BATCH: BEHAVIOR ALIGNMENT FOR CROSS-EMBODIMENT LEARNING FROM HUMANS

Anonymous authors
Paper under double-blind review

ABSTRACT

Human demonstrations provide a scalable source of diverse manipulation data, but leveraging them for robot learning remains challenging due to differences in embodiment, viewpoint, action spaces, and temporal structure. We introduce **BATCH**, a cross-embodiment learning framework that explicitly aligns human and robot behaviors. BATCH learns self-supervised temporal representations with a cycle-consistency objective to establish cross-embodiment correspondences without manual annotation. We further introduce two complementary training objectives that leverage these correspondences within an autoregressive video-action diffusion model: behavior-aligned co-training and human-to-robot style transfer. On real-world manipulation tasks, BATCH improves generalization to unseen task configurations compared with training on unaligned data and prior cross-embodiment baselines, with additional gains from the style-transfer objective. Our results demonstrate that explicit behavior alignment enables more effective transfer from human demonstrations to robot policies. Anonymous website: `https://batch-learning.github.io/`.

1 INTRODUCTION

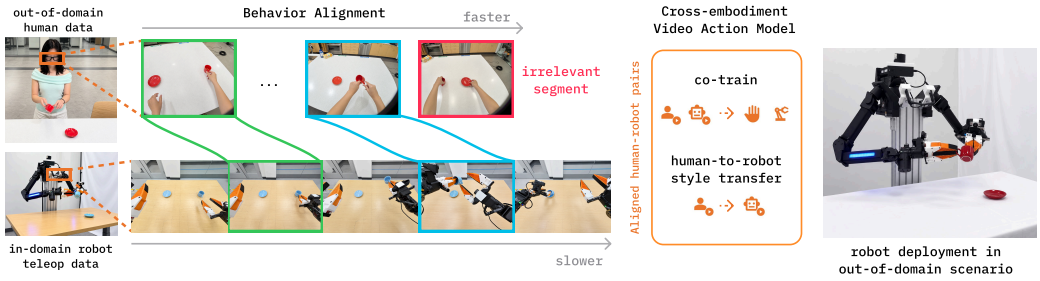


Figure 1: **BATCH** performs behavior alignment between diverse out-of-domain human data and limited in-domain robot teleop data, accounting for differences in execution speed and irrelevant human segments. The aligned pairs are leveraged through two objectives within a downstream cross-embodiment video action model: co-training and human-to-robot style transfer. The resulting policy transfers visual knowledge from human data to robot deployment in out-of-domain scenarios.

Human demonstrations offer a scalable way to expand the scene and behavior diversity of robot datasets. In particular, wearable egocentric sensing devices, such as smart glasses (Engel et al., 2023), provide a lightweight and portable solution for capturing natural human manipulation across diverse environments, while on-device SLAM cameras enable accurate hand-pose tracking. Despite this promise, directly leveraging human demonstrations for robot learning remains challenging due to the substantial embodiment gap between humans and robots, including both visual appearances and physical motions.

Prior studies have shown that directly co-training on cross-embodiment data can improve visual robustness and policy generalization to unseen backgrounds and objects (Kareer et al., 2025; Yang et al., 2026). Yet cross-embodiment data is not simply additional data from a different visual domain: even when performing the same task, humans and robots can differ substantially in execution speed, temporal structure, action spaces, and visual observations due to differences in embodiment. Human

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

demonstrations may also contain irrelevant segments that have no corresponding robot behavior. As a result, naively scaling human demonstrations does not necessarily provide useful supervision for robot behavior. Instead, models can separate the two embodiments and rely on embodiment-specific cues, limiting the extent to which human experience transfers to robot control. Empirically, this can lead to diminishing gains or even negative transfer as more poorly matched human data is incorporated (Wei et al., 2025; Punamiya et al., 2026). We therefore hypothesize that effective cross-embodiment transfer requires explicitly establishing behavioral correspondences between human and robot trajectories, providing a learning signal for identifying task-relevant structure that should transfer while discounting embodiment-specific variation.

We define **behavior alignment** as establishing temporal frame-level correspondences between equivalent phases of task execution across embodiments, based on task-relevant states and interactions rather than embodiment-specific appearance, viewpoint, or action representation. The key challenge is to discover these correspondences automatically, without requiring manually annotated phase labels or frame-level matches between human and robot trajectories. Achieving this requires features that capture behavioral progression over time while suppressing embodiment-specific variation. Prior non-learning approaches commonly apply dynamic time warping (DTW) over actions or static visual features, which can be unreliable under large embodiment gaps or when visual states are ambiguous. Instead, we learn self-supervised features with a temporal cycle-consistency objective (Dwibedi et al., 2019; Donahue & Elhamifar, 2024), where each frame representation incorporates information from multiple neighboring frames to capture behavioral progression rather than relying on a single observation. We then apply subsequence DTW in this learned feature space to retrieve temporally corresponding human and robot segments, accommodating differences in execution speed and unmatched segments.

Another key contribution of our approach is how we leverage the aligned trajectories for cross-embodiment learning. We instantiate our method within an autoregressive video-action diffusion model (Ye et al., 2026) pretrained on in domain robot demonstrations, and finetune with two complementary objectives over aligned human and robot trajectories. A behavior-aligned co-training objective learns visual dynamics and motions from both embodiments, while a cross-embodiment style-transfer objective translates human observations into temporally corresponding robot-domain videos. Together, these objectives learn from both embodiments while explicitly coupling them under matched behavior, enabling more effective transfer from diverse human demonstrations to robot policies.

We evaluate our approach on two manipulation tasks from EgoVerse (Punamiya et al., 2026): single-arm *Object in Container*, which requires continuous execution, and bimanual *Cup on Saucer*, which requires precise grasping and bimanual coordination. We consider out-of-domain test scenarios presented in the human demonstrations but absent from the robot training data. Across the two tasks, our approach improves out-of-domain policy performance by an average of 45% over co-training alone and 138% over the robot-only pretrained model. Removing behavior alignment substantially degrades performance, highlighting the importance of reliable human-robot correspondences, while the style-transfer objective provides further gains with aligned pairs. These results demonstrate that both establishing behavioral correspondences and effectively leveraging them are critical for cross-embodiment transfer.

2 Related Work

Video Temporal Alignment. Video temporal alignment establishes frame-level correspondences across videos of similar behaviors despite variations in embodiment, viewpoint, and appearance. One foundational class of approaches is Temporal Cycle-Consistency (TCC) (Dwibedi et al., 2019; Hadji et al., 2021), which learns a shared embedding space from unaligned videos without manual labels by enforcing cycle-consistent nearest-neighbor matches. GTCC (Donahue & Elhamifar, 2024) further improves the robustness of TCC through multi-neighbor, multi-cycle matching modeled with Gaussian mixtures. In robotics, TCC-style representations have been used to derive dense task-progress rewards for cross-embodiment learning (Zakka et al., 2022). Separately, Dynamic Time Warping (DTW) is a common technique used to align cross-embodiment demonstrations with different execution speeds (Punamiya et al., 2025; Liu et al., 2025; Yang et al., 2026). Our approach combines GTCC and DTW: GTCC learns cross-video correspondences in a shared embedding space, while DTW produces globally consistent temporal alignments across human and robot demonstrations while accounting for different execution speeds.

Cross-Embodiment Learning from Human Data. Human demonstrations provide scalable manipulation data with greater scene and behavior diversity than costly robot demonstrations, but effective transfer remains challenging due to embodiment gap. Prior attempts include learning shared skill representations (Xu et al., 2023; Kim et al., 2025), using transferable intermediate representations such as object flow (Xu et al., 2025), or editing human videos into robotized demonstrations (Lepert et al., 2025a;b; Wang et al., 2026). However, these approaches can be limited in scalability with additional processing and tuning. More recent approaches directly co-train on human and robot data (Kareer et al., 2025; Qiu et al., 2025; Yuan et al., 2025; Tao et al., 2025; Zhu et al., 2026; Punamiya et al., 2026), with follow-up works seeking to improve cross-embodiment transfer through observation alignment (Punamiya et al., 2025), embodiment-agnostic visual prediction (Li et al., 2026a), or image- and action-space alignment (Li et al., 2026c). Our work studies whether temporal behavior alignment between human and robot trajectories can further support cross-embodiment transfer while integrating naturally into existing co-training formulation.

