
EgoBridge: Domain Adaptation for Generalizable Imitation from Egocentric Human Data

Ryan Punamiya¹ Dhruv Patel¹ Patcharapong Aphiwetsa¹ Pranav Kuppili¹
Lawrence Y. Zhu¹ Simar Kareer^{1*} Judy Hoffman^{1*} Danfei Xu^{1*}
¹Georgia Institute of Technology *Equal advising
rpunamiya6@gatech.edu

Abstract

Egocentric human experience data presents a vast resource for scaling up end-to-end imitation learning for robotic manipulation. However, significant domain gaps in visual appearance, sensor modalities, and kinematics between human and robot impede knowledge transfer. This paper presents EgoBridge, a unified co-training framework that explicitly aligns the policy latent spaces between human and robot data using domain adaptation. Through a measure of discrepancy on the joint policy latent features and actions based on Optimal Transport (OT), we learn observation representations that not only align between the human and robot domain but also preserve the action-relevant information critical for policy learning. EgoBridge achieves a significant absolute policy success rate improvement by 44% over human-augmented cross-embodiment baselines in three real-world single-arm and bimanual manipulation tasks. EgoBridge also generalizes to new objects, scenes, and tasks seen *only* in human data, where baselines fail entirely. Videos and additional information can be found at <https://ego-bridge.github.io/>

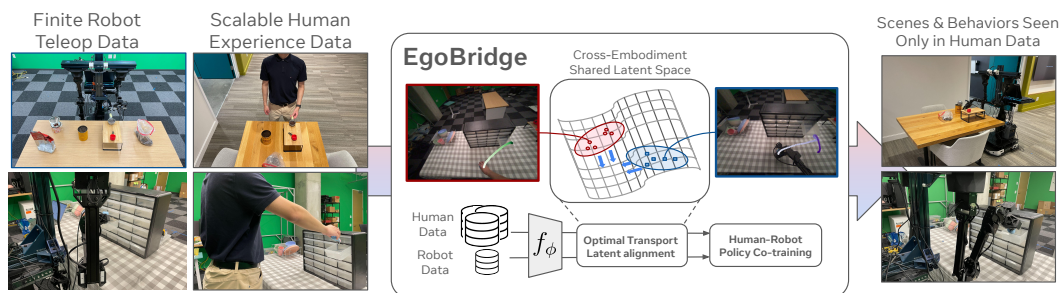


Figure 1: EgoBridge enables rich knowledge transfer from human to robot, based on our key hypothesis: aligned latent representations yield stronger transfer. Our algorithm, which adapts optimal transport, aligns behaviors which are similar across embodiments. This enables EgoBridge to generalize to objects, scenes and even motions demonstrated only in human data.

1 Introduction

Supervised imitation learning methods such as behavior cloning have emerged as a promising path to scaling robot performance across diverse objects, tasks, and environments. However, while large-scale models in vision and language have achieved remarkable generalization through Internet-sourced data, replicating this success in robotics remains challenging due to the labor-intensive nature of collecting teleoperated demonstrations. Deploying physical robots to many new environments to collect data with enough coverage and diversity is economically and practically intractable.

In this work, we aim to enable robots to learn from egocentric recordings of natural human behavior, collected by increasingly ubiquitous wearable devices (e.g., XR devices and smart glasses). Without a robot in the loop, such data is cheap and scalable to collect and captures natural human interactions with the world. More importantly, it reflects the *embodied human experience*, as it contains both observations (e.g., egocentric RGB images) and actions (e.g., hand motions). Unlike unstructured data sources such as Internet videos, the rich embodied information allows us to treat human data and robot data as equal parts in a continuous spectrum of demonstration data and potentially learn from both with a unified learning framework.

However, the multitudes of *domain gaps* between human and robot pose significant challenges in designing such a framework. Human bodies and robots have different visual appearances. Even within a shared action space, kinematic differences can lead to behavior distribution shifts. Robots also have additional sensing modalities such as wrist cameras that are often missing from embodied human data. While recent works such as EgoMimic [1] have attempted to bridge the embodiment gaps with techniques such as visual masking, data normalization, and motion retargeting, such domain gaps still largely remain. More broadly, simply co-training from cross-domain data does not automatically yield effective knowledge transfer, as suggested by recent studies [2]. Such challenges prevent policies from scaling their performance primarily with human data.

We formalize the human-robot cross-embodiment learning problem as a *domain adaptation problem*, where human and robot data represent two labeled distributions with significant *covariate shifts* in observations due to embodiment gaps. Standard domain adaptation approaches often rely on global distribution alignment techniques such as adversarial training [3] and maximum mean discrepancy minimization [4]. However, they primarily address high-level tasks such as image classification and fail to preserve detailed action-relevant information—a critical requirement for robot learning where actions and observations are temporally correlated under compounding covariate shift.

To address these challenges, we propose **EgoBridge**, a novel domain adaptation approach that uses Optimal Transport (OT) to align latent representations from human and robot domains as part of the policy co-training objective. Unlike conventional domain alignment methods, our OT formulation explicitly exploits the inherent relationship between motion similarities in human and robot domains to form *pseudo-pairs* as supervision for the adaptation process. Concretely, we use the dynamic time warping (DTW) distance among human and robot motion trajectories to shape the OT ground cost. This encourages the transport map to find a minimal-cost coupling between human and robot data exhibiting similar behaviors. As such, EgoBridge aligns policy representations across domains via a differentiable OT loss (Sinkhorn distance), while preserving action-relevant information for policy learning. Importantly, we show that EgoBridge learns a shared latent representation that *generalizes* beyond the paired data. This enables the policy to learn behaviors observed *only within the human dataset*, effectively enabling the policy to scale primarily with human data.

We evaluate **EgoBridge** on both a reproducible simulation benchmark task and three challenging real-world manipulation tasks. Our results show that **EgoBridge** consistently improves policy success rates compared to human-augmented cross-embodiment baselines, for up to 44% absolute success rate improvement, and effectively transfers behaviors from diverse human demonstrations to robotic execution in tasks requiring spatial, visual, and task generalization.

2 Related Work

Supervised Imitation Learning. Supervised Imitation Learning (SIL), notably Behavior Cloning, leverages expert demonstrations for policy learning and has achieved significant success in robotics, particularly with large-scale datasets [5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. State-of-the-art Vision-Language-Action (VLA) models [10, 8, 12, 11] integrate Vision-Language Models (VLMs) with action decoders, enhancing generalization by incorporating semantic understanding from internet-scale data. Despite these advances, even top-performing models like Pi 0.5 [11] require extensive labeled robot manipulation data for robust physical world interaction and broad capabilities. Given the intractability of scaling robot teleoperation data, alternative sources such as more accessible human data are being explored. Consequently, our work leverages scalable human demonstrations alongside in-domain robot data to improve learning outcomes.

Learning from Human Data. Human data presents two main opportunities for robot learning: abundant unlabeled online videos and curated, labeled demonstrations [15, 16, 1]. Unlabeled web

videos, though plentiful, require pseudo-labeling of actions via inverse dynamics models [17] or point tracking [18, 19, 20] for policy training, forming a basis for some foundation models [12], yet often still necessitating in-domain robot data. Alternatively, labeled human demonstrations can be co-trained with robot data as distinct embodiments [1, 21, 22], enhancing robustness and scene understanding. However, generalizing to novel behaviors observed only in human data remains challenging. To address this limitation, we propose a novel learning framework for jointly aligning observation-action spaces across human and robot embodiments to improve generalization.

Domain Adaptation and Optimal Transport. Domain Adaptation (DA) aims to reduce reliance on target-specific data by leveraging labeled source domain data to bridge distribution gaps and improve performance on unlabeled target domains. In cross-embodiment learning, DA has been applied for shared dynamics modeling [23], unsupervised reward modeling [24], and high-level planning [25]. However, many DA methods primarily focus on global distribution alignment, which can neglect fine-grained action information crucial for transfer across robot embodiments. To address this, computer vision research introduced Optimal Transport (OT) as a loss function for DA to align both local and global distributions [26, 27, 28]. Building on these insights, we propose an action-aware DA approach using OT to learn shared representations across embodiments, thereby improving observation and behavior generalization.

3 Preliminaries and Problem Statement

3.1 Optimal Transport

Optimal Transport for Domain Adaptation. Optimal Transport (OT) offers a principled framework for comparing probability distributions by considering the geometry of their sample spaces. Given two distributions, μ_S (source) and μ_T (target), over a common metric space $\mathcal{X}$, and a cost function $\mathcal{C}(x^S, x^T)$ measuring the effort to move mass from $x^S \in \mathcal{X}$ to $x^T \in \mathcal{X}$, OT finds a probabilistic coupling $\gamma \in \mathcal{P}(\mathcal{X} \times \mathcal{X})$ that minimizes the expected transport cost:

$$\gamma^* = \arg \min_{\gamma \in \Pi(\mu_S, \mu_T)} \mathbb{E}_{(x^S, x^T) \sim \gamma} [\mathcal{C}(x^S, x^T)],$$

where $\Pi(\mu_S, \mu_T)$ is the set of all joint distributions whose marginals are μ_S and μ_T . For discrete empirical distributions from N_S source samples $\{x_i^S\}$ and N_T target samples $\{x_j^T\}$, the cost matrix is $C_{ij} = \mathcal{C}(x_i^S, x_j^T)$, and the total cost is $\langle \gamma, C \rangle_F = \sum_{i,j} \gamma_{ij} C_{ij}$.

Differentiable Optimal Transport as a Loss Function. When used as a cost function to align representations, the standard OT problem is often regularized. The Sinkhorn algorithm [29] introduces an entropic regularization term to the OT objective, yielding a differentiable approximation T_ϵ^* to the optimal transport plan:

$$T_\epsilon^* = \arg \min_{T \in \Pi(\mu_S, \mu_T)} \mathbb{E}_{(x^S, x^T) \sim T} [\mathcal{C}(x^S, x^T)] - \epsilon H(T),$$

where $\epsilon > 0$ is the regularization strength and $H(T)$ is the entropy of the coupling. This regularization makes the problem strictly convex and efficiently solvable. The resulting regularized optimal transport cost, $\sum_{i,j} (T_\epsilon^*)_{ij} C_{ij}$, is differentiable with respect to the cost matrix C . This allows OT to serve as a loss function within deep learning frameworks, enabling the learning of feature encoders that map inputs to a space $\mathcal{X}$ where their distributions are aligned by minimizing this transport cost.

