LEARNING FROM HUMAN AND ROBOT REFERENCES FOR IN-CONTEXT CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting robot policies to new tasks through post-training incurs demonstration-collection and optimization costs. In-context learning (ICL) avoids test-time weight updates, but robot-reference methods still require demonstrations from the same embodiment. We ask whether human videos can provide references without robot teleoperation. Human-motion retargeting often relies on hand-designed kinematic correspondences, tying motion transfer to a robot-specific mapping. We introduce MIRAL (Mixed-embodiment In-context Reference Action Learning), a cross-embodiment ICL pretraining framework with a shared human–robot action interface. Disjoint action coordinates and modality-specific reference-action projectors support joint learning with human and robot references and queries. To discourage reference neglect, Context Error Margin (CEM) training fits query actions under matched references and raises mismatched prediction error. On The Imitator Game, we achieve 35% zero-shot success, 22 percentage points above the strongest listed comparator. Real-robot experiments on soybean pouring and tableware arrangement demonstrate effective human-reference-conditioned control.

1 INTRODUCTION

Adapting a pretrained robot policy to a new task often relies on post-training with task-specific robot demonstrations (Sridhar et al., 2025). This workflow incurs robot data-collection costs and a new optimization cycle for each target task. In-context learning (ICL) offers adaptation through reference demonstrations while keeping policy parameters fixed at test time (Fu et al., 2024; Sridhar et al., 2025). However, robot-reference ICL still depends on robot demonstrations for contextual adaptation (Fu et al., 2024; Sridhar et al., 2025). Human videos offer a way to specify behavior without robot teleoperation. This raises a question: *Can a policy learn jointly from human and robot demonstrations, then use human videos as references for robot control?*

Connecting human motion to robot control requires an interface that accounts for differences in morphology and actuation. Retargeting-based transfer uses prescribed wrist-to-end-effector or hand-keypoint correspondences, making the motion-to-control mapping depend on the target robot's kinematics (Qin et al., 2021; Yang et al., 2025; Shah et al., 2025). Video-conditioned prediction learns to use human visual references for robot control (Jain et al., 2024; Zhou et al., 2026a). Our focus is to retain explicit human motion in those references while learning the human–robot relationship through joint pretraining, with each modality keeping its own action coordinates.

We introduce MIRAL (Mixed-embodiment In-context Reference Action Learning), a cross-embodiment ICL pretraining framework built on the $\pi_{0.5}$ vision–language–action (VLA) model (Physical Intelligence et al., 2025). As Figure 1 illustrates, human motion and robot controls occupy disjoint coordinates in a shared action tensor. Modality-specific projectors encode reference actions for the vision–language backbone. The shared action expert predicts either modality, supporting all four human/robot reference–query directions. Joint training learns how each modality's reference actions inform the query actions. At deployment, the policy predicts robot controls directly in the robot coordinates under human or robot references.

Providing references alone does not ensure that a policy uses them. The query observation and instruction can already explain the training target, allowing action fitting to succeed with weak reference dependence. We therefore introduce the *Context Error Margin* (CEM) objective. For

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.

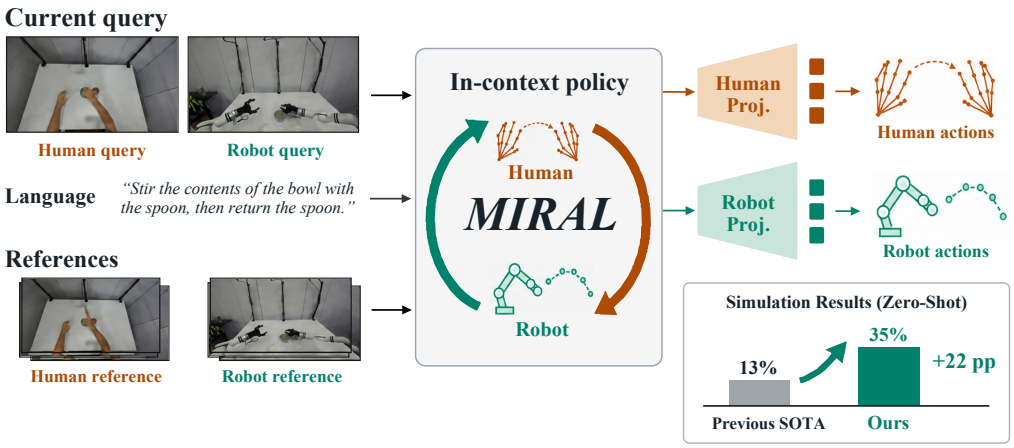


Figure 1: **Human and robot demonstrations guide a shared policy.** Reference images and actions condition MIRAL alongside the current query. Output "Proj." denotes fixed coordinate selection from the shared action tensor, without additional learned output heads. Photographs illustrate input roles; motion graphics are schematic.