Video Action Models jointly model future visual observations and robot actions, providing richer supervision by learning scene dynamics alongside robot control. Early methods such as GR-1 and GR-2 combine visual generation with action prediction (Wu et al., 2024; Cheang et al., 2024). More recent approaches, including UVA (Li et al., 2025), UWM (Zhu et al., 2025), and VideoVLA (Shen et al., 2025), more tightly couple video and action modeling, while Cosmos Policy (Kim et al., 2026), DreamZero (Ye et al., 2026), LingBot-VA (Li et al., 2026b), and LingBot-VA 2.0 (Zhang et al., 2026) scale this paradigm with large video backbones and video-action pretraining. Our work builds on this paradigm to study cross-embodiment policy learning from behaviorally aligned human and robot trajectories.

3 METHOD

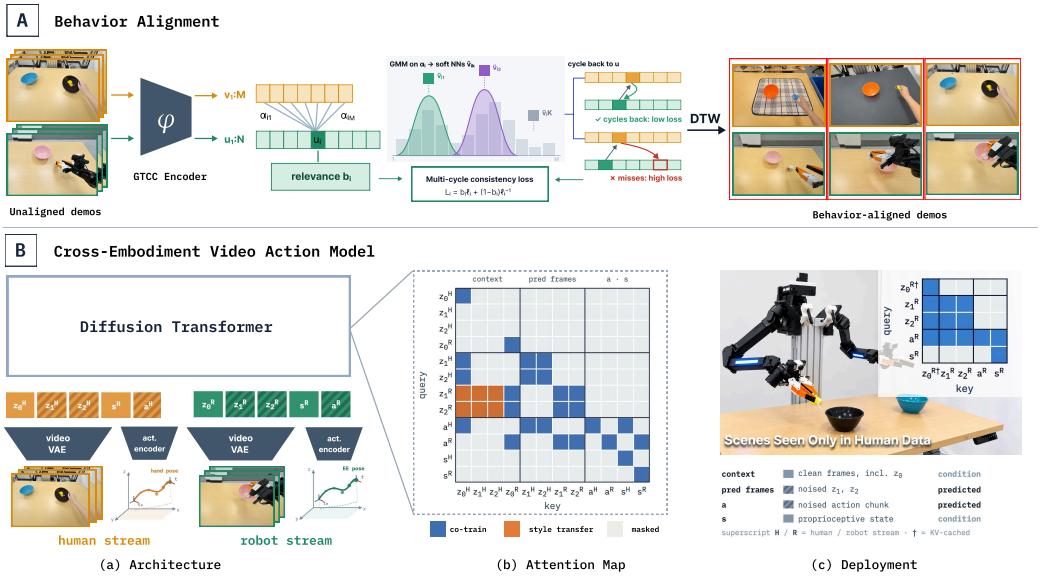


Figure 2: **BATCH method.** (A) *Behavior Alignment (Section 3.1)*: GTCC encoder trained with multi-cycle consistency learns shared temporal embeddings from unaligned demonstrations; the trained encoder then combines with DTW to retrieve behavior-aligned human and robot segments. (B) *Cross-Embodiment Video Action Model* (Section 3.2): (a) aligned human and robot trajectories are encoded as separate streams and input to a diffusion transformer; (b) the attention pattern supports both joint co-training and human-to-robot style transfer objective within the same model; and (c) at deployment, the robot stream is used to execute policy in scenarios only seen in human data.

3.1 CROSS-EMBODIMENT BEHAVIOR ALIGNMENT

We consider human and robot demonstrations of the same manipulation task. Following EgoVerse (Punamiya et al., 2026), human demonstrations are collected using Aria glasses, while robot demonstrations are collected through teleoperation (see Appendix Section A for data collection

details). Despite sharing the same task objective, the two datasets can differ substantially in embodiment, background, viewpoint, and execution speed, making direct frame-level matching unreliable. We therefore seek a representation that is robust to these cross-embodiment differences while preserving temporal task progress. Starting from pre-extracted image features, we learn a shared temporal embedding space using the self-supervised cycle-consistency objective of Donahue & Elhamifar (2024), encouraging human and robot frames at corresponding stages of task execution to be close in the learned space. These correspondences are characterized by similar object states and interactions rather than visual appearance alone. To further account for differences in execution speed, we recover explicit frame-level correspondences using subsequence dynamic time warping (DTW), which finds the minimum-cost alignment in the learned embedding space while allowing non-uniform temporal warping.

3.1.1 Problem Setup

Given a set of robot demonstration segments $\mathcal{D}^{\mathrm{r}} = \{X_i^{\mathrm{r}}\}_{i=1}^{N_{\mathrm{r}}}$ and human demonstration segments $\mathcal{D}^{\mathrm{h}} = \{X_j^{\mathrm{h}}\}_{j=1}^{N_{\mathrm{h}}}$, where each segment corresponds to a temporal portion of the same manipulation task, our goal is to learn a shared encoder f_θ that maps egocentric head-camera videos from both embodiments into sequences of per-frame embeddings. For a robot query segment X^{r} of length N and a human target segment X^{h} of length M, we denote the corresponding embeddings as $\mathbf{u}_{1:N} = f_\theta(X^{\mathrm{r}})$ and $\mathbf{v}_{1:M} = f_\theta(X^{\mathrm{h}})$, respectively.

Feature extraction. We extract per-frame feature maps of shape $1024 \times 14 \times 14$ using a frozen ImageNet ResNet-50 encoder (He et al., 2016). For each frame t, features from the causal temporal window $[t-k+1, t]$ of length k are stacked and processed by two sequential 3D convolutional layers. The resulting representation is max-pooled over space and time and projected by a three-layer MLP to a 128-dimensional embedding.

Multimodal soft nearest neighbors. As illustrated in Figure 2, for each robot query frame i, we compute a soft nearest-neighbor distribution over the human target sequence based on distance in the learned embedding space,

$$\alpha_{ij} = \frac{\exp\bigl(-\|\mathbf{u}_i - \mathbf{v}_j\|_2^2/\tau\bigr)}{\sum_{j'} \exp(-\|\mathbf{u}_i - \mathbf{v}_{j'}\|_2^2/\tau)}, \qquad j = 1, \ldots, M,$$

τ denotes temperature. Standard TCC (Dwibedi et al., 2019) summarizes this distribution with a single soft nearest neighbor, $\tilde{\mathbf{v}}_i = \sum_j \alpha_{ij}\mathbf{v}_j$. However, manipulation trajectories may revisit similar task states, so a query frame can have multiple plausible correspondences at different target times. Averaging across these modes can therefore produce a representation that matches none of them. Following Donahue & Elhamifar (2024), we instead approximate α_i with a K-component Gaussian mixture over target time,

$$\min_{\{\pi_{ik},\mu_{ik},\sigma_{ik}\}_{k=1}^{K}} D_{\mathrm{KL}}\left(\alpha_i \,\middle\|\, \sum_{k=1}^{K} \pi_{ik}\mathcal{N}(\cdot; \mu_{ik}, \sigma_{ik}^2)\right),$$

where K is the number of mixture components, and π_{ik}, μ_{ik}, and σ_{ik} denote the weight, temporal center, and spread of component k. Each component defines a normalized distribution $\hat{g}_{ik}$ over target frames and yields a separate soft nearest neighbor, $\tilde{\mathbf{v}}_{ik} = \sum_j \hat{g}_{ikj}\mathbf{v}_j$. This preserves multiple plausible correspondence hypotheses rather than collapsing them into a single averaged match.