3.2 Human and Robot Data Sources

We consider egocentric human data ($\mathcal{D}_H$) and teleoperated robot data ($\mathcal{D}_R$). $\mathcal{D}_H = \{(o_t^H, a_t^H)\}_{t=1}^{N_H}$ consists of N_H egocentric human demonstrations, where $o_t^H \in \mathcal{O}^H$ are observations from wearable sensors (e.g., head-mounted cameras) and $a_t^H \in \mathcal{A}$ are human actions in a common action space (e.g., robot end-effector and human hand poses). This data is abundant and captures natural, diverse behaviors. Conversely, $\mathcal{D}_R = \{(o_t^R, a_t^R)\}_{t=1}^{N_R}$ comprises N_R robot experiences, typically from teleoperation, with $o_t^R \in \mathcal{O}^R$ being robot sensor observations (e.g., ego-centric/wrist cameras, joint states) and $a_t^R \in \mathcal{A}$ the robot actions. This data is often scarce. We describe how each data source is captured and processed in more detail in Sec. 4.3. We assume actions are in trajectory chunks, which is shown to improve prediction temporal consistency of the trained policies [5, 6].

3.3 Cross-Embodiment Imitation Learning: Challenges and Objectives

Our primary goal is to effectively learn from both limited robot demonstrations (D_R) and more abundant, diverse egocentric human demonstrations (D_H). We train a feature encoder $f_\phi : \mathcal{O}^H \cup \mathcal{O}^R \rightarrow \mathcal{Z}$ to project observations from both human ($\mathcal{O}^H$) and robot ($\mathcal{O}^R$) into the shared latent space $\mathcal{Z}$. We jointly train a policy π_θ that maps these learned latent representations $z \in \mathcal{Z}$ to actions $a \in \mathcal{A}$.

Cross-Embodiment Co-Training. A popular approach [1, 21] involves training the policy end-to-end using a standard Behavior Cloning (BC) loss on the aggregated dataset:

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

To effectively learn from both data sources, a critical assumption is that a shared latent space $\mathcal{Z}$ would naturally emerge where the mapping from latent states to actions is domain-invariant, resulting in $P_R(a|f_\phi(o_R)) \approx P_H(a|f_\phi(o_H))$ for observations o_R and o_H from aligned underlying states.

Challenge: Observation Covariate Shift. However, we argue that, without explicit mitigation, the induced marginal distributions over these latents, $\mu_H = P(f_\phi(\mathcal{O}^H))$ and $\mu_R = P(f_\phi(\mathcal{O}^R))$, will exhibit a significant *covariate shift* ($\mu_H \neq \mu_R$). This shift arises from inherent domain gaps in observations (e.g., differing visual appearances, viewpoints, sensor modalities like robot wrist cameras absent in human setups) and embodiment kinematics. We also empirically show that such co-trained representations often form disjoint latent clusters (Sec. 5.3). This covariate shift in the marginal latent distributions undermines the foundational assumption of consistent conditional action distributions across domains, thereby limiting effective knowledge transfer from human to robot.

Goal: Generalizable Cross-Embodiment Transfer. This motivates our method that aims at *joint domain adaptation* (Sec. 4.1), where we explicitly seek to align the latent representations from human and robot data while preserving action-relevant information. Successfully addressing this latent misalignment should enable two crucial levels of generalization: First, for tasks present in both $\mathcal{D}_H$ and $\mathcal{D}_R$, the system must achieve **observation generalization**. This involves effectively bridging visual and sensor gaps, which include appearance changes that do not affect behaviour. Second, and more ambitiously, the system should enable **behavior generalization** (*Beh. Gen.*) allowing the robot to perform tasks or handle novel situations (e.g. task variations) observed *only* in $\mathcal{D}_H$. This requires the learned encoder f_ϕ to generalize beyond scenarios with paired human and robot data which require motion information to be transferred, such as spatial variations in goal pose.

4 EgoBridge

EgoBridge is a co-training framework designed to effectively imitate embodied human demonstrations and robot demonstrations. It explicitly addresses the domain gap between human and robot experiences through an Optimal Transport (OT)-based domain adaptation mechanism integrated into the policy learning process. The core of EgoBridge lies in aligning the *joint distributions of latent policy features and corresponding actions* across the human and robot domains. The following sections detail this joint domain adaptation formulation (Sec. 4.1), the design of its OT cost function (Sec. 4.2), and the overall training process and system details (Sec. 4.3).

4.1 Joint Domain Adaptation via Optimal Transport

To address the latent covariate shift (Sec. 3.3) and generalizable cross-embodiment transfer, EgoBridge builds on Optimal Transport (OT, Sec. 3.1) to directly shape the shared feature encoder f_ϕ . Unlike standard domain adaptation techniques [30] that often aligns only the marginals $P(f_\phi(O))$, which can discard action-relevant information, EgoBridge optimizes f_ϕ to align the joint distributions of its output latent features and their corresponding actions, i.e., $P(f_\phi(O), A)$.

Given mini-batches of human data $\{(o_i^H, a_i^H)\}_{i=1}^{N_H}$ and robot data $\{(o_j^R, a_j^R)\}_{j=1}^{N_R}$, where a represents a temporally-extended action trajectory, we define an OT-based loss to guide the learning of f_ϕ . The differentiable Sinkhorn OT formulation [29] allows us to compute a loss based on the alignment of the empirical distributions of $(f_\phi(o^H), a^H)$ and $(f_\phi(o^R), a^R)$:

$$\mathcal{L}_{\text{OT-joint}}(\phi) = \sum_{i,j} (T_\epsilon^*)_{ij} \cdot \mathcal{C}((f_\phi(o_i^H), a_i^H), (f_\phi(o_j^R), a_j^R)).$$

Here, $(T_\epsilon^*)_{ij}$ is the optimal transport plan coupling the i -th human and robot (latent, action) pairs. The cost function $\mathcal{C}(\cdot, \cdot)$ measures the dissimilarity between these joint entities. Its design is crucial for capturing meaningful behavioral similarities across domains, which we detail in Sec. 4.2.

Minimizing $\mathcal{L}_{\text{OT-joint}}(\phi)$ directly influences the parameters ϕ of the encoder f_ϕ . The gradients from this loss encourage f_ϕ to produce latent features $f_\phi(o_i^H)$ and $f_\phi(o_j^R)$ that minimize the transport cost required to align them, especially when their associated actions a_i^H and a_j^R are behaviorally similar (as determined by $\mathcal{C}$). At each step a transport plan is computed which influences the feature encoder to couple the action pairs. This iterative process shapes the latent space $\mathcal{Z}$ to be domain-invariant with respect to the joint observation-action manifold.

4.2 Designing OT Cost Function for Action-Aware Joint Adaptation

Our joint OT formulation (Section 4.1) relies on a cost function $\mathcal{C}((z^H, a^H), (z^R, a^R))$ to measure the dissimilarity between joint human and robot latent feature-action pairs. A critical challenge is designing this cost to be robust to inherent domain differences. Specifically, we aim to account for *temporal misalignments*, where human and robot often execute the same task at of different speed, e.g., humans might be 2-3 times faster than teleoperated robots, and *kinematic variations*, where even within a shared SE(3) end-effector action space and hand-eye alignment through an egocentric coordinate frame (Sec. 4.3), minor kinematic differences exist.

Dynamic Time Warping. To identify behaviorally similar action sequences while accounting for these differences, we propose to leverage Dynamic Time Warping (DTW) to guide the OT alignment. DTW [31] has been effective in prior work to compare time series data and trajectories. Formally, given two action sequences $\mathbf{a}^H = (a_1^H, \dots, a_T^H)$ from human data and $\mathbf{a}^R = (a_1^R, \dots, a_T^R)$ from robot data of identical length T , DTW finds an alignment path $\pi \subseteq \{1, \dots, T\} \times \{1, \dots, T\}$ that minimizes the cumulative distance:

$$\text{DTW}(\mathbf{a}^H, \mathbf{a}^R) = \min_{\pi \in \mathcal{A}(T)} \sum_{(i,j) \in \pi} \|a_i^H - a_j^R\|^2$$

where $\mathcal{A}(T)$ is the set of admissible monotonic alignments constrained to start at $(1, 1)$ and end at (T, T) , while allowing small local shifts to account for temporal variations.

Soft Supervision. With DTW, we can identify highly correlated samples from both domains. However, directly utilizing the DTW cost is noisy and instead is a much stronger measure of relative pairing between human and robot samples. As such the DTW cost can be used to *pseudo-pair* D_H and D_R . On a mini-batch of size B of sampled ground-truth state-action pairs from D_S and D_T , we form a DTW cost matrix $A \in \mathbb{R}^{B \times B}$. Here, $A_{i,j} = \text{DTW}(a_i^S, a_j^T)$. The row-wise minimum cost gives us the most behaviorally similar human pseudo-pair for each robot sample: $i^*(j) = \arg \min_i A_{i,j}$.

Given the standard OT Euclidean distance cost $D_{ij} = \|f_\phi(o_i^H) - f_\phi(o_j^R)\|^2$, we define the joint cost $\tilde{\mathcal{C}}((f_\phi(o_i^H), a_i^H), (f_\phi(o_j^R), a_j^R))$ for L_{OT} as:

$$\tilde{C}_{ij} = \begin{cases} D_{ij} \cdot \lambda & \text{if } i = i^*(j) \\ D_{ij} & \text{otherwise} \end{cases}$$

where $0 < \lambda \ll 1$ is a small scalar. This cost function strongly incentivizes OT to match robot samples with their behaviorally closest human pseudo-pair (identified by DTW) by significantly reducing the cost for these pairs. For all other pairs, the cost is simply the distance in the latent feature space. This soft supervision from DTW guides the latent space alignment towards behaviorally relevant correspondences across embodiments.

4.3 Putting it all together: EgoBridge

With all the ingredients for joint distribution adaptation using OT with joint policy co-training, we present EgoBridge as a unified cross-embodiment imitation learning algorithm and describe its corresponding robot learning system and policy architecture.