the same query, CEM compares action prediction under matched and mismatched references with shared diffusion noise and time. It fits the matched target and raises mismatched error only until the latter reaches a margin relative to matched error.

The Imitator Game benchmarks robots' ability to imitate human demonstrations across changes in scene layout, objects, and affordances (Zhou et al., 2026b). On this benchmark, MIRAL achieves 84% seen-task and 35% zero-shot success, exceeding the strongest listed comparators by 3 and 22 percentage points, respectively. Ablations compare matched, absent, and mismatched references and isolate the effects of CEM, reference actions, and temporal selection. We also evaluate real-robot control on soybean pouring and tableware arrangement, including transfer to an unseen bowl in the tableware task.

We make three contributions. First, MIRAL jointly pretrains a policy on cross-embodiment demonstrations, allowing either modality to serve as reference or query while retaining distinct action semantics. Second, CEM couples matched and mismatched predictions for the same query, applying a bounded negative-error update until a relative margin holds. Third, MIRAL reaches 84% seen-task and 35% zero-shot success on The Imitator Game and demonstrates human-reference-conditioned control for soybean pouring and tableware arrangement.

2 Related Work

In-context learning. MOSAIC combines self-attention with temporal contrastive learning for one-shot visual imitation (Mandi et al., 2021), while Instant Policy models demonstrations, observations, and actions through graph diffusion (Vosylius & Johns, 2024). RoboPrompt converts object poses and keyframe actions into textual examples for an off-the-shelf language model (Yin et al., 2024). ICRT instead predicts sensorimotor sequences autoregressively (Fu et al., 2024), and RICL adds demonstration segments to a pretrained VLA's prefix (Sridhar et al., 2025). RoboTTT scales visuomotor context through recurrent fast weights that update during inference, supporting one-shot imitation from human videos (Jiang et al., 2026). MIRAL builds on demonstration conditioning with CEM: its contrastive signal acts on matched and mismatched action-prediction errors for the same query.

Learning from egocentric and human videos. Human videos support both pretraining and demonstration-conditioned control: R3M learns visual representations from egocentric video for downstream robot policies (Nair et al., 2022), while EgoMimic jointly trains on human hand trajectories and robot demonstrations (Kareer et al., 2024). EgoVLA predicts human wrist and hand actions before retargeting and robot fine-tuning (Yang et al., 2025); MimicDroid uses human play as in-