Training objective. For each query frame i, we minimize a multi-cycle consistency loss

$$\ell_i = \sum_{k=1}^{K} \pi_{ik}\left[(i - \hat{\mu}_{ik})^2 + \lambda \log \hat{\sigma}_{ik}\right],$$

where $\hat{\mu}_{ik}$ and $\hat{\sigma}_{ik}$ denote the mean and uncertainty of the cycle-back distribution for correspondence hypothesis k, and λ controls the uncertainty term. This objective encourages each hypothesis to cycle back to the original temporal location with low uncertainty. To handle frames without a meaningful counterpart in the other embodiment, an auxiliary head predicts a relevance weight b_i for each embedding, and we optimize $L_i = b_i\ell_i + (1 - b_i)\ell_i^{-1}$. Thus, alignable frames are encouraged to achieve low cycle-consistency error, while unalignable frames are down-weighted.

3.1.2 Dataset Generation with DTW Retrieval

To construct behavior-aligned human-robot pairs, we use subsequence DTW in the learned embedding space. Given a robot query window $\mathbf{u}_{1:L}$ and a candidate human sequence $\mathbf{v}_{1:T}$, we define the pairwise embedding cost as $C_{ij} = \|\mathbf{u}_i - \mathbf{v}_j\|_2$. Subsequence DTW finds the minimum-cost human sub-segment while allowing temporal warping to account for differences in execution speed. For each human candidate, we select the endpoint $j^\star = \arg\min_j D[L, j]$, where $D[L, j]$ is the cumulative DTW cost of aligning the full robot query to a human sub-segment ending at j, and backtrack from $j^\star$ to recover the matched sub-segment. We rank all human candidates by their normalized DTW cost and retain the top-k lowest-cost matches.

3.2 Cross-Embodiment Video Action Model

Given the aligned human and robot trajectories, we train a cross-embodiment video action model aiming to transfer task-relevant visual knowledge from human demonstrations to the robot domain. Our model (shown in Figure 2B) builds on a video action diffusion transformer (Ye et al., 2026) that jointly predicts future visual observations and chunks of robot actions. Training consists of two stages: we first pretrain on robot demonstrations, then finetune on aligned human-robot trajectories with two complementary objectives: (1) *joint co-training*, which processes human and robot trajectories as attention-isolated streams with shared model parameters, and (2) *human-to-robot style transfer*, which predicts robot future video from the aligned human video, explicitly encouraging human-to-robot visual transfer. At deployment, the model predicts actions from unseen robot observations observed only in human data.

Backbone. Our policy builds on Wan2.1-I2V-14B (Team & Group, 2025). We retain its first 8 transformer blocks, video VAE, and CLIP ViT-H/14 image encoder (Ilharco et al., 2021). Since we consider a single-task setting, we remove text conditioning.

Tokenization. Each training sample consists of 9 head-camera frames, a 16-step future action chunk, the current proprioceptive state, and one conditioning frame from each wrist camera. For both human and robot demonstrations, states and actions are expressed in the coordinate frame of the Aria glasses. The video VAE compresses the 9 head-camera frames into three latent frames, where the first latent frame provides conditioning context and the remaining two form a causal block to be predicted.

For each embodiment, the transformer input is organized into four token groups: (1) *clean video tokens*, obtained by patchifying uncorrupted latent frames on a shared canvas by concatenating the head and wrist latents along the width dimension; (2) *noisy video tokens*, obtained by patchifying the noised head-camera latents; (3) a *register* consisting of 16 action tokens and one state token, projected using embodiment-specific encoders; and (4) *CLIP context tokens*, obtained by encoding the first frame from each available camera into 257 tokens per image and injected through cross-attention. The transformer predicts flow velocities at the noisy video and action token positions, with the corresponding prediction targets determined by the training objective. Token counts and remaining details are given in Appendix Table 2.

Human and robot streams. For each aligned human-robot pair sampled from the dataset described in Section 3.1.2, we construct human and robot token streams and concatenate them into a single transformer sequence. Both streams share the same transformer weights; they differ only in their inputs and embodiment-specific state/action encoders and decoders. The robot stream receives head and wrist camera observations, whereas the human stream contains only the head-camera observation, with the missing wrist-camera regions zero-padded.

As shown in Figure 2 (b), attention within each stream follows a block-causal structure. Noisy video tokens attend to the clean visual context and to tokens within their own noisy video block. Action tokens additionally attend to the noisy video block and the current state token. We further design the two streams so that each can be processed independently of whether the other is present. First, rotary positional indices are reset at the beginning of each stream, preventing the positional representation of one embodiment from depending on the length of the other. Second, CLIP cross-attention is separated by embodiment: robot queries attend only to robot CLIP tokens, while human queries attend only to human CLIP tokens. As a result, concatenating the two streams does not alter the within-stream computation, while still allowing them to share weights.

3.2.1 TRAINING OBJECTIVES

We train video and action prediction with flow matching. A clean latent $\mathbf{x}_0$ is corrupted to $\mathbf{x}_\sigma = (1-\sigma)\mathbf{x}_0 + \sigma\boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, and the network regresses the velocity $\boldsymbol{\epsilon} - \mathbf{x}_0$. Noise levels are sampled independently for video blocks and action tokens. We write $\mathcal{L}^e_{\text{vid}}$ and $\mathcal{L}^e_{\text{act}}$ for the flow-matching losses on the future head camera latents and on the action chunk of embodiment $e \in \{\text{r}, \text{h}\}$.

Robot pre-training. We first train on robot demonstrations for 20,000 steps with just robot stream:

$$\mathcal{L}_{\text{pre}} = \mathcal{L}^{\text{r}}_{\text{vid}} + \lambda_a \mathcal{L}^{\text{r}}_{\text{act}}.$$

Joint co-training. We then fine-tune the model on aligned human-robot pairs. The two streams share model parameters but remain *attention-isolated*: human tokens attend only within the human stream, and robot tokens only within the robot stream. Each stream predicts its own future video and actions:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}^{\text{r}}_{\text{vid}} + \lambda_a \mathcal{L}^{\text{r}}_{\text{act}} + \lambda_{\text{hv}} \mathcal{L}^{\text{h}}_{\text{vid}} + \lambda_{\text{ha}} \mathcal{L}^{\text{h}}_{\text{act}}.$$

Thus, human demonstrations affect the robot pathway only through the shared model parameters while preserving the robot-only inference pathway.

Human-to-robot style transfer. The second fine-tuning objective explicitly couples the aligned streams by predicting the robot future video from the corresponding human video clip. The full human clip is provided as clean context, while the robot future latents are noised and denoised:

$$\mathcal{L}_{\text{ST}} = \mathbb{E}_{\sigma,\boldsymbol{\epsilon}} \left\| v_\theta \left(\mathbf{x}^{\text{r}}_\sigma \,\middle|\, \mathbf{x}^{\text{h}}_0, \mathbf{c}^{\text{r}} \right) - (\boldsymbol{\epsilon} - \mathbf{x}^{\text{r}}_0) \right\|_2^2,$$

where $\mathbf{c}^{\text{r}}$ denotes the robot CLIP context. To prevent the model from bypassing the human stream, we mask the robot conditioning latent and zero both action/state registers. The robot retains only the CLIP embeddings of its initial head and wrist observations for coarse scene context. This encourages the robot future prediction to rely on the aligned human trajectory together with the robot scene context. Only the robot video is supervised under this objective.

During fine-tuning, we train for 1,000 steps and alternate between $\mathcal{L}_{\text{joint}}$ and $\mathcal{L}_{\text{ST}}$, each with probability 0.5. The joint objective learns future motion and dynamics within each embodiment, while the style-transfer objective establishes cross-embodiment associations. At deployment, only the robot stream is used. The policy predicts 16-step action chunks and synchronously executes the first 8 actions before replanning. See Appendix Section B for more details on the training procedure.