Policy Co-Training with Joint Adaptation. EgoBridge jointly optimizes the feature encoder f_ϕ and π_θ , with the joint OT loss applied on the feature encoder and the BC co-training loss applied end-to-end through both components: $\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{BC-cotrain}}(\phi, \theta) + \alpha \mathcal{L}_{\text{OT-joint}}$, with tunable weight α . Detail of the algorithm and hyperparameter choices are described in Appendix.

Egocentric Human Data. We largely follow EgoMimic [1] and leverage a wearable smart glass platform Meta Project Aria [32] as our main data collection platform. The platform allows us to collect *exteroceptive*, egocentric first person POV RGB images (I_{ego}^H), and *proprioceptive* data (q^H), cartesian pose for both arms $\in SE(3) \times SE(3)$. We take inspiration from EgoMimic to construct stable reference frames to form action sequences of cartesian pose (a^H) in the egocentric camera frame.

Teleoperated Robot Data. We base our robot platform on the open-source Eve robot [1]. In particular, we leverage Aria glasses as the main egocentric perception sensor for the robot and mount it in a way that emulates the hand-eye configuration of a human adult (I_{ego}^R). This effectively mitigates the human-robot camera device gap, allowing us to specifically study the appearance, kinematic and behaviour gaps. The robot additionally provides RGB streams from its two RealSense D405 wrist cameras, I_{wrist}^R . The actions consist of a sequence of corresponding future end-effector poses, $a^R \in SE(3) \times SE(3)$.

Shared Policy Architecture. Inspired by recent cross-embodiment policy learning [33] and DETR-style architectures [34], our policy employs a shared transformer encoder “trunk” (f_ϕ) and a shared transformer policy decoder “head” (π_θ) (Fig. 2). We perform embodiment-specific gaussian normalization to the proprioception and actions. The encoder f_ϕ begins with *stems*—shallow networks that tokenize raw observations; notably, a *shared* vision stem processes main egocentric RGB images (I_{ego}) from both human and robot to enforce visual alignment, while separate stems handle robot wrist camera inputs (I_{wrist}). The subsequent multi-layer encoder trunk processes these concatenated tokens, along with M prepended learnable context tokens upon which the OT loss is applied. The multi-block decoder head then generates actions by attending to this encoded context, utilizing T learnable action tokens and injecting context through alternating self and cross-attention blocks.

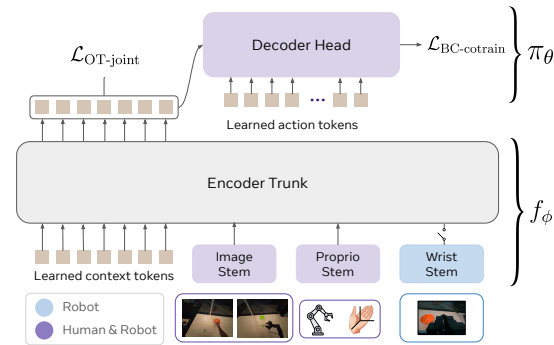


Figure 2: EgoBridge policy co-training with joint adaptation. The encoder f_ϕ consists of modality-specific input stems and the encoder trunk, while the policy π_θ consists of a shared multi-block transformer decoder. $\mathcal{L}_{OT-joint}$ optimizes the encoder while $\mathcal{L}_{BC-cotrain}$ optimizes the entire network.

5 Experiments

In this section, we aim to validate three core hypotheses. **H1:** EgoBridge improves co-training performance for scenarios present in both human and robot data. **H2:** EgoBridge enables generalization to scenarios only seen in human data. **H3:** EgoBridge learns a shared latent space where human and robot data are aligned in task-relevant manners. We validate the hypotheses through a standard simulation benchmark task (Sec. 5.1) and three complex real-world manipulation tasks (Sec. 5.2).

5.1 Simulation Evaluation

To facilitate reproducible study and eliminate the confounding factors in real robot systems, we study a well-explored planar pushing task [6], where the goal is to push a T-shaped object to a desired goal location. We emulate “human” (source) data through a blue circle pusher and “robot” (target) data through a salmon triangle pusher (Fig. 3), with lower floor friction. The differences aim to analogize the appearance and agent-environment dynamics gaps between human and robot data.

Source and Target Domain Data. In our “robot” target domain, we collect demos in the standard push-T setting, but in our “human” source domain, we alter the background color to purple and change the T configuration to be mirrored, requiring a new motion to slot into place 3. The change in background color is analogous to the human data containing new visual scenery, and the change in starting configuration is analogous to the human demonstrating new motions in their demonstrations.

Training and Evaluation. To eliminate factor from model design (Sec. 4.3), we choose a standard ResNet-UNet Diffusion Policy [6] and apply the OT-joint loss on the feature outputs of the ResNet encoder. We perform standard action normalization and co-train the policy on both the triangle and

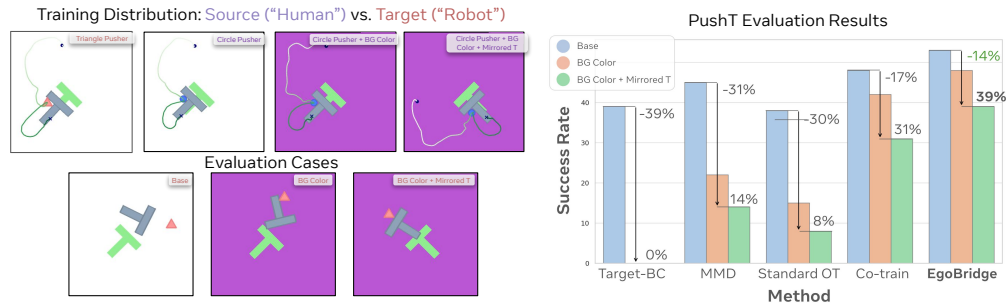


Figure 3: In the simulated Push-T experiments, we probe a toy version of visual and motion level generalization from human to robot. We have narrow target "robot" data represented by the triangle pusher on a white background, and diverse source "human" data represented by the circle pusher with changes in background color and T configuration. We test our "robot" on the diverse human scenarios, and find that EgoBridge outperforms traditional Domain Adaptation baselines.

Table 1: Real World Evaluation Results: In-Distribution and Generalization

Method	Scoop Coffee (SR)			Drawer (SR)			Laundry (Pts SR)
	In-Dist.	Obj. Gen.	Scene+Obj Gen.	Total (Pts SR)	Place Toy	Beh. Gen.	
Robot-only BC	33%	40%	7%	38 9%	28%	0%	38 28%
Co-train	53%	46%	0%	55 22%	42%	0%	41 33%
EgoMimic [1]	60%	53%	0%	49 14%	39%	0%	38 33%
MimicPlay [25]	33%	27%	0%	33 14%	22%	0%	32 28%
ATM [19]	47%	33%	0%	56 6%	17%	8%	35 28%
EgoBridge	67%	60%	27%	77 47%	72%	33%	48 72%

circle pusher data. We evaluate the policy on 3 cases: Triangle in the standard setting, Triangle with purple background, Triangle with purple background and flipped T. We evaluate a total of 100 fixed seeds across all the models and report the mean reward (max IoU with goal) and the success rate (reward ≥ 0.9).

Baselines. In the more controlled simulation settings, we choose to compare EgoBridge against conventional domain adaptation baselines. We choose **Maximum Mean Discrepancy (MMD)** [30] as an alternative domain adaptation loss to joint-OT on the feature encoder. We also test **Standard OT**, which performs marginal alignment instead of joint alignment. The **Co-train** baseline trains on evenly-sampled data from both domains without an alignment loss. Finally, **Target-only** is a control study which trains the policy only on the target (triangle) data.

5.2 Real World Evaluation

We evaluate EgoBridge on three challenging real-world manipulation tasks, as illustrated in Fig. 4.

Drawer: The robot interacts with a 6x4 drawer array, tasked to pick a toy, place it into a pre-opened drawer, and close it. Robot data (144 demonstrations) covers three of the four array quadrants, each quadrant being a 3x2 arrangement. Human data (1 hour) covers all four quadrants, providing demonstrations for motions into the fourth, robot-unseen quadrant. This setup specifically tests *behavior generalization* to drawer locations only seen in human data. Points (**Pts**) are awarded for successful completion of each stage and a trial is considered a success only if the robot completes all actions. Evaluation uses 48 trials (2 rollouts for each of the 24 drawers).

Scoop Coffee: The robot uses its left arm to scoop coffee beans with a spoon and empty them into a target. Robot data (50 demonstrations) involves a specific target (can) in one scene. Human data (2 hours) includes demonstrations with both the can and a new target (grinder), across two distinct scenes, one of which is novel to the robot. Target object positions are randomized (30x23 cm area). We evaluated *observation generalization* for: (1) the new grinder target, and (2) the new scene with the new target, that is, scooping to the grinder in the new scene, seen in human data only. Performance is measured by success rate over 15 rollouts across 5 distinct target locations.

Laundry: This is a bimanual task where the robot needs to fold the shirt in 50 x 22 cm range with a rotation range ± 30 degrees. The robot uses both arms to fold the right sleeve, the left sleeve, and

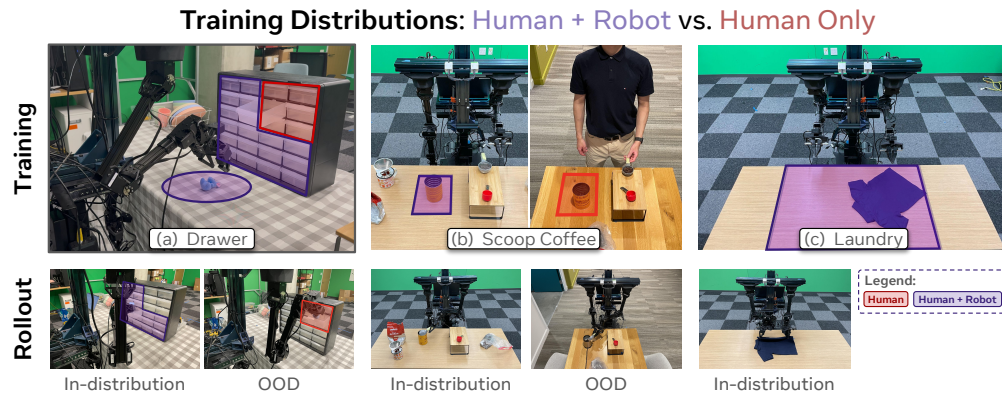


Figure 4: **Training Data and Evaluation Settings.** We show the distribution of human and robot training data (top) and evaluation setting, where in-distribution scenarios are in both human and robot data, while out-of-distribution (OOD) scenarios is seen only in human data.

then the final stage to fold the shirt in half. We award **Pts** for each successful stage and consider it success if all the individual stages are successful. We collect 2 hours of robot data which include 300 demonstrations across 3 shirts, and 2 hours of human data comprising approximately 700 demonstrations. We conduct 18 evaluations with diverse shirt initial placement and colors.