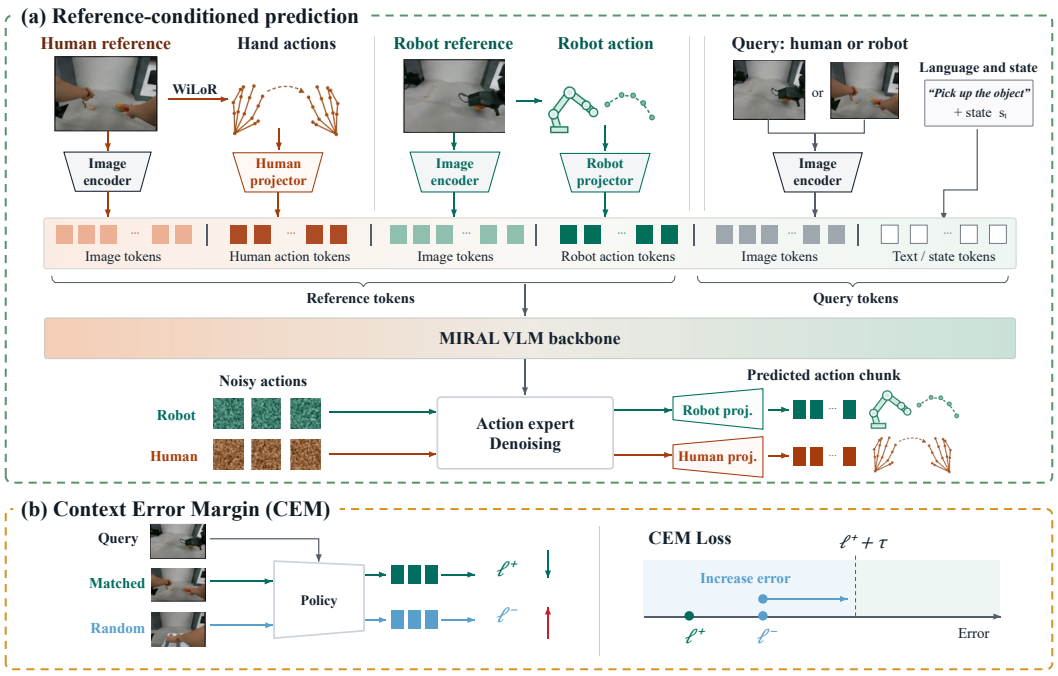


Figure 2: **Reference-conditioned prediction and CEM training.** (a) Human and robot reference actions enter the vision–language prefix through separate projectors. (b) Matched and mismatched branches share the query, parameters, noise, and time. Photographs illustrate dataset inputs; the diagram summarizes the prediction and training pathways.

context examples and retargets predicted motion to a humanoid (Shah et al., 2025). Vid2Robot conditions actions on demonstration videos through cross-attention and aligns human and robot video representations with auxiliary contrastive losses (Jain et al., 2024). Zero-WAM jointly predicts video and action chunks from human-video context (Zhou et al., 2026a), while WiLoR supplies explicit hand reconstruction from images (Potamias et al., 2025). MIRAL combines human–robot pretraining with continuous reconstructed-action references, predicting native actions for either modality through the shared action interface and training reference dependence with CEM.

Cross-embodiment learning. Open X-Embodiment pools robot datasets and demonstrates transfer between robot platforms through joint training (Open X-Embodiment Collaboration, 2023). VLA models combine this robot experience with pretrained visual and language representations. OpenVLA adapts a vision–language model to action prediction across robot embodiments (Kim et al., 2024), while π_0 couples a pretrained backbone with a flow-matching action expert (Black et al., 2024). $\pi_{0.5}$ learns from heterogeneous data to support generalization across robots and environments (Physical Intelligence et al., 2025). The Imitator Game extends evaluation to imitation from human demonstrations under controlled scene changes (Zhou et al., 2026b). We combine cross-embodiment pretraining with reference conditioning: human and robot actions share an interface with separate modality coordinates, and both modalities participate as references and queries during joint training.

3 Method

MIRAL predicts a query action chunk from demonstrations that may come from a different embodiment. We represent human motion and robot controls in one action tensor and project reference actions into the vision–language prefix of $\pi_{0.5}$. Task-level matching identifies eligible reference episodes; window selection supplies local visual and motion information. CEM trains the policy to fit the query with matched references and maintain a prediction-error margin against mismatched references.

3.1 REFERENCE-CONDITIONED ACTION PREDICTION

A query $q = (I, s, l, m)$ contains a main-view RGB image I, state s, instruction l, and modality $m \in \{r, h\}$ for robot or human. Given K reference windows $C = (c_1, \ldots, c_K)$, the policy predicts

$$\widehat{A} = \pi_\theta(q, C), \qquad A \in \mathbb{R}^{H \times D}, \qquad c_k = (I_k, s_k, l_k, A_k, m_k). \tag{1}$$

Here H is the action horizon and D is the unified action dimension. Each reference pairs its starting image with a known action chunk, providing both scene appearance and motion. The query contributes its observation, instruction, and available state to the prefix. During training, the action expert receives noise-corrupted query targets.