4 EXPERIMENTS

4.1 BEHAVIOR ALIGNMENT EVALUATION

We evaluate whether the learned representation supports temporal alignment between robot and human demonstrations. Given a robot query segment, we retrieve and align the corresponding segment from human demonstrations, and evaluate the resulting frame-level correspondences against ground-truth phase annotations.

Tasks and evaluation protocol. We evaluate on two tasks with distinct temporal structures and embodiment-dependent speed differences, shown in Figure 3. In *Object in Container*, each episode contains 6-7 repetitions of the four-phase manipulation cycle, creating multiple valid occurrences of the same phase within a trajectory. Human demonstrations execute the cycle approximately $2\times$ faster than the robot. In *Cup on Saucer*, each episode progresses through three phases, with the human operating $3.4\times$ faster than the robot. For both tasks, we evaluate against an in-domain (ID) human set collected in the same environment as the robot demonstrations and an out-of-distribution (OOD) set collected in different scenes. For each robot episode, we sample 400 evenly spaced query windows. Each query is aligned against every human episode in the candidate set. We retain the minimum-cost alignment from each human episode and average the resulting alignment metrics. Details of the evaluation dataset can be found in the Appendix Table 2 and Section C.1.

We compare several representation and matching variants. GTCC w/ DTW applies DTW to temporal embeddings learned with GTCC, while GTCC w/o DTW uses the same embeddings with rigid

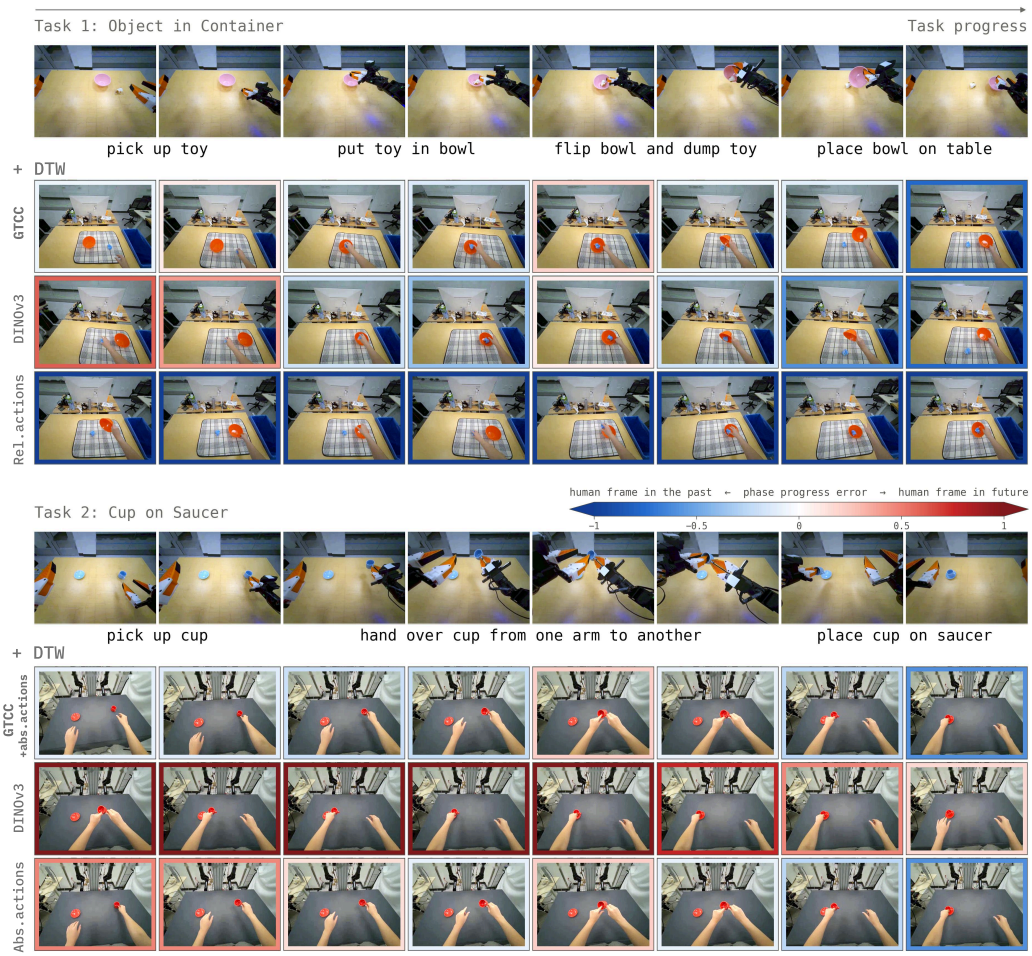


Figure 3: **Qualitative alignment results.** For each task, the top row shows robot query frames and the rows below show the corresponding human frames retrieved by each method. Border color denotes signed phase-progress error (blue: lagging, white: matched, red: ahead). Our method retrieves human frames that more closely follow the robot's task progress.

sliding, isolating the effect of temporal warping. GTCC + rel./abs. actions concatenates each GTCC feature with a 7-DoF end-effector pose per arm before applying DTW. We consider either absolute poses or poses expressed relative to the first frame of the query window. Rel./abs. actions performs DTW using these pose features alone, isolating the contribution of action information without GTCC features. DINOv3 w/ DTW uses frozen DINOv3 ViT-S/16 (Siméoni et al., 2025) class-token features to compare GTCC against a generic visual representation, and DINOv3 + rel./abs. actions augments them with the corresponding end-effector poses described above. Proportional mapping matches frames according to normalized temporal position within the target episode.

Phase overlap. We measure coarse phase alignment as the fraction of matched frame pairs that share the same phase label, i.e., the number of correctly matched frames divided by the total number of matched frames.

Phase progress error. Phase overlap does not distinguish between frames at different points within the same phase. To measure finer temporal alignment, we assign each frame a progress value that encodes both its phase and its progress within that phase. Specifically, for a frame f in phase k_m, spanning frames $s_m, \dots, e_m$, we define

$$p(f) = \frac{1}{K}\left(k_m + \frac{f - s_m}{e_m - s_m + 1}\right), \qquad E = \frac{1}{|\mathcal{P}|}\sum_{(f_q, f_t)\in\mathcal{P}} \delta(p(f_q), p(f_t)) \,. \tag{1}$$

	Object in Container				Cup on Saucer			
	Overlap (%) ↑		Prog. err. ↓		Overlap (%) ↑		Prog. err. ↓	
Method	**ID**	**OOD**	**ID**	**OOD**	**ID**	**OOD**	**ID**	**OOD**
GTCC variants								
GTCC + DTW	**69.0**	**57.8**	.061	**.098**	64.8	71.7	.088	.075
(−) DTW	61.8	24.7	.080	.231	61.9	70.8	.092	.085
(+) rel. actions	68.9	54.9	**.060**	.104	61.8	62.7	.100	.123
(+) abs. actions	43.7	54.6	.124	.106	**79.2**	**76.5**	**.070**	**.073**
Feature and action baselines								
DINOv3 + DTW	39.4	42.3	.148	.156	48.9	22.1	.294	.433
(+) rel. actions	35.1	43.9	.171	.153	49.1	25.3	.282	.377
(+) abs. actions	22.1	35.6	.234	.178	45.7	33.8	.222	.317
Rel. actions + DTW	25.6	39.6	.229	.169	34.2	29.2	.390	.433
Abs. actions + DTW	17.1	33.8	.264	.183	70.4	67.1	.118	.123
Temporal baseline								
Proportional mapping	34.9	25.5	.194	.218	32.5	60.4	.217	.139

Table 1: **Cross-embodiment alignment results.** We report frame-level overlap and phase-progress error on in-domain (ID) and out-of-domain (OOD) human demonstrations. GTCC yields the strongest visual-only alignment across both tasks, substantially outperforming DINOv3 features, while DTW further improves alignment, particularly for *Object in Container*. Pose cues provide limited benefit when the action gap is large, but become more informative for *Cup on Saucer*, where combining absolute end-effector poses with GTCC achieves the best overall performance.