Baselines. We adopt the following baselines for real-world tasks. *Co-train*: Direct co-training of the robot and human data using BC loss, without any latent alignment. *EgoMimic*: [1] Co-training with explicit vision and action-space alignment using masking, shared end-effector pose head (human and robot) and a separate joint-space head for robot. *Mimicplay*: [25] A hierarchical policy with a latent high-level planner co-trained on human and robot data, and an action decoder fine-tuned on robot data. *Any-Point Trajectory Modeling (ATM)*: [19] A hierarchical policy where the high-level planner is initially co-trained on 2D point tracks derived from both robot and human video data. These point tracks are obtained via Co-tracker [35]. Following this, high-level planner is frozen and an action decoder is fine-tuned specifically on robot data. *Target-only BC*: Trained only on robot data.

5.3 Results

EgoBridge improves in-domain task performance (H1). Over our set of in-domain tasks, we observed an improvement of 7-44% in absolute task success rate over both human-augmented imitation learning baselines and robot-only behaviour cloning policies. While methods like naive co-training, EgoMimic and ATM also improve in-domain performance across all tasks, EgoBridge consistently outperforms them. We hypothesize that the strongly aligned latent representations facilitate better cross-embodiment transfer.

EgoBridge enables generalization to objects and scenes only seen in human data (H2). While it is difficult to collect robot data across diverse scenes and objects, it is trivial to do for human data, so it's critical that we can transfer this knowledge from human to robot, inspiring our experimental setup. Specifically, in the *Scoop Coffee* task, our human data introduces a new coffee grinder, table, lighting and height variations, completely unseen to the robot. We find EgoBridge outperforms all baselines when tested on the new coffee grinder (7-33%). Further, most methods fail entirely when tested on the new grinder + scene, but EgoBridge retains a performance of 27%. We observe similar robustness trends in our simulated benchmark Fig. 3, where EgoBridge enables generalization in the push-T task to a new background and starting configuration, outperforming all baselines.

EgoBridge enables generalization to new behaviors only seen in human data (H2). In our most challenging setting, we seek to show that we can learn *entirely new* motions from human data alone. In the drawer task, the robot data covers 3/4 drawer quadrants, whereas the human data covers all 4 quadrants. We evaluate our policy's performance on these new drawers, and find that EgoBridge is able to generalize to these locations with a success rate of 33%, whereas most methods fail entirely (Tab. 1). While all the methods were exposed to the same human data, only EgoBridge was able to effectively transfer the human motion to a novel robot action. We attribute this success to the well aligned latent representations, which enables human to robot knowledge interpolation. We also

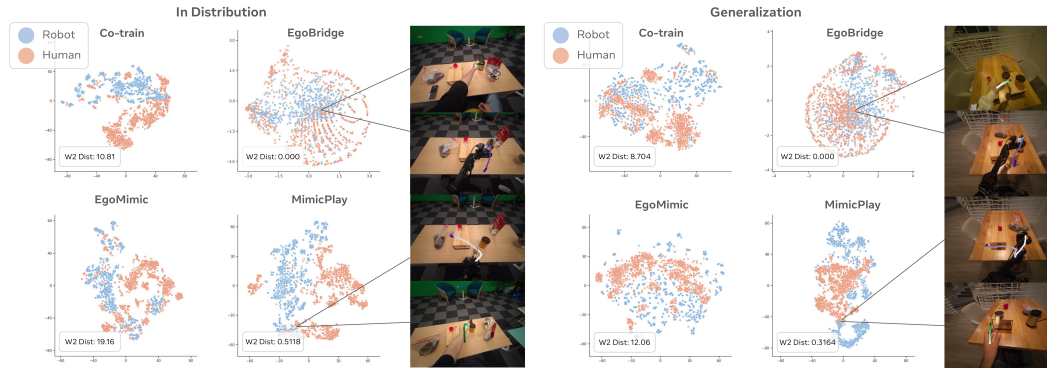


Figure 5: Visualization of TSNE plots on encoded features for EgoBridge and baselines, with the mean Wasserstein-2 distance and KNN pairs of aligned human-robot data visualized.

observed a similar trend in the simulated benchmark where EgoBridge faces the lowest performance drop of 14% compared to all baselines in the *mirrored-T + background colour* reflected in Fig. 3.

EgoBridge learns a shared latent space that aligns human and robot data in a task relevant manner (H3). We hypothesize that an ideal latent space for transfer would jointly embed human and robot data into a space with high overlap and semantic interoperability. To probe this, we create a TSNE visualization of the action tokens from our transformer backbone. EgoBridge not only exhibits the highest latent overlap between human and robot as measured by Wasserstein distance as seen in Fig. 5, but also upon inspecting K-nearest neighbor pairs in latent space, exhibits the most semantically similar neighbors. For instance, we see the human and robot performing the same phase of a given task, whereas in baselines like MimicPlay that aligns marginals with KL-div, the semantic similarity is lacking. This result is highly correlated with the task success rates for in-distribution and generalization where baselines with poor alignment perform lower consistently across all evaluations.

Ablation. We ablate three key components of our method 1) replacing our DTW-based pairing metric to instead use simple MSE, 2) replacing the joint OT objective $\mathcal{L}_{OT-joint}$ with standard marginal alignment, and 3) removing any auxiliary alignment objectives (direct co-training). We find that replacing the cost function with MSE leads to the largest performance drop for in-distribution policy success rate from 47% to 17%, seen in Tab. 2, which emphasizes the importance of creating semantically similar pseudo-pairs. Ablating Joint-OT also shows a large performance drop in both in-distribution and generalization cases which emphasizes how naive marginal alignment cannot transfer knowledge from human data effectively. Ablating an auxiliary alignment loss also shows a significant performance drop for in-distribution success rate and leads to failure in generalization cases, emphasizing the need for joint distribution alignment.

Table 2: Ablation Results (Drawer)

Method	Drawer (SR)	Beh. Gen. (SR)
EgoBridge	47%	33%
MSE	14%	17%
Standard-OT	33%	17%
Co-train	22%	0%

6 Conclusion and Limitations

We presented EgoBridge, a novel co-training framework designed to enable robots to learn effectively from egocentric human data by explicitly addressing domain gaps. By leveraging Optimal Transport on joint policy latent feature-action distributions, guided by Dynamic Time Warping cost on action trajectories, EgoBridge successfully aligns human and robot representations while preserving critical action-relevant information. Our experiments demonstrated significant improvements in real-world task success rates (up to 44% absolute gain) and, importantly, showed robust generalization to novel objects, scenes, and even tasks observed only in human demonstrations, where baselines often failed.

While EgoBridge demonstrates promising single-task transfer, the DTW-based action alignment cost may not be as informative in multi-task joint domain adaptation settings. In future works, we seek to adopt more generalizable alignment cost such as natural language embedding distances [36] obtained from Vision-Language models and visual features extracted from foundation models. Other avenues for future work include extending the joint distribution adaptation to multiple embodiments and leveraging Internet-sourced human data without action labels.

7 Acknowledgment

This work was supported in part by a research gift from Meta Platforms, Inc, the Defense Advanced Research Projects Agency (DARPA) under the TIAMAT program and by the National Science Foundation under Grant No. 2144194. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the supporting organizations.

References

- [1] 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.
- [2] Adam Wei, Abhinav Agarwal, Boyuan Chen, Rohan Bosworth, Nicholas Pfaff, and Russ Tedrake. Empirical analysis of sim-and-real cotraining of diffusion policies for planar pushing from pixels. *arXiv preprint arXiv:2503.22634*, 2025.
- [3] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation, 2017.
- [4] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks, 2017.
- [5] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.
- [6] 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.
- [7] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.
- [8] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishk Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huang Tran, Vincent Vanhoucke, Quan Vuong, Ayzan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.
- [9] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
- [10] 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.
- [11] 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.

- [12] 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.
- [13] 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.
- [14] 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.
- [15] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild, 2022.
- [16] Zipeng Fu, Qingqing Zhao, Qi Wu, Gordon Wetzstein, and Chelsea Finn. Humanplus: Humanoid shadowing and imitation from humans, 2024.
- [17] 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.
- [18] 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.
- [19] Chuan Wen, Xingyu Lin, John So, Kai Chen, Qi Dou, Yang Gao, and Pieter Abbeel. Any-point trajectory modeling for policy learning, 2023.
- [20] Juntao Ren, Priya Sundareshan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning, 2025.
- [21] 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.
- [22] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos, 2025.
- [23] Sean J. Wang and Aaron M. Johnson. Domain adaptation using system invariant dynamics models. In Ali Jadbabaie, John Lygeros, George J. Pappas, Pablo A. Parrilo, Benjamin Recht, Claire J. Tomlin, and Melanie N. Zeilinger, editors, *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pages 1130–1141. PMLR, 07 – 08 June 2021.
- [24] Laura Smith, Nikita Dhawan, Marvin Zhang, Pieter Abbeel, and Sergey Levine. Avid: Learning multi-stage tasks via pixel-level translation of human videos, 2020.
- [25] Chen Wang, Linxi Fan, Jiankai Sun, Ruohan Zhang, Li Fei-Fei, Danfei Xu, Yuke Zhu, and Anima Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. *arXiv preprint arXiv:2302.12422*, 2023.
- [26] Nicolas Courty, Rémi Flamary, Amaury Habrard, and Alain Rakotomamonjy. Joint distribution optimal transportation for domain adaptation, 2017.
- [27] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation, 2016.