Robot controls and human motion descriptors occupy separate coordinates. For $a^r \in \mathbb{R}^{d_r}$ and $a^h \in \mathbb{R}^{d_h}$, let $D = d_r + d_h$ and define

$$E_r(a^r) = [a^r, 0_{d_h}], \qquad E_h(a^h) = [0_{d_r}, a^h]. \tag{2}$$

Robot coordinates retain the dataset's joint and gripper ordering; human coordinates describe camera-frame hand motion. The supplementary material gives their detailed ordering and descriptor definitions. This separation lets the policy learn from both modalities while retaining their native action meanings. We choose D to match the pretrained action expert's input and output width. The output projections in Figure 1 select the coordinates of the query modality. We keep these output projections fixed and learn the reference-action projectors. Reference modality determines how the policy encodes a demonstration; query modality determines which coordinates it predicts and supervises. These roles are independent. A human reference can therefore condition a robot target without treating hand descriptors as robot commands, and the same interface supports the reverse direction.

Modality-specific linear maps $P_r, P_h : \mathbb{R}^D \to \mathbb{R}^{d_{\text{vlm}}}$ convert reference actions into tokens of width d_{vlm} for the vision–language backbone. We concatenate the reference tokens before the query tokens. Each reference includes its image, instruction, available state, and action sequence, with the causal ordering used by RICL. The action expert attends to this prefix when denoising the query actions. We jointly train the reference projectors, backbone, and action expert.

Writing normalized actions as $\widetilde{a}$, each valid reference action $\widetilde{a}_{k,h}$ produces a token $z_{k,h} = P_{m_k}(\widetilde{a}_{k,h})$. If O_k denotes the visual, instruction, and available state tokens of reference k, its token sequence is

$$Z_k = [O_k; z_{k,1}; \ldots; z_{k,H}], \qquad Z_{\text{prefix}} = [Z_1; \ldots; Z_K; O_q], \tag{3}$$

where O_q contains the query inputs and semicolons denote sequence concatenation. The reference image identifies the scene in which its action chunk occurs. The action tokens retain the order of motion within that chunk. Coordinate masks determine which outputs receive supervision, whereas token masks determine which reference timesteps enter attention. This distinction prevents padded targets from contributing to the loss and invalid reference tokens from supplying context.

3.2 TASK MATCHING AND VISUAL REFERENCE RETRIEVAL

Reference selection first identifies demonstrations of the query task, then chooses a window within those demonstrations. We split episodes before constructing task-level correspondence tables, so training and held-out queries use separate tables. A matched reference shares the query's task but comes from another episode. During training, the K reference slots draw episodes independently and may select the same episode. Training supports human-to-robot (H2R), robot-to-human (R2H), robot-to-robot (R2R), and human-to-human (H2H) conditioning.

Two thirds of sampling draws use cross-modal references, equally divided between H2R and R2H. The remaining third permits all four directions under same-task matching, without balancing their frequencies. These sampling groups differ from the benchmark's scene-difficulty levels L0–L3.

At deployment, DINO image embeddings compare the current query observation with candidate window images from the task-matched episodes (Caron et al., 2021). Visual similarity selects the reference windows and their accompanying action chunks. This online selector uses the current image and the reference bank; it requires neither future query actions nor the duration of the ongoing rollout.

For offline training pairs, relative episode progress supplies a temporal alignment cue, detailed in the supplementary material. This distinguishes training-time alignment from online visual retrieval. Human sequences retain only frames with valid hand detections and their aligned actions; robot sequences retain their original frames.

A mismatched reference comes from a different task while preserving each positive slot's modality and real/simulation domain. We replace its image, instruction, state, and action chunk together and apply the same progress rule. A shared Bernoulli draw removes all task text from the query and both branches with probability 0.4. Otherwise, each reference retains its own instruction. Training always includes references.

3.3 Context Error Margin (CEM) Loss to Avoid Context Ignoring

CEM fits actions under matched references while penalizing mismatched references that predict the same target too well. We follow the action-flow convention of $\pi_{0.5}$ (Physical Intelligence et al., 2025). Draw Gaussian noise $\epsilon \sim \mathcal{N}(0, I)$ and time t:

$$X_t = t\epsilon + (1-t)\widetilde{A}, \qquad Y = \epsilon - \widetilde{A}. \tag{4}$$

For query i, both reference branches use identical X_t, t, Y and query augmentation. Their per-query velocity losses are

$$\ell_i^{\pm} = \frac{\sum_{h,d} M_{i,h,d}\big(v_\theta(X_t, t, q_i, C_i^{\pm})_{h,d} - Y_{i,h,d}\big)^2}{\max(1, \sum_{h,d} M_{i,h,d})}, \tag{5}$$

where M selects valid timesteps and active modality coordinates. Sharing the noisy target and diffusion time makes reference content the controlled difference between branches. Preserving reference modality and domain further prevents a negative example from differing solely through its action format or real/simulation origin.