Here, $p(f) \in [0, 1)$ represents the frame's normalized position within one task cycle, and E is the average absolute progress difference between matched query and target frames. Thus, lower error indicates more precise temporal correspondence. We provide quantitative and qualitative results in Table 1 and Figure 3, respectively. More results can be found in Appendix Figure 8 and Figure 9.

GTCC provides substantially stronger cross-embodiment visual correspondence than generic visual features. Using visual features alone, GTCC + DTW consistently outperforms DINOv3 + DTW across both tasks and both splits. On *Object in Container*, GTCC substantially improves both overlap and progress error over DINOv3 across ID and OOD settings. The gap is bigger on *Cup on Saucer*, where DINOv3's overlap drops sharply under OOD scene changes, while GTCC remains robust at 71.7% OOD overlap. These results suggest that temporal cycle consistency learns representations that transfer across embodiment and scene changes more reliably than generic visual features.

Temporal warping improves alignment under execution-speed differences. On *Object in Container*, removing DTW substantially degrades performance, with OOD overlap dropping from 57.8% to 24.7% and progress error increasing from .098 to .231. DTW also provides smaller but consistent gains in the remaining settings, showing that non-uniform temporal warping generally helps align human and robot trajectories that execute the same behavior at different rates.

Action information helps only when the gap is small across embodiments. On *Object in Container*, adding pose features provides little benefit over GTCC alone, and pose-only baselines are substantially weaker. In contrast, on *Cup on Saucer*, absolute poses are highly informative, and combining them with GTCC yields the best performance. This suggests that pose cues are most useful when human and robot trajectories share a meaningful spatial correspondence, while GTCC remains effective regardless.

4.2 Policy Learning

To evaluate the effectiveness of behavior-aligned human-robot demonstrations and our cross-embodiment training objectives, we test policy on two real-world manipulation tasks. For *Object in Container*, which is cyclic, we use a fine-grained score to capture partial task progress. The policy continuously executes the four-skill sequence shown in Figure 3 for 60 seconds, receiving one point for each successfully completed skill. We evaluate four OOD settings, shown in Figure 4 (a), with 20 rollouts in total, randomizing the initial object positions in each rollout. Following EgoVerse (Punamiya et al., 2026), we report the score normalized by the average in-domain performance (e.g.,

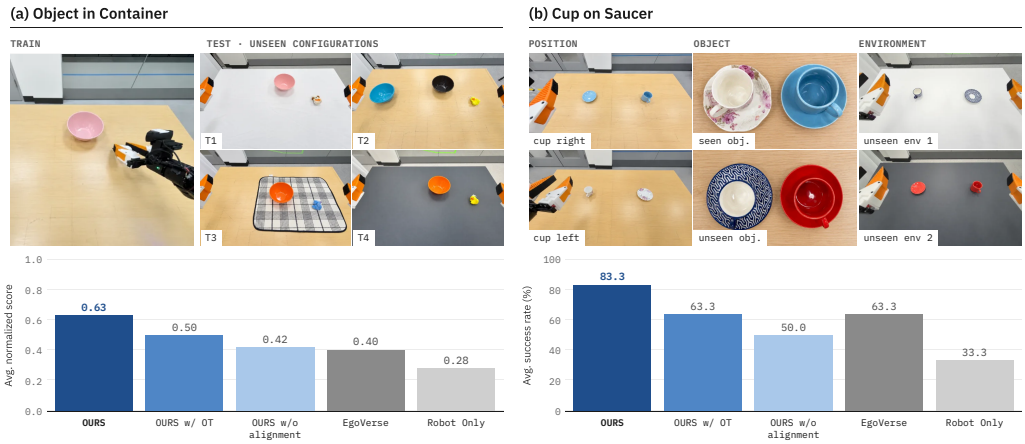


Figure 4: **Real-world rollout results.** We evaluate *Object in Container* across four unseen configurations including unseen background, target objects, and distractor objects, and *Cup on Saucer* across unseen objects and environments. Our method achieves the strongest overall performance.

55 points). For *Cup on Saucer*, we evaluate 30 rollouts across OOD scenarios with unseen objects and environments, as shown in Figure 4 (b), and report the number of successful trials. We compare against the following baselines and ablations:

OURS w/o alignment. We remove behavior alignment and instead randomly sample human and robot trajectories from their respective datasets.

OURS w/ OT. We replace the human-to-robot style transfer objective with an optimal transport (OT) loss applied to the input observation tokens, including image and state tokens, across the human and robot streams, following a formulation similar to EgoBridge (Punamiya et al., 2025).

EgoVerse (Punamiya et al., 2026). We remove the cross-embodiment style-transfer objective and simply co-train on human and robot data using the action and video prediction objectives.

Robot only. We use the pretrained robot policy trained solely on in-domain robot demonstrations, without finetuning with human data.

Our results in Figure 4 highlight the complementary benefits of human data finetuning, behavior alignment, and explicit cross-embodiment transfer objective. EgoVerse improves over Robot Only on both tasks, showing that co-training with human demonstrations is beneficial. However, OURS achieves the strongest performance on both tasks, indicating that explicitly coupling the human and robot streams through the style-transfer objective provides additional gains beyond co-training alone.

Replacing the style-transfer objective with an OT loss in OURS w/ OT also yields strong performance, outperforming EgoVerse on *Object in Container* and matching its performance on *Cup on Saucer*, while still falling short of OURS. This suggests that our style-transfer objective provides a more effective cross-embodiment learning signal than directly matching input representations with an OT loss. Meanwhile, removing behavior alignment in OURS w/o alignment degrades performance on both tasks. This underscores the importance of pairing human and robot trajectories based on behavioral correspondence when applying cross-embodiment supervision. Overall, **behavior alignment and the cross-embodiment style-transfer objective provide complementary gains for human-to-robot transfer.**

5 Conclusion

We present **BATCH**, a cross-embodiment learning framework that combines *behavior alignment* with a *cross-embodiment video-action model*. We use GTCC with DTW to temporally align human and robot behaviors in a learned embedding space, then train on the aligned demonstrations with joint co-training and human-to-robot style transfer. Across two real-world manipulation tasks, training with aligned data consistently outperforms unaligned training, while style transfer provides further gains beyond co-training alone. These results demonstrate the value of explicitly establishing and leveraging behavior correspondences for effective human-to-robot transfer.

6 AI USE STATEMENT

Claude Design Opus 5.5 was used to assist with layout design, visual element generation, and image editing (e.g., brightening) for Figure 2, Figure 3, Figure 4, and Figure 7. ChatGPT was used to identify relevant prior works and references and to refine wording and sentence structure. Claude Code Opus 5.5 was used to assist with data visualization and data processing scripts, navigating and understanding the DreamZero (Ye et al., 2026) codebase, and managing evaluation experiments reported in Table 1. All technical decisions, experiments, analyses, and final manuscript content were reviewed by the authors and validated.

7 ETHICS STATEMENT

This work studies cross-embodiment robot learning from human and robot demonstrations. The main ethical considerations concern human data privacy and robot safety. Human demonstrations, particularly egocentric recordings, may inadvertently capture bystanders or sensitive information, motivating appropriate consent, careful data handling, and review before data release. In addition, learned robot policies may behave unpredictably outside the training distribution, so real-world deployment should include appropriate human supervision. More broadly, while learning from human data may reduce the cost of robot data collection and enable more capable robotic systems, these capabilities could also be applied in settings that raise privacy, safety, labor, or surveillance concerns, making responsible data collection, evaluation, and deployment important considerations for future work.

8 REPRODUCIBILITY STATEMENT

We have made extensive efforts to ensure the reproducibility of our work. Details of the model architecture, training objectives, and hyperparameters are provided in Section 3 in the main paper and in Appendix Section B. The datasets used in the real-world experiments are described in Appendix Section A and Section B.1.1. Evaluation metrics and baseline implementations are specified in Section 4 in the main paper. We also provide an anonymous project website (`https://batch-learning.github.io/`) with qualitative results and visualizations, and we will release code, datasets, and model checkpoints upon acceptance to further support reproducibility.