- [28] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation, 2018.
- [29] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transportation distances, 2013.
- [30] Arthur Gretton, Karsten Borgwardt, Malte J. Rasch, Bernhard Scholkopf, and Alexander J. Smola. A kernel method for the two-sample problem, 2008.
- [31] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1): 43–49, 1978.
- [32] 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 Eickenhoff, 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.
- [33] Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers, 2024.
- [34] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020.
- [35] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together, 2024.
- [36] Albert Yu, Adeline Foote, Raymond Mooney, and Roberto Martín-Martín. Natural language can help bridge the sim2real gap. *arXiv preprint arXiv:2405.10020*, 2024.
- [37] Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trounev, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2681–2690, 2019.
- [38] 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.

A Algorithm Pseudocode

Overview. Algorithm 1 describes the joint policy co-training of the human and robot data with the joint OT loss.

Mini-batch sampling. At each iteration we draw two size- B mini-batches, one from the human dataset $\mathcal{D}_H$ and one from the robot dataset $\mathcal{D}_R$ (Lines 3–4).

Shared encoder. Both batches are fed through a *shared* encoder f_ϕ that maps raw observations o to a latent embedding $z \in \mathbb{R}^d$ (Line 6).

Algorithm 1 EgoBridge Co-Training for Human $\leftrightarrow$ Robot Imitation

Require: Human demos $\mathcal{D}_H$, robot demos $\mathcal{D}_R$, OT loss weight α , DTW discount λ , Sinkhorn regularization ε , LR schedule η

Ensure: Encoder f_ϕ and policy decoder π_θ

- 1: Initialize network parameters ϕ, θ
- 2: **while** training not converged **do**
- 3: $\mathcal{B}_H = \{o_H^{(i)}, a_H^{(i)}\}_{i=0}^B \leftarrow \text{SAMPLEBATCH}(\mathcal{D}_H, B)$ ▷ sample human data
- 4: $\mathcal{B}_R = \{o_R^{(i)}, a_R^{(i)}\}_{i=0}^B \leftarrow \text{SAMPLEBATCH}(\mathcal{D}_R, B)$ ▷ sample robot data
- 5: $z_H \leftarrow f_\phi(o_H), z_R \leftarrow f_\phi(o_R)$ ▷ encode raw observation batch into shared latent space
- 6: **for all** $i \in \{0, \dots, |B_H|-1\}, j \in \{0, \dots, |B_R|-1\}$ **do**
- 7: $A_{ij} \leftarrow \text{DTW}(a_H^{(i)}, a_R^{(j)})$ ▷ DTW distance between actions
- 8: **end for**
- 9: $i^*(j) \leftarrow \arg \min_i A_{ij} \forall j \in \{0, \dots, |B_R|-1\}$ ▷ pick the most similar human traj. for each robot traj.
- 10: **for all** $i \in \{0, \dots, |B_H|-1\}, j \in \{0, \dots, |B_R|-1\}$ **do**
- 11: $D_{ij} \leftarrow \|z_H^{(i)} - z_R^{(j)}\|_2^2$ ▷ observation feature cost
- 12: **if** $i = i^*(j)$ **then**
- 13: $\tilde{C}_{ij} \leftarrow \lambda D_{ij}$ ▷ discount cost if DTW says the pair matches
- 14: **else**
- 15: $\tilde{C}_{ij} \leftarrow D_{ij}$
- 16: **end if**
- 17: **end for**
- 18: $\mu_H \leftarrow \frac{1}{|B_H|} \mathbf{1}, \mu_R \leftarrow \frac{1}{|B_R|} \mathbf{1}$ ▷ uniform probability mass on the minibatch supports
- 19: $T^* \leftarrow \text{SINKHORN}(\mu_H, \mu_R, \tilde{C}, \varepsilon)$ ▷ compute differentiable OT loss with entropic reg.
- 20: $\mathcal{L}_{\text{OT-joint}} \leftarrow \sum_{i,j} T_{ij}^* \tilde{C}_{ij}$
- 21: $\hat{a}_H \leftarrow \pi_\theta(z_H), \hat{a}_R \leftarrow \pi_\theta(z_R)$ ▷ predict actions for both domains
- 22: $\mathcal{L}_{\text{BC-cotrain}} \leftarrow \mathcal{L}_{\text{BC}}(\hat{a}_H, a_H) + \mathcal{L}_{\text{BC}}(\hat{a}_R, a_R)$ ▷ behaviour-cloning loss across domains
- 23: $\mathcal{L} \leftarrow \mathcal{L}_{\text{BC-cotrain}} + \alpha \mathcal{L}_{\text{OT-joint}}$
- 24: $(\phi, \theta) \leftarrow \text{OPTIMIZERSTEP}(\phi, \theta, \nabla_{\phi, \theta} \mathcal{L}, \eta)$
- 25: **end while**
- 26: **return** f_ϕ, π_θ

Behavioural pairing via DTW. The Dynamic Time Warping (DTW) cost is applied on the *action* sequences $a_H^{(i)}$ and $a_R^{(j)}$ from both domains. The assignment $i^*(j)$ (Line 11) identifies a single pseudo-pair in $\mathcal{D}_H$ for each robot trajectory j which minimizes the DTW cost.

Sinkhorn. The Sinkhorn-Knopps algorithm is used to compute an entropy-regularized Optimal Transport plan from the uniform probability mass on the minibatch supports.

Joint-OT. Latent distances $D_{ij} = \|z_H^{(i)} - z_R^{(j)}\|_2^2$ are discounted by $\lambda < 1$ when (i, j) is behaviourally matched (Lines 12–17), producing a shaped cost $\tilde{C}$ that biases the OT loss computed from the Sinkhorn OT transport plan.

Behaviour cloning and objective. The decoder π_θ produces domain-specific actions and is trained with a supervised loss $\mathcal{L}_{\text{BC-cotrain}}$ (Lines 22–24). The overall objective (Line 26)

$$\mathcal{L} = \mathcal{L}_{\text{BC-cotrain}} + \alpha \mathcal{L}_{\text{OT-joint}}$$

balances imitation fidelity with latent-space alignment, with an α balancing factor between the two losses.

B Network Architecture

The following section describes how f_ϕ and π_θ are parameterized in the simulation and real-world experiments. All hyperparameters are summarized in Table 3, Table 4 for real-world and in Table 5 for simulation.

B.1 Real World Experiments

Table 3: Hyperparameters for Real-World Experiments

Symbol	Value
$H \times W \times C$	$480 \times 640 \times 3$
d_{proj}	256
D_{stem}	8
d_{attn}	64
L	16
$d_q(\text{Robot})$	7 or 14 (joint positions)
$d_q(\text{Human})$	6 or 12 (end effector pose as xyz + euler)
D_{trunk}	8
N_{trunk}	16
d	256
M	8
D_{head}	8
N_{head}	8
d_{head}	64
k	100
d_a	7 or 14 (xyz + euler angles + gripper position)
$L_{BC}(\text{Robot})$	$\text{SmoothL1}(\text{xyz}) + \text{SmoothL1}(\text{gripper}) + 0.5 \cdot \text{MSE}(\text{euler})$
$L_{BC}(\text{Human})$	$\text{SmoothL1}(\text{xyz})$
$\mathcal{L} = \mathcal{L}_{BC-\text{cotrain}} + \alpha \mathcal{L}_{OT-\text{joint}}$	$\alpha = 0.7$

B.1.1 Observation Encoder: f_ϕ

f_ϕ consists of two components, "stems" for each observation modality and a "trunk". The stems are shallow networks that encode heterogenous input spaces into a fixed representation to allow joint learning between human and robot observations.

Vision Stem. Given an RGB frame $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, we normalize using ImageNet normalization and then pass it through a ResNet-18 encoder truncated *before* the global-pool layer to obtain a $7 \times 7 \times 512$ feature map. The 49 spatial patches are flattened and projected with a single-layer MLP of hidden dimension d_{proj} , producing 49 tokens of size d_{proj} per image. A single multi-head cross-attention block (D_{stem} heads, per-head width d_{attn}) employs L learnable query tokens of dimension d that attend to these 49 patch tokens; the resulting L query outputs constitute the vision tokens passed to the shared trunk.

Proprioceptive Stem. The proprioceptive observation vector $\mathbf{q} \in \mathbb{R}^{d_q}$ (joint angles, end effector pose etc.) is z-score normalized and passed through a single-layer MLP of hidden dimension d_{proj} , producing one token $k_{\text{proprio}} \in \mathbb{R}^{d_{\text{proj}}}$. A single multi-head cross-attention block (D_{stem} heads, per-head width d_{attn}) uses the L learnable query tokens as in the vision branch to attend to this key/value token; the resulting L query outputs constitute the proprioceptive tokens that are forwarded to the shared trunk.

Trunk. The *shared* "trunk" is a standard multi-block transformer encoder, which consists of D_{trunk} heads, N_{trunk} blocks, and an embedding dimension of d . Each head has an attention dimension of d/D_{trunk} . A token sequence of length $L \cdot m$ by concatenating the token outputs of m observation encoding stems. A set of M learnable context tokens are prepended to the token sequence. The new sequence of $M + m \cdot L$ are input into the trunk. The first M are extracted from the output sequence and represent the feature output z of f_ϕ , and consequently where $\mathcal{L}_{OT-\text{joint}}$ is applied.

B.1.2 Policy: π_θ

π_θ is conditioned on the output z of f_ϕ and is parameterized by a DETR-style multi-block transformer decoder "head" [34].

Head. The head consists of N_{head} blocks and D_{head} heads, a cross-attention dimension of $d_{\text{head}} \cdot D_{\text{head}}$ and a self-attention dimension of $d_{\text{head}}/D_{\text{head}}$. This hybrid attention is designed to promote increased conditioning from the latent z . k learnable tokens of d_{head} are input into the

model, where K corresponds to the action chunk length k . Each block in the decoder consists of alternating self-attention on the k input tokens, and cross-attention with the context M tokens (z) from the trunk. After the final block, a linear layer projects the output to d_a corresponding to the cartesian action dimension, which is described precisely in Section D. Since the human data only comprises of 3D position, the loss is a masked loss which is only calculated on the values in the prediction corresponding to the 3D position for human data. The loss $\mathcal{L}_{BC}$ on human data is Smooth L1 loss on the 3D position, while the loss $\mathcal{L}_{BC}$ on robot data is Smooth L1 loss on the 3D position, gripper value and MSE loss on the orientation.

