensates for limited per-scene data, while per-scene data quantity becomes less critical once the overall budget is sufficiently large. In *cup-on-saucer*, the decrease is slightly less smooth but still consistent overall, with stronger gains under higher data usage, suggesting that this task benefits from both sufficient per-scene data and expanded scene diversity.

4) Mixed Diversity Scaling: Under a fixed 4-hour data budget, we study the joint effect of scaling scene diversity (from 4 to 8 scenes) and demonstrator diversity (from 4 to 8 demonstrators), evaluating on unseen demonstrators and scenes collected in other labs. As shown in Figs. 17(c) and 18(c), increasing scene diversity consistently reduces Avg-MSE for both *fold-clothes* and *cup-on-saucer*, indicating that broader environmental coverage is a reliable driver of scene generalization. Demonstrator diversity provides an additional but task-dependent benefit: for *fold-clothes*, using 8 demonstrators consistently outperforms 4, suggesting that increased demonstrator variation yields a richer motion distribution that better covers unseen scenarios. In contrast, for *cup-on-saucer*, the 4-demonstrator setting slightly outperforms 8 demonstrators, implying that for this more constrained task, additional demonstrator-induced behavioral noise may outweigh its coverage benefits, making generalization primarily driven by scene variation.

Summary. We conduct controlled diversity studies on *fold-clothes* and *cup-on-saucer* using a standardized dataset of 16 demonstrators and 16 scenes, systematically varying demonstrator and scene diversity under fixed data budgets and evaluating generalization with the offline Avg-MSE metric. Across four scaling regimes, we find consistent evidence that *increasing diversity improves generalization*, with scene diversity serving as the most reliable driver across tasks. Demonstrator diversity provides additional gains, particularly for *fold-clothes*, where demonstrator variation meaningfully enriches the motion distribution, while its benefit is more task-dependent for *cup-on-saucer*. Overall, these results highlight

TABLE VIII: **Single-scene demonstrator scaling (fixed 2-hour budget).** Training data are collected in one fixed scene. As the number of training demonstrators increases, the per-demonstrator data duration decreases so that the total budget remains 2 hours. Evaluation is performed on a held-out 17th demonstrator with 7.5 mins of data in the same scene. We refer to a demonstrator–scene pair as a DS Pair.

# Train Demonstrators	Mins / DS Pair
1	120.0
2	60.0
4	30.0
8	15.0
16	7.5

TABLE IX: **Multi-scene demonstrator scaling (fixed 8-hour budget).** Training data are collected across 8 fixed scenes. As the number of training demonstrators increases, the per-demonstrator data duration decreases proportionally to maintain the total budget of 8 hours. Evaluation is conducted on unseen demonstrators within the same 8 scenes. We refer to a demonstrator–scene pair as a DS Pair.

# Train Demonstrators	Mins / DS Pair
4	15.0
8	7.5
12	3.75

the complementary roles of scene and demonstrator diversity in shaping generalization for human data, with their relative importance varying by task structure and constraint.

TABLE X: **Mixed diversity scaling (fixed 4-hour budget).** This table details the distribution of the fixed 4-hour data budget across varying numbers of scenes and demonstrators. Scaling both axes allows for the study of joint environmental and motion diversity effects. Evaluation is conducted on unseen demonstrators and scenes. We denote a demonstrator as D, a scene as S, and a demonstrator–scene pair as a DS pair.

# S	# D	Mins / D	Mins / S	Mins / DS Pair
4	4	60.0	60.0	15.00
5	4	60.0	48.0	12.00
6	4	60.0	40.0	10.00
7	4	60.0	34.3	8.57
8	4	60.0	30.0	7.50
4	8	30.0	60.0	7.50
5	8	30.0	48.0	6.00
6	8	30.0	40.0	5.00
7	8	30.0	34.3	4.29
8	8	30.0	30.0	3.75

L. Latent Space Visualization

We choose UMAP over t-SNE for dimensionality reduction because, while both preserve local neighborhood structure, UMAP also retains a degree of global structure.

EgoVerse-I Dataset Visualization For the EgoVerse-I dataset visualization, we load all training data in sequential order and sample uniformly (selecting every 15 datapoints).The data points are then passed into DinoV3 [45] large model which returns image embeddings. These image embeddings then have their dimension reduced to 2 with UMAP, which we visualized in Fig. 5.

Demonstrator Diversity Visualization For the demonstrator diversity visualization, we load all training data in sequential order and also sample uniformly (selecting every 15 datapoints). The datapoints are then passed into the trained HPT model. We then take the 64 action conditioned tokens that condition the Flow Matching Action Decoder and flatten them into a single latent vector. UMAP is then applied on these latent vectors to generate 2 dimensional vectors which we visualized in Fig. 12.

TABLE XI: **Scene diversity scaling data composition.** Total training budget (minutes) for different scene counts and data-usage fractions (relative to 60 min/scene at 100%). We evaluate all models on unseen demonstrators at unseen scenes.

# Scenes	Data usage fraction				
	6.25%	**12.5%**	**25%**	**50%**	**100%**
1	3.75	7.5	15	30	60
2	7.5	15	30	60	120
4	15	30	60	120	240
8	30	60	120	240	480
16	60	120	240	480	960

All values are minutes of recording time for training data.