REFERENCES

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Gerard Donahue and Ehsan Elhamifar. Learning to predict activity progress by self-supervised video alignment. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18667–18677, 2024. doi: 10.1109/CVPR52733.2024.01766.

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1801–1810. IEEE, 2019.

Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023.

Isma Hadji, Konstantinos G Derpanis, and Allan D Jepson. Representation learning via global temporal alignment and cycle-consistency. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11063–11072. IEEE, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Gabriel Ilharco, Mitchell Wortsman, Ross Wightman, Cade Gordon, Nicholas Carlini, Rohan Taheri, Achal Cohen, Ludwig Channel, and Et Al. Openclip, July 2021. URL `https://doi.org`. If you use this software, please cite it as below.

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

Hanjung Kim, Jaehyun Kang, Hyolim Kang, Meedeum Cho, Seon Joo Kim, and Youngwoon Lee. Uniskill: Imitating human videos via cross-embodiment skill representations. In Joseph Lim, Shuran Song, and Hae-Won Park (eds.), *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 4269–4294. PMLR, 27–30 Sep 2025.

Moo Jin Kim, Yihuai Gao, Tsung-Yi Lin, Yen-Chen Lin, Yunhao Ge, Grace Lam, Percy Liang, Shuran Song, Ming-Yu Liu, Chelsea Finn, and Jinwei Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *International Conference on Learning Representations (ICLR)*, 2026.

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Masquerade: Learning from in-the-wild human videos using data-editing. *arXiv preprint arXiv:2508.09976*, 2025a.

Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025b.

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

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Richard Li, Aditya Prakash, Andrew Wen, Saurabh Gupta, Yilun Du, and Pulkit Agrawal. What matters when cotraining robot manipulation policies on everyday human videos? In *RSS 2026 Workshop on Data-Centric Robotics: What Data Do Robots Really Need?*, 2026c.

Shuang Li, Yihuai Gao, Dorsa Sadigh, and Shuran Song. Unified video action model. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025. doi: 10.15607/RSS.2025.XXI.074.

Yangcen Liu, Woo Chul Shin, Yunhai Han, Zhenyang Chen, Harish Ravichandar, and Danfei Xu. Immimic: Cross-domain imitation from human videos via mapping and interpolation. *arXiv preprint arXiv:2509.10952*, 2025.

Ryan Punamiya, Dhruv Patel, Patcharapong Aphiwetsa, Pranav Kuppili, Lawrence Y Zhu, Simar Kareer, Judy Hoffman, and Danfei Xu. Egobridge: Domain adaptation for generalizable imitation from egocentric human data. In *Human to Robot: Workshop on Sensorizing, Modeling, and Learning from Humans*, 2025.

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

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

Yichao Shen, Fangyun Wei, Zhiying Du, Yaobo Liang, Yan Lu, Jiaolong Yang, Nanning Zheng, and Baining Guo. Videovla: Video generators can be generalizable robot manipulators. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

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

Tony Tao, Mohan Kumar Srirama, Jason Jingzhou Liu, Kenneth Shaw, and Deepak Pathak. Dexwild: Dexterous human interactions for in-the-wild robot policies. *arXiv preprint arXiv:2505.07813*, 2025.

Wan Team and Alibaba Group. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Ye Wang, Pei Lin, Xiong-Hui Chen, Haoqi Yuan, Zhixuan Liang, Yiyang Huang, Anzhe Chen, Zixing Lei, Jie Zhang, Tao Zhang, et al. Ego2robot: Scalable robot data synthesis from egocentric human data. *arXiv preprint arXiv:2608.02580*, 2026.

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. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5625–5632. IEEE, 2025.

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. In *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery. In *Conference on robot learning*, pp. 3536–3555. PMLR, 2023.

Mengda Xu, Zhenjia Xu, Yinghao Xu, Cheng Chi, Gordon Wetzstein, Manuela Veloso, and Shuran Song. Flow as the cross-domain manipulation interface. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2475–2499. PMLR, 06–09 Nov 2025.

Jonathan Yang, Chelsea Finn, and Dorsa Sadigh. Data analogies enable efficient cross-embodiment transfer. *arXiv preprint arXiv:2603.06450*, 2026.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

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

Kevin Zakka, Andy Zeng, Pete Florence, Jonathan Tompson, Jeannette Bohg, and Debidatta Dwibedi. Xirl: Cross-embodiment inverse reinforcement learning. In *Conference on Robot Learning*, pp. 537–546. PMLR, 2022.

Qihang Zhang, Lin Li, Luyao Zhang, Shuai Yang, Yiming Luo, Shuaiting Li, Ruilin Wang, Junke Wang, Jiahao Shao, Gangwei Xu, Jiaming Zhou, Yishu Shen, Yudong Jin, Fangyi Xu, Shuailei Ma, Jiaqi Liao, Guanxing Lu, Zifan Shi, Yongkun Wen, Yujie Zhao, Weixuan Tang, Xinyang Wang, Chaojian Li, Jiapeng Zhu, Ka Leong Cheng, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Native video-action pretraining for generalizable robot control. *arXiv preprint arXiv:2607.08639*, 2026.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified world models: Coupling video and action diffusion for pretraining on large robotic datasets. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

Lawrence Y Zhu, Pranav Kuppili, Ryan Punamiya, Patcharapong Aphiwetsa, Dhruv Patel, Simar Kareer, Sehoon Ha, and Danfei Xu. Emma: Scaling mobile manipulation via egocentric human data. *IEEE Robotics and Automation Letters*, 2026.

Appendix

A Data Collection Details

Human demonstrations are recorded using the main RGB camera of Aria Gen 1 glasses worn by the demonstrators. As shown in Figure 5, our robot platform consists of two ARX5 arms side-mounted on a custom 3D-printed shoulder structure, providing a human-like bimanual workspace. To obtain corresponding robot observations, we mount an Aria Gen 1 glass on the robot for the head-camera view and a Logitech RGB webcam on each wrist for close-up manipulation views.

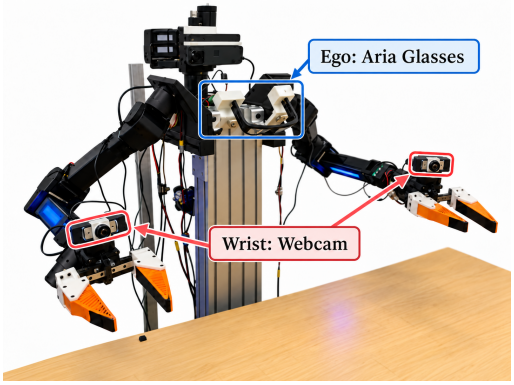


Figure 5: **Robot hardware setup.**

Human hand poses. Human hand poses are obtained using the Machine Perception Services (MPS) provided by Aria Gen 1. The hand poses are transformed into the coordinate frame of the tracked head camera. Each hand is represented by a 3D position and a quaternion orientation, resulting in a 7-dimensional representation per hand. Because no gripper-width equivalent is available in the human data, we zero-pad this dimension and exclude it from the human action loss during training.

Robot proprioception and actions. Robot actions are derived from proprioceptive measurements of both arms. For each arm, we use the 3D end-effector position, quaternion orientation, and gripper width, resulting in an 8-dimensional representation per arm and a 16-dimensional bimanual action representation. The end-effector poses are transformed into the robot head-camera coordinate frame using calibrated extrinsics.

B Method Details

We evaluate on two tasks: single-arm *Object in Container* and bimanual *Cup on Saucer*. The model, objectives, and training recipe are identical; what changes is the action dimension, the number of wrist camera inputs, and hence the width of the conditioning canvas and the token sequence length. The table below summarizes these differences.