Complete Architecture. The network consists of a *shared* vision stem processes main egocentric RGB images ($I_{ego} \in \mathbb{R}^{H \times W \times 3}$) from both human and robot and produces L tokens. Followed by this are individual *embodiment-specific* proprioceptive stems that map q^H and q^R proprioceptive features to an additional L tokens. To the vision and proprioceptive tokens, left and right wrist RGB images ($I_{wrist} \in \mathbb{R}^{H \times W \times 3}$) are mapped to additional tokens by one additional wrist vision stem for single-arm robot data and two additional wrist vision stems for bimanual robot data. M learnable tokens are prepended to the token sequence and passed to the trunk. The OT-joint loss is applied on the first M tokens of the output. The same tokens are decoded by the head into $a_{t:t+k} \in SE(3) + gripper$, where an action for a single arm $\in R^{k \times 7}$ with the orientation expressed as euler angles in the yaw-pitch-roll convention. For bimanual tasks, the action comprises of two such action trajectories concatenated resulting in an output $\in R^{k \times 14}$.

Training Details. We train the real world EgoBridge model on a single L40s gpu for 100000 iterations on the Drawer task, 110000 iterations on the Laundry task, and 120000 iterations on the Scoop Coffee task, which takes about 24 hours. More details are in Table 4.

Table 4: Training Details for Real World Experiments

Parameter	Value
Optimizer	AdamW
Learning Rate	5×10^{-5}
Weight Decay	0.0001
Scheduler	Linear
Batch Size	32
Data Augmentations	ColorJitter + ImageNet Normalization (ResNet)
OT Loss	GeomLoss [37]
Blur	0.05
Distance	Sinkhorn

B.2 Sim PushT

Data Processing. We follow the PushT environment setup introduced in the Diffusion Policy benchmark suite [6]. Each demonstration comprises a sequence of RGB observations and associated low-dimensional proprioception, along with corresponding action labels. The RGB observations $I_t \in \mathbb{R}^{96 \times 96 \times 3}$ are normalized using ImageNet statistics. The proprioception consists of 2D end-effector positions (x, y) , which are min-max normalized per-dimension using statistics computed over the entire dataset. The action at each timestep is defined as the target 2D position of the end-effector, also normalized with per-dimension min-max normalization. Action chunks of length $k = 16$ are extracted from each demonstration trajectory. No observation history is used in the input, i.e., the observation context window is 1.

Encoder (f_ϕ). We employ the same vision encoder architecture as in the original Diffusion Policy paper [6], which uses a truncated ResNet-18 to extract spatial feature maps from $96 \times 96 \times 3$ RGB images. The proprioceptive input (normalized x, y position) is concatenated channel-wise to the image feature maps before being passed as the global conditioning to the UNet-based diffusion decoder. This architecture has been shown to effectively fuse visual and proprioceptive features for policy learning in low-dimensional manipulation tasks like PushT. The Joint-OT loss is applied on the image-proprio concatenated feature output.

Policy (π_θ). The policy architecture mirrors the conditional diffusion policy design [6]. The output is a denoising network trained to generate a trajectory segment of length $k = 16$ for 2D end-

effector coordinates in normalized space. During training, the policy receives a noisy version of the ground-truth action chunk, and learns to iteratively denoise the sample using the concatenated latent observations. The diffusion loss is the weighted MSE over all denoising steps in the sampling process. During inference, the policy performs iterative denoising starting from Gaussian noise to generate 2D trajectory samples conditioned on the current observation.

Training Details. We train the simulation EgoBridge model on a single A40 GPU for 130000 iterations, which corresponds to around 2 hours of training time. The hyperparameters and training details are summarized in Table 5.

Table 5: Training Details for Simulation Experiments (PushT)

Parameter	Value
Optimizer	AdamW
Learning Rate	1×10^{-4}
Weight Decay	1×10^{-6}
Scheduler	Cosine
Warmup Steps	500
Iterations	130,000
Batch Size	32
Exponential Moving Average (EMA)	Power = 0.75
Data Augmentations	ImageNet Normalization (ResNet)
$\mathcal{L}_{BC}$ (Triangle & Circle)	$MSE(xy)$
$\mathcal{L} = \mathcal{L}_{BC-cotrain} + \alpha \mathcal{L}_{OT-joint}$	$\alpha = 0.2$
OT Loss	GeomLoss [37]
Blur	0.01
Distance	Sinkhorn

C Baselines

C.1 Real World Experiments

All baselines instantiate the encoder f_ϕ and decoder π_θ architecture described in Section B and Table 3 with modifications to input and output spaces.

Robot-only Behavioral Cloning (Robot-only BC). This baseline is trained only on robot observations. The encoder includes vision stems for egocentric and wrist camera inputs and a proprioceptive stem for joint-space inputs. The resulting tokens and M context tokens are passed to the transformer trunk. The decoder head uses K learnable tokens to produce future actions, where each action is a 7-dimensional Cartesian command: 3D position, 3D orientation, and scalar gripper. The loss is computed using Smooth L1 for position and gripper, and MSE for orientation.

Co-train. Co-train uses both human and robot demonstrations during training and represents an ablated version of the EgoBridge model, where the joint-OT loss ($\mathcal{L}_{OT-joint}$) is not applied.

EgoMimic [1]. EgoMimic follows the Co-train architecture but includes two modifications: (1) The egocentric RGB input is masked with an overlay and red line augmentation, following [1] before it is input into the encoder, and (2) The decoder uses two heads: one for robot joint-space commands $\in R^7$ and another for shared cartesian pose $\in SE(3) + gripper$ for human and robot data. The joint-space predictions are supervised using a Smooth L1 loss on the entire prediction vector, while cartesian pose predictions are supervised by the loss described in Section B. During training, the appropriate head is selected based on the data source, where robot data provides gradient updates through the joint predictions and cartesian predictions, while the human data provides gradient updates for the model through the cartesian predictions. The identical overlay is applied during evaluation.

ATM [19]. For fair comparisons, we implement ATM with an identical architecture as our method and extend it with an additional head designed to predict future dense 2D motion trajectories in the image plane, as described in the original ATM paper. The goal is to test whether access to pixel-level dynamics improves downstream behavior cloning performance via improved latent structure.

Ground-truth 2D pixel tracks are generated using CoTracker [35] and organized into a temporal stack of length 10, covering 100 future frames. The action-track head outputs future 36 keypoints across 6×6 grid over the image plane. Training proceeds in two stages: first, the trunk, stems, and the 2D track head are pre-trained for 500 epochs to jointly predict human and robot 2D action tracks; second, the trunk and 2D head are frozen, and only the cartesian action head is fine-tuned on robot data. We opt for stacked, non-autoregressive, predictions of the full temporal keypoint trajectory, diverging from the sequential autoregressive formulation in the original ATM [19], in order to isolate the representational benefits of point-track conditioning on downstream imitation performance.

MimicPlay [25]. MimicPlay is implemented as a hierarchical policy which consists of a "high-level" policy which is co-trained on both human and robot data and a "low-level" policy which is trained only on robot data and is conditioned on the context output of the "high-level" policy.

High-level policy. We parameterize the high level policy of MimicPlay with a *shared* vision stem for I_{ego} for both human and robot and a shallow "trunk" which consists where $N_{trunk} = 2$, while the other hyperparameters for the trunk remain the same to match the parameter size of the original high-level in MimicPlay. 2 tokens corresponding to the high level context are prepended to the vision tokens. The first 2 output tokens are the latent context z_{high} that are passed to the low-level policy. A KL-divergence loss is applied between the robot data z_{high}^R and the human data z_{high}^H . A Gaussian Mixture Model decoder with identical parameters to the MimicPlay paper, takes z_{high} and predicts the mixing coefficient, mean, and the variance of GMM. With these predictions, a sequence of 100 3D positions are sampled from the GMM for a given input observation and supervised with the ground-truth actions for the observation with a negative log likelihood loss. The high-level planner is trained until convergence which is around 60000 iterations.

Low-level policy. The low level policy is parameterized the Robot-only BC model. The high-level policy is frozen and the latent output z_{high} is concatenated to the M tokens produced by f_ϕ which are then decoded into cartesian actions supervised by $\mathcal{L}_{BC}$ on robot data.

C.2 Sim PushT

In the more controlled Sim PushT environment, we benchmark EgoBridge against standard domain adaptation techniques commonly used in feature-level alignment. All baselines instantiate the same encoder f_ϕ and decoder π_θ described in Section B.2, with differences only in the training objective and data source used.

Maximum Mean Discrepancy [30]. We evaluate domain alignment using the MMD loss [30] as an alternative to the Joint OT loss. This baseline uses the same training configuration as EgoBridge but replaces the Joint-OT objective with the MMD loss applied on the latent embeddings from the shared encoder. The overall training objective is:

$$\mathcal{L} = \mathcal{L}_{BC-cotrain} + \lambda_{MMD} \cdot \mathcal{L}_{MMD}, \quad \text{where} \quad \lambda_{MMD} = 1.0$$

The MMD loss is computed as:

$$\mathcal{L}_{MMD} = \frac{1}{n^2} \sum_{i,j} k(z_H^{(i)}, z_H^{(j)}) + \frac{1}{m^2} \sum_{i,j} k(z_R^{(i)}, z_R^{(j)}) - \frac{2}{nm} \sum_{i,j} k(z_H^{(i)}, z_R^{(j)})$$

where $k(\cdot, \cdot)$ is a Gaussian RBF kernel:

$$k(x, y) = \exp \left(-\frac{\|x - y\|_2^2}{2\sigma^2} \right)$$

and σ is the kernel bandwidth hyperparameter.

Standard OT. This baseline implements marginal alignment using an unshaped Optimal Transport cost matrix without any DTW-based behavioural pairing. It follows the same formulation as EgoBridge (see Algorithm 1) but removes the DTW pairing step (Lines 11–17), using direct squared ℓ_2 distance $\tilde{C}_{ij} = D_{ij}$ for all (i, j) in the batch. This serves as an ablation that tests the benefit of behaviourally guided pairings in EgoBridge.

Co-train. Similar to the real-world setting, the co-train baselines ablates the Joint OT loss from the sim EgoBridge architecture to evaluate the baseline performance of training a shared policy on both embodiments.

Target-only BC. This baseline is trained using only robot demonstrations from the triangle pusher in the white background with original T configuration (100 trajectories total). It uses the same architecture as EgoBridge but is never exposed to data from the circle pusher (source domain).

D Robot Data

Bimanual Manipulator. To effectively utilize egocentric human data for manipulation, the robot hardware platform must resemble human sizes and kinematic workspaces. Drawing inspiration from the “Eve” robot platform introduced in EgoMimic [1], we develop a custom mobile manipulator that comprises of two 6-DoF ViperX 300s mounted in an identical inverted configuration on a height-adjustable rig. Similar to Eve, we propose to leverage the Project Aria glasses [32] as the main egocentric perception sensor for the robot and mount it in a way that emulates the hand-eye configuration of a human adult. This effectively mitigates the human-robot camera device gap and reduces the sensor-manipulator kinematic gap. Each arm is equipped with an Intel Realsense D405 on its wrist to facilitate precise near-range manipulation. The raw RGB data from the aria glasses is undistorted to a linear camera model, resized and reshaped to $480 \times 640 \times 3$. Fig. 6 shows an annotated picture of the bimanual manipulator. The bimanual manipulator is teleoperated using a leader-follower system similar to ALOHA [5], where two WidowX 6-DoF arms in an inverted configuration as seen in Fig. 6

Data Processing. Each robot demonstration comprises synchronized streams of joint-space proprioception, RGB image observations, and action labels recorded at 50Hz. At each timestep t , we collect the current joint configuration $\mathbf{q}_t^R \in \mathbb{R}^{d_q}$, where d_q denotes the number of actuated joints for one or two robot arms. RGB images are recorded from two sources: (1) an egocentric camera ($I_{ego,t}^R \in \mathbb{R}^{480 \times 640 \times 3}$) mounted using Project Aria glasses in a head-like configuration, and (2) wrist-mounted RGB cameras ($I_{wrist,t}^R \in \mathbb{R}^{480 \times 640 \times 3}$) placed near the end-effectors.

The raw action at each timestep is defined as the commanded joint position $\mathbf{a}_t^R \in \mathbb{R}^{d_q}$, representing the control input issued at time t . To obtain cartesian actions used for training, we use analytical forward kinematics to compute the $SE(3)$ cartesian pose. The cartesian pose is then projected into the image frame via the Aria glasses extrinsics (T_R^{aria}) obtained through hand-eye calibration, which represent the transformation between the robot base frame and the Aria glasses camera frame. The ground-truth actions $\mathbf{a}_t^R \in \mathbb{R}^{d_q}$ are obtained through

$$\mathbf{a}_t^R = (T_R^{aria})^{-1} \cdot \text{FK}(\mathbf{a}_{raw,t}^R)$$

E Embodied human data

We use the Project Aria [32] glasses to collect *human embodied experience data*. The glasses are accompanied by the Aria Machine Perception Services (MPS). The raw data from the Aria glasses consist of an RGB camera stream, SLAM cameras, IMU, eye cameras. This raw data is uploaded to a cloud service provided by Project Aria, known as the MPS. The service returns device pose estimated using the SLAM camera, hand tracking relative to the device frame, a semi-dense point-cloud of the environment and eye gaze estimation. This processed data is obtained as CSV files with aligned timestamps. The hand tracking is in the form of 3D positions $p^H \in SE(3) \times SE(3)$ for both hands in device frame, while the head pose $T^{aria} \in SE(3)$ in world frame. Each hand position p_t^H is in device frame T_t^{aria} for timestep t . The raw RGB data is undistorted to a linear camera model and reshaped to $480 \times 640 \times 3$. **Constructing human actions.** One challenge with unifying the reference frames for joint policy learning is that robot actions are typically in a fixed reference frame (usually the base frame), while human hand tracking is in the device frame, which moves. Following the idea of predicting action chunks [6, 5], we aim to construct action chunks $a_{t:t+k}^H$. Taking the single-arm case without loss of generality, the raw trajectory is $SE(3)$ poses $[p_t^H, p_{t+1}^H, p_{t+2}^H, \dots, p_{t+k}^H]$, where each hand position p_t^H is in device frame T_t^{aria} . We choose to create *pseudo-reference frames* taking inspiration from [1], where we construct an action $a_{t:t+k}^H$ by projecting k future hand positions into the device frame at timestep t . This allows the policy to predict trajectories with respect to a stable reference at each timestep. As such, the trajectory is constructed by:

$$a_{t:t+k}^H = \left[(T_t^{aria})^{-1} T_{t+i}^{aria} \cdot p_{t+i}^H \right]_{i=1}^k$$

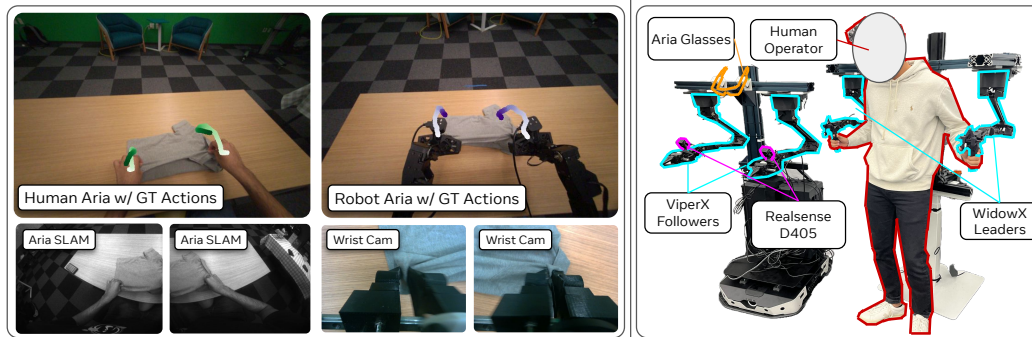


Figure 6: We employ the Aria glasses to capture Egocentric RGB images for both human and robot embodiments. The Aria uses its side SLAM cameras to estimate device pose and hand tracking (left). Ground truth (GT) cartesian actions for both embodiments are projected to pixel space using Aria intrinsics and visualized on the images. Our robot system is a leader-follower system with two Viper X arms as the followers and two Widow X arms as the leaders which are teleoperated by the human demonstrator (right).

F Data processing and alignment.

Recent cross-embodiment works [1, 38, 10] show that individually normalizing proprioception and actions for both embodiments, helps co-training. Similarly, we employ *z-score* normalization by subtracting the per-embodiment dataset mean and dividing by the standard deviation

$$\text{norm}(p) = (p - \mu_p) / \sigma_p$$

We normalize actions identically, using per-dimension mean and standard deviation.

For both human and robot, our action chunk size $k = 100$. Human data sensor streams (RGB, hand tracking and SLAM) are received at 30hz. We construct a human action chunk by taking 10 sample pose values at an interval of 3 frames, and interpolating them to a trajectory of length 100. This corresponds to a trajectory of length 100 to 0.9 seconds in real time. We construct a robot action chunk by concatenating the ground-truth actions of 100 successive timesteps which correspond to 2 seconds in real time. The human and robot data sources are visualized in Fig. 6.

G Additional Task Details

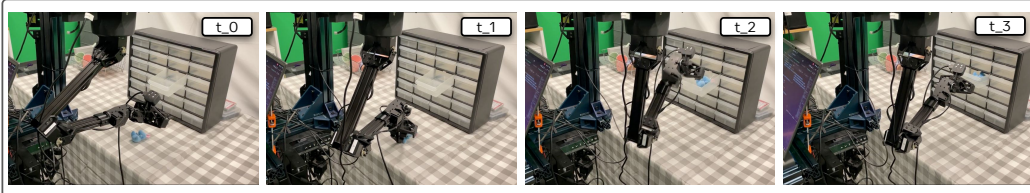
G.1 Real World Experiments

Drawer. For this task, we collect one hour of human data and 30 minutes of robot data comprising 144 demonstrations. The human data was equally distributed across all four quadrants, while the robot demonstrations were equally divided among three quadrants (excluding the top-right), with each receiving 48 robot demonstrations (8 demonstrations per drawer). The main failure modes for this task are the inability to correctly grasp the toy, taking the toy towards the incorrect drawer and being unable to completely push the drawer shut.

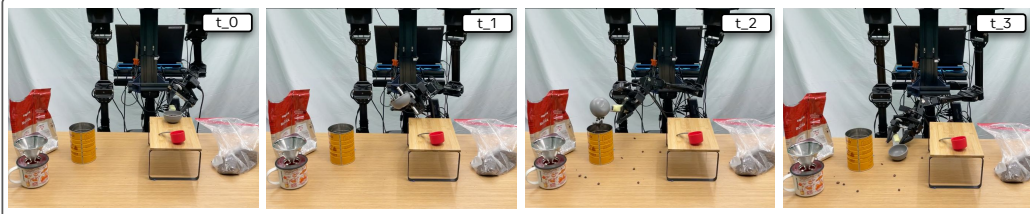
Scoop Coffee. The scoop coffee task emphasizes observation generalization with novel objects and scenes. To address this, we gathered 180 minutes of human data, equally split across three scenarios: the base scene, a new target object (grinder), and a new target object (grinder) with a new scene. For robot data, we collected 50 demonstrations with the original target object and scene configuration. The two primary failure modes in this task is an inability to precisely grasp the scoop and imprecise targeting of the target container.