	Object in Container	**Cup on Saucer**
Embodiment	single arm	bimanual
State / action dim	8 (pos 3, quat 4, gripper 1)	16 (2×8)
Human action dim	7 (one hand)	14 (two hands)
Wrist cameras	1	2
Conditioning canvas	240×560	240×800
Clean tokens / frame	525	750
Noisy tokens / frame	300	300
Robot CLIP tokens	514 (2×257)	771 (3×257)
Human CLIP tokens	257	257
Joint co-training pass	2,284 tokens	2,734 tokens
Style-transfer pass	3,334 tokens	4,234 tokens
Deployment	1,142 tokens	1,367 tokens
robot pretrain demos	90 (28,740 windows)	270 (25,480 windows)
human finetune demos	5 per scenario	8 per scenario
Robot demo duration	$\sim$12 s	$\sim$15 s
Human demo duration	$\sim$7 s	$\sim$4 s

Table 2: Per-task configuration.

B.1 Cross-embodiment Video Action Model

B.1.1 Data Processing

Temporal sampling. Recordings are at 30 fps. A training window takes 9 head-camera frames at a stride of 6 (5 fps, ≈1.6 s) and 16 actions at a stride of 3 (10 Hz), spanning the same interval.

Image geometry. Head-camera frames are resized to 240×320. Each wrist image (224×224) is zero-padded to 240×240 and concatenated with the head view along their latent width dimension. This produces an equivalent 240×560 canvas for *Object in Container*, which uses one wrist view, and a 240×800 canvas for *Cup on Saucer*, which uses two wrist views. Human streams contain only the head view, and the corresponding wrist regions are zero-padded. After $8\times$ spatial VAE compression and 2×2 patchification, a head-only noisy frame produces $15 \times 20 = 300$ tokens, while the clean conditioning frame produces 525 and 750 tokens for *Object in Container* and *Cup on Saucer*, respectively. Following Wan2.1-I2V (Team & Group, 2025), each video token contains 36 input channels: 16 noisy latent channels, a 4-channel mask indicating the known conditioning frame, and the 16-channel VAE latent of that frame.

States and actions. Robot states and actions use 8 dimensions per arm (position, orientation quaternion, gripper width): 8 dimensions for *Object in Container* and 16 for *Cup on Saucer*. Human hand poses use the same per-arm layout without the gripper dimension, which is zero-padded and masked out of the human action loss. Each arm's pose trajectory is expressed relative to that arm's pose at the first frame, $\tilde{T}_t = T_0^{-1} T_t$, while the gripper stays absolute. All dimensions are normalized with 1st/99th-percentile statistics computed per embodiment.

Augmentation. Training video receives random crops (scale 0.95), color jitter (brightness 0.3, contrast 0.4, saturation 0.5, hue 0.08), random grayscale ($p = 0.1$), and random posterization to 4 bits ($p = 0.1$). One set of parameters is sampled per clip and shared by all frames and by the head and wrist views. Deployment inputs are not augmented.

B.1.2 Attention Structure

Each embodiment is represented as a separate stream consisting of clean conditioning tokens, noisy future-video tokens, state tokens, and action tokens. Within each stream, we use block-causal self-attention over the conditioning context and the predicted block. Noisy video tokens attend to the stream's clean conditioning context and to the noisy video tokens within the same block. Action tokens additionally attend to the corresponding noisy video block, state token, and action tokens.

Positional information is also kept stream-specific: rotary position indices for both image tokens and register tokens (e.g., state, action tokens) restart from zero for each embodiment. Cross-attention to the CLIP image features of the conditioning frame is likewise separated by embodiment, such that each stream attends only to its own conditioning image features.

The two finetune objectives differ only in how clean-context attention is shared across the human and robot streams:

- **Joint co-training.** Human and robot streams are attention-isolated. Each stream attends only to its own clean conditioning context, with separate rotary positions and CLIP conditioning. As a result, the robot predictions are independent of the presence of the human stream, while both streams are optimized jointly within the same model.
- **Style transfer.** The human stream is placed before the robot stream, and the robot stream attends to the clean human video context together with its robot CLIP conditioning; the robot conditioning latent and registers are masked.

B.1.3 Flow Matching and Noise Sampling

Noise levels follow a flow-matching schedule with shift 5 over 1,000 discrete timesteps. Video timesteps are drawn as $t = \lfloor (1 - u) \cdot 1000 \rfloor$ with $u \sim \mathrm{Beta}(\alpha, 1)$, which biases training toward high noise, and are shared by the frames of a causal block. Action timesteps are drawn uniformly and independently for each action token. The conditioning latent frame is kept clean and excluded from

the video loss. Losses are weighted per timestep by the schedule's training weight. Video losses are computed on head-camera latents only.

In joint co-training, human video and action timesteps are drawn independently of the robot's, from the same distributions. In style transfer, all human latent frames sit at $t = 0$. Every robot latent frame, including the conditioning frame whose latent is withheld, is noised at a timestep drawn from the video prior, and the loss covers the two future frames. The registers of both streams are zeros at $t = 0$.

B.1.4 Temporal Pairing

Alignment. Human-robot pairs are computed once per task before training. Every frame of the robot and human recordings is embedded with the GTCC encoder, and each robot window is aligned to the human data by subsequence dynamic time warping with an ℓ_2 cost on those embeddings. For each window we store its k ($k = 5$ for *Object in Container* and $k = 1$ for *Cup on Saucer*) lowest-cost human candidates together with their warping paths per human demonstration, in raw frame units.

Sampling. At each training step a robot window draws one of its human candidates uniformly and reads the human frames, state, and actions at the indices of that candidate's warping path, which time-warps the human clip onto the robot window.

B.1.5 Robot Pretraining

Pretraining initializes from the first 8 blocks of Wan2.1-I2V-14B (Team & Group, 2025) and uses a single robot stream. We jointly train all 8 transformer blocks, the patch embedding, timestep and image-embedding projections, output head, and action/state encoders and decoder for 20,000 steps. Training uses 4 NVIDIA RTX PRO 6000 Blackwell Server Edition 96GB GPUs with a batch size of 16 per GPU, a learning rate of 1×10^{-4}, and video timesteps sampled from $\mathrm{Beta}(3, 1)$.

B.1.6 Cross-Embodiment Finetuning

The finetuning stage initializes from the pretrained checkpoint and runs for 1,000 steps at a learning rate of 2×10^{-5} with layer-wise decay 0.95: the last transformer block trains at the base rate and each earlier block at $0.95\times$ the rate of the block after it, while non-block parameters use the base rate. Video timesteps are drawn from $\mathrm{Beta}(1.5, 1)$. Each step optimizes exactly one of the two objectives, with a probability of 0.5 for each.

B.1.7 Inference

At deployment the sequence contains only the robot stream: the current head and wrist images are encoded once into a key-value cache as the clean conditioning frame, and the policy denoises one video block together with the 16 action tokens, conditioned on the current state.

We sample with a 16-step UniPC multistep flow solver (shift 5). The transformer is evaluated on 8 of the 16 steps under a fixed schedule and the remaining steps reuse the most recent velocity prediction. The action schedule runs to $\sigma = 0$, while the video schedule is rescaled to end at $\sigma = 0.25$. We execute the first 8 of the 16 predicted actions at 10 Hz, then reset the cache and replan from a new observation. The inference latency is 0.35s for single arm policy and 0.42s for bimanual policy on one GeForce RTX 5090 GPU.