Laundry. The laundry task demands precise bimanual coordination. We collected a similar amount of human data (2 hours) as the scoop task, comprising 700 demonstrations, alongside two hours of robot data (300 demonstrations). Common failure modes involve missing one or both sleeves during any of the three sub-stages, leading to an imprecise or unsuccessful fold. The qualitative task successes are visualized in Fig. 7 and common failure modes for each task are visualized in Fig. 8, while the data mixture is shown in Table 6.

a) Drawer



b) Scoop Coffee



c) Laundry

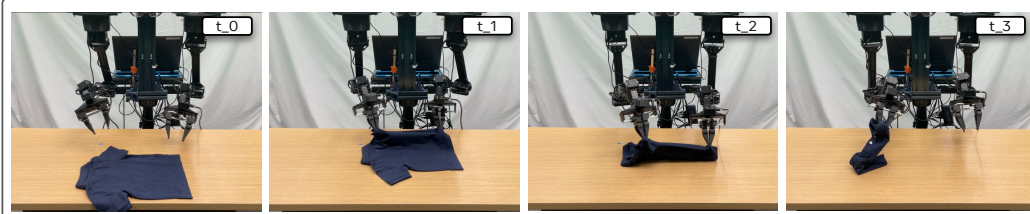
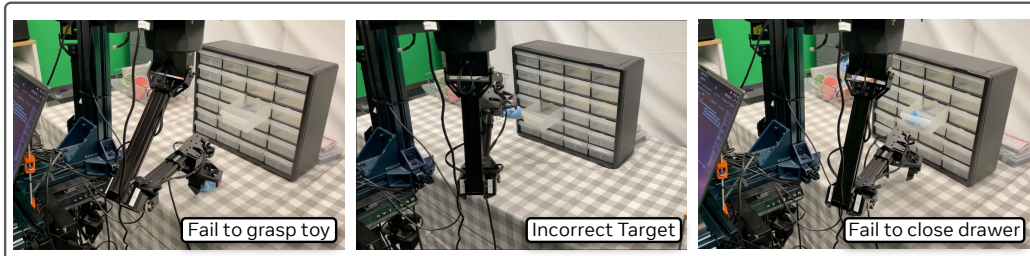
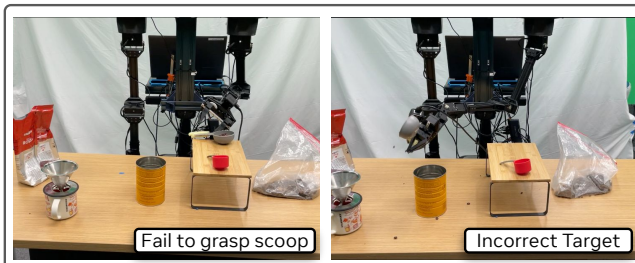


Figure 7: Qualitative success of EgoBridge on each of our real world tasks.

a) Drawer - **Failure** Modes



b) Scoop Coffee - **Failure** Modes



c) Laundry - **Failure** Modes

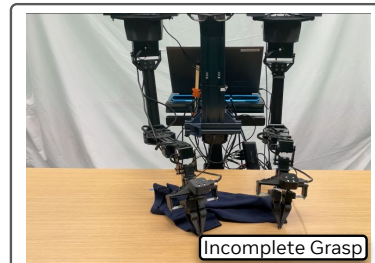


Figure 8: Common failure modes for each of the real world evaluation tasks.

Table 6: Data collection overview for both Human(H) and Robot(R) data. We report both the number(#) of total task demonstrations and the time(min) took to collect them.

Task	H #	H min	H #/min	R #	R min	R #/min
Drawer	360	60	6	144	29	5
Scoop Coffee	720	180	4	50	13.3	3.75
Laundry	700	120	5.8	300	120	2.5

G.2 Sim PushT

We collect a total of 350 demonstrations across 4 scenarios using the PyMunk simulator environment published with Diffusion Policy [6]. We collect 100 demonstrations each for the circle pusher and triangle pusher in the original T configuration with the white background. In addition, we collect 50 demonstrations each of the circle pusher with original T configuration on purple background, mirrored T on purple background and mirrored T on white background. We apply a friction coefficient multiplier of 0.7 on the triangle pusher to embody a kinematic gap like human and robot.

H Supplementary Simulation Results

We evaluate 3 settings : In distribution, which corresponds to the triangle pusher with the original T configuration on a white background; Observation Generalization which corresponds to the previous setting with a purple background; Observation and Behaviour Generalization which corresponds to the triangle pusher with a mirrored T on a purple background. For evaluation, we select a total of 100 seeds between 101 and 9999 using a deterministic random sampling with seed 42. The complete results are summarized in Table 7. The Mean Reward is calculated using the average max Intersection-over-Union with the goal across 100 seeds and Success Rate (SR) is computed using the number of seeds with a reward of over 0.9.

Table 7: Sim PushT results across 3 evaluation settings

Method	In-distribution		Generalization			
	(Mean Reward SR)		Purple Bg		Purple Bg + Mirrored T	
			(Mean Reward SR)		(Mean Reward SR)	
EgoBridge	0.7605	53.00%	0.7206	48.00%	0.6520	39.00%
Target Only	0.5555	39.00%	0.0904	0.00%	0.0992	0.00%
Cotrain	0.7062	48.00%	0.6899	42.00%	0.6214	31.00%
Standard OT	0.7009	38.00%	0.5303	15.00%	0.5109	8.00%
MMD	0.6439	45.00%	0.4867	22.00%	0.5876	14.00%

I Pseudo-pair Visualisation

To show why Dynamic Time Warping (DTW) is an ideal cost function to identify behaviourally similar trajectories to align, we visualize randomly sampled mini-batch pairs from the dataset for each of the tasks and qualitatively compare them with the Mean Square Error (MSE) cost function pairs used for the MSE ablation. A pair, as defined earlier, is the row-wise cost function minimum in the cost matrix computed from a batch of human and robot data. Qualitatively, DTW is more robust to temporal shifts, viewpoint shifts and is more spatially precise when compared to MSE. Specifically, MSE often pairs the incorrect sub-task in tasks like Laundry, and loses precise spatial location of the target container in tasks like Drawer and Scoop Coffee. The few failure modes of DTW are where the trajectory overlaps a few different stages which leads to temporal misalignments and when locations are spatially apart but visually close due to an extreme viewpoint shift. Despite this, DTW enables pairings that are able to capture very fine-grained changes in trajectory to find the most behaviorally similar between human and robot data. This is visualized in Fig. 9.



Figure 9: Qualitative visualization of randomly sampled mini-batch pairing of MSE and DTW.

J Supplementary Real Results

In order to test EgoBridge’s ability to generalize to new appearances under a more challenging real-world setting, we conduct two additional evaluation experiments on the **Laundry** task. The first, *Observation Generalization*, introduces a new white table surface unseen in robot data, along with 30 minutes of human data recorded on this new setup. The second, *Combinatorial Observation Generalization*, evaluates a harder scenario where the robot must handle a new shirt color on the white surface, with that shirt seen only in the human data and excluded entirely from robot demonstrations. Both evaluations follow the same protocol as the main experiments, performing 20 rollouts across 5 randomly sampled workspace positions. As summarized in Table 8, EgoBridge achieves substantially higher success rates and sub-task points than the Co-train baseline in both settings, demonstrating stronger robustness to visual appearance shifts and compositional variations in object–scene configurations.

Table 8: **Generalization experiments on Laundry.** Evaluation of Co-train and EgoBridge under two generalization settings. We report success rate (SR) and sub-task points.

Model	Obs. Generalization		Comb. Obs. Generalization	
	SR (%)	Points	SR (%)	Points
Co-train	25	25	5	7
EgoBridge	50	43	30	35



Figure 10: Qualitative visualization of the learned latent space using T-SNE and K-Nearest Neighbors.

K Latent Space Visualisation

We visualize the learned latent feature outputs of f_ϕ for the EgoBridge and Co-train models using T-SNE for the Drawer and Scoop Coffee tasks. To evaluate EgoBridge’s ability to achieve joint distribution alignment, we use K-Nearest Neighbors to identify the closest learned human feature representations to a give robot feature representation. We plot the ground truth actions associated with those observations onto the input images and visualize the nearest neighbors. In addition to an overall lower Wasserstein-2 distance, which indicates global alignment, EgoBridge also aligns observations that correspond to similar behaviors in the latent space. This is visualized in Fig 10.

NeurIPS Paper Checklist

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

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

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

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

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

IMPORTANT, please:

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

1. Claims

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

Answer: [Yes]

Justification: The experiment section show real world and simulation evaluation settings that support the claims made within the abstract and the introduction.

Guidelines:

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

2. Limitations

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

Answer: [Yes]

Justification: We have described our limitations alongside the conclusion section.

Guidelines:

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

3. Theory assumptions and proofs

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

Answer: [Yes]

Justification: The purposed method and corresponding optimal transport theory contains a detailed proof and is included in the supplementary material.

Guidelines:

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

4. Experimental result reproducibility

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

Answer: [Yes]

Justification: The EgoBridge framework details and hardware used in experiments are discussed in the methods and experiments section.

Guidelines:

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

5. Open access to data and code

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

Answer: [No]

Justification: No. But we plan to release after acceptance.

Guidelines:

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

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

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

Answer: [\[Yes\]](#)

Justification: Detailed information regarding data collection, splits, hyperparameter tuning and training details of EgoBridge is mentioned in our supplementary material.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [\[No\]](#)

Justification: In real-world robot learning experiments [1, 25, 19], conducting the numerous repetitions on physical robots required for robust error bar computation is prohibitively expensive and time-consuming, incurring significant hardware and human supervision costs. Moreover, repeated trials can induce hardware wear, introducing uncontrolled variability that hinders achieving consistent experimental setups. Consequently, reporting standard error bars is often impractical. Our evaluation prioritizes full task and sub-task success and qualitative results over statistical variance, aligning with standard evaluation methods in this domain.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

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

Answer: [\[Yes\]](#)

Justification: Detailed instructions of compute resources to reproduce our work are provided in the supplementary material

Guidelines:

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

9. Code of ethics

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

Answer: [\[Yes\]](#)

Justification: We do not conduct formal human study. Demonstration data (human and robot) is collected by authors with explicit consent. We have reviewed the NeurIPS Code of Ethics and ensured full compliance with its guidelines for this research.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[NA\]](#)

Justification: Our work advances co-training methodologies for cross-embodied human and robot datasets, representing a small increment in foundational robot learning research. As such, it does not present direct societal impacts at this stage.

Guidelines:

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

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

11. Safeguards

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

Answer: [NA]

Justification: We have not released our data or models and we believe our potential future release of this paper does not pose any such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly cited all existing code package and datasets and mentioned their licenses in the supplementary material.

Guidelines:

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

13. New assets

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

Answer: [NA]

Justification: This work does not release new assets.

Guidelines:

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

14. Crowdsourcing and research with human subjects

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

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human participants. All human demonstration data is collected by authors with explicit consent.

Guidelines:

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

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

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

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human participants. All human demonstration data is collected by authors with explicit consent.

Guidelines:

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

16. Declaration of LLM usage

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

Answer: [NA]

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

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.