C Additional Alignment Results

C.1 Evaluation Scenarios

We evaluate GTCC alignment using robot query trajectories and human demonstrations collected under both in-domain (ID) and out-of-domain (OOD) conditions. Figure 7 visualizes the evaluation scenarios for the two tasks. For *Object in Container*, the ID human demonstrations use the same table and objects as the robot demonstrations despite viewpoint differences, while the OOD sets introduce additional changes in object appearance, table color, and background. For *Cup on Saucer*, the ID demonstrations are collected on the same wooden table as the robot data using the in-domain

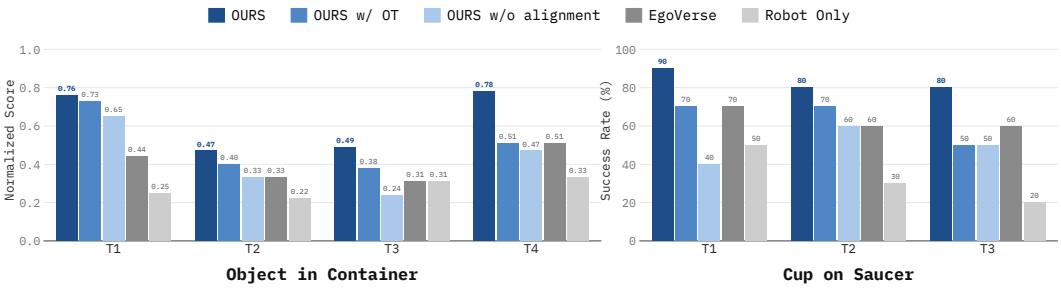


Figure 6: Policy performance per scenario.

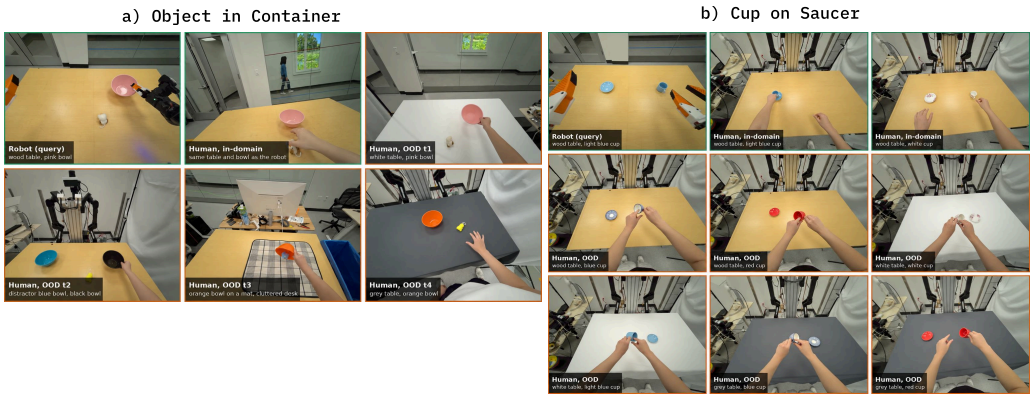


Figure 7: Alignment evaluation scenarios for *Object in Container* and *Cup on Saucer*.

light-blue and white cup-saucer sets. The OOD evaluation varies both object and scene appearance, including unseen cup-saucer sets (e.g., red and blue) and white or gray table cloths. For each scenario, we use 10 robot demonstrations and all available human demonstrations for each scenario to compute the alignment evaluation metrics.

C.2 Qualitative Alignment Results

Figures 8 and 9 show additional qualitative alignment examples for both tasks. GTCC consistently retrieves human frames that are more closely aligned in behavioral progress with the robot query.

D Additional Policy Results

D.1 Per-Scenario Performance

We conduct 5 one-minute rollouts per scenario for *Object in Container* and 10 rollouts per scenario for *Cup on Saucer*. For *Cup on Saucer*, T1 tests object generalization using only unseen object sets (red and blue), balanced across left- and right-side cup placements. T2 introduces a white tablecloth and mixes unseen object sets (e.g., red and blue) with seen object sets (e.g., light blue and white), while evenly varying the cup side across rollouts. T3 uses the same four object sets on a gray tablecloth, with 5 rollouts using seen objects and 5 using unseen objects, and an even split between left- and right-side cup placements.

Figure 6 breaks down policy performance across individual evaluation scenarios. Across all scenarios on both tasks, OURS achieves the strongest performance, with especially large gains over EgoVerse under more background distribution shifts (e.g., T1, T4 in *Object in Container* and T2, T3 in *Cup on Saucer*). The consistent drop for OURS w/o alignment further shows that behavior alignment is important for effectively transferring information from human demonstrations.



Figure 8: Additional qualitative alignment examples for *Object in Container*.

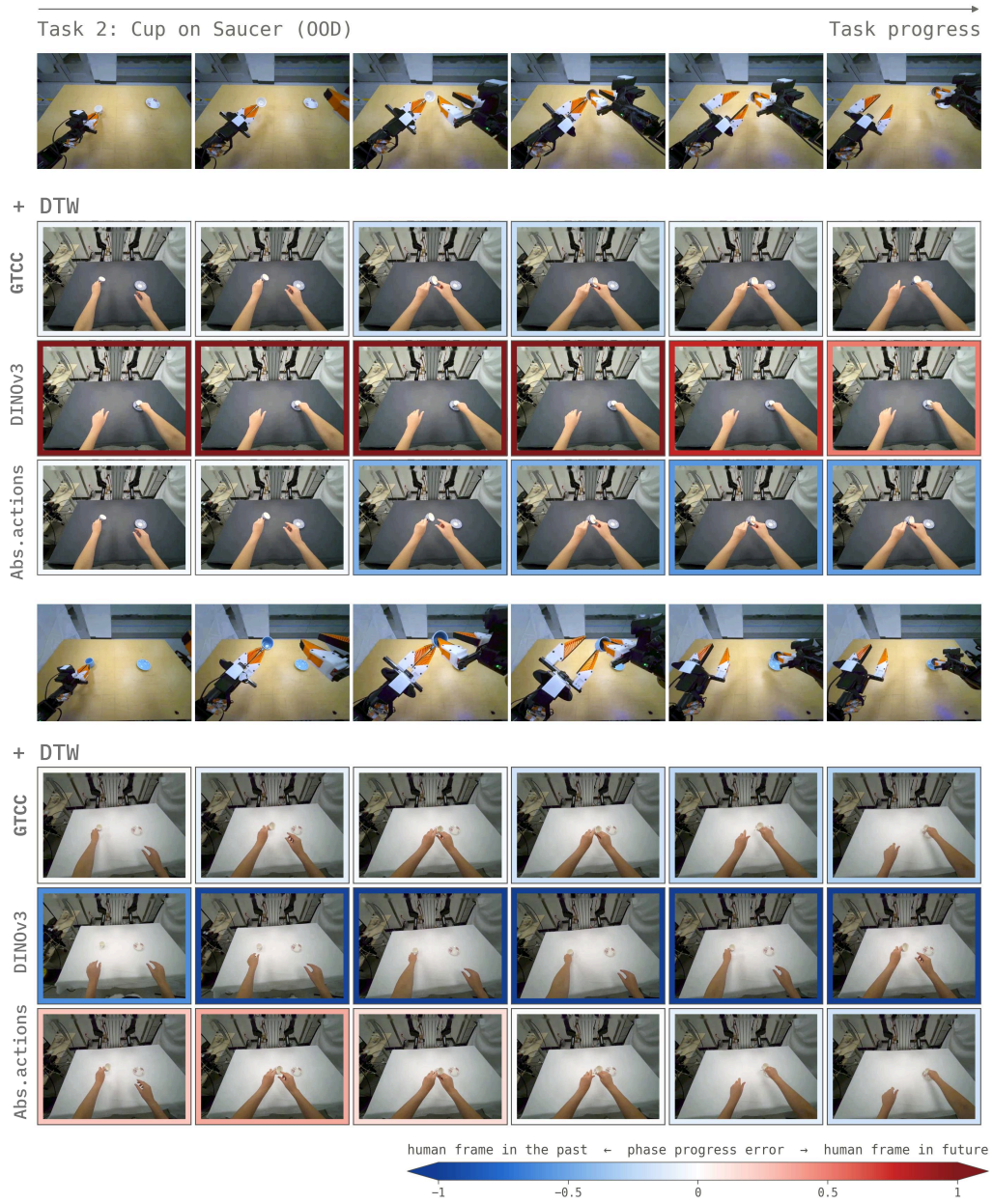


Figure 9: Additional qualitative alignment examples for *Cup on Saucer*.