The matched branch supplies the action-fitting loss. To penalize insufficient separation, define $p_i = \mathrm{sg}(\ell_i^+)$, where sg denotes stop-gradient:

$$\tau_i = \alpha p_i, \qquad s_i = \max(\alpha |p_i|, 10^{-6}), \tag{6}$$

$$w_i = \mathrm{clip}\left(\frac{p_i + \tau_i - \ell_i^-}{s_i}, 0, 1\right), \qquad \mathcal{L} = \frac{1}{B}\sum_i \left[\ell_i^+ + \lambda\, \mathrm{sg}(w_i)\left[p_i + \tau_i - \ell_i^-\right]_+\right]. \tag{7}$$

Here τ_i is the margin above the detached matched loss p_i. We use $\alpha = 0.10$ and $\lambda = 1$. Away from the denominator floor, $w_i = 1$ when $\ell_i^- \le p_i$; it decreases linearly to zero as ℓ_i^- reaches $(1+\alpha)p_i$. Once the negative loss meets the threshold, the penalty supplies no gradient.

The detached anchor and weight give

$$\nabla_\theta \mathcal{L} = \frac{1}{B}\sum_i \left[\nabla_\theta \ell_i^+ - \lambda\, \mathrm{sg}(w_i)\mathbf{1}\{\ell_i^- < p_i + \tau_i\}\nabla_\theta \ell_i^-\right] \tag{8}$$

away from nondifferentiable boundaries. The positive branch lowers prediction error, while an active negative branch raises its error with bounded coefficient λ. We sum both gradients at the same parameters, clip their global norm, and take one optimizer step. The negative branch has no action-fitting term. The objective can satisfy the margin by increasing only negative error. Evaluation therefore measures the benefit of matched references separately from the response to mismatched ones.

At inference, the action expert starts from Gaussian noise and integrates the learned velocity field toward the data endpoint, keeping the query and retrieved prefix fixed throughout denoising. We then invert the normalization and select the coordinates for the query modality. CEM supplies a training signal; deployment requires only the prediction branch conditioned on the retrieved references.

4 Experiments

We evaluate MIRAL's imitation performance and separate it from the contribution of reference conditioning through controlled comparisons and ablations.

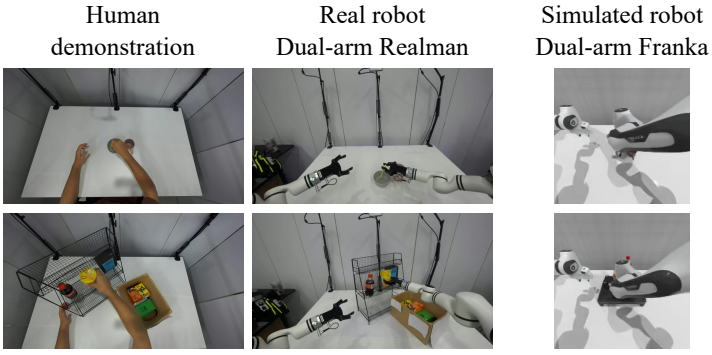


Figure 3: The Imitator Game spans human demonstrations, real robots, and simulated robots (Zhou et al., 2026b). Rows show same-task examples from IG-10K, selected independently rather than as synchronized frames. These are dataset demonstrations, not policy rollouts.

Table 1: Main simulation results in The Imitator Game (Zhou et al., 2026b). Seen denotes evaluation on training tasks; Zero-shot denotes evaluation on unseen tasks without additional parameter updates. Scratch denotes few-shot training from scratch; P+FT denotes pretraining followed by finetuning. Bold marks the highest value among the listed methods.

Method	Seen	Zero-shot	Scratch	P+FT
OpenVLA	0.29	0.06	0.14	0.20
RDT-1B	0.43	0.07	0.35	0.21
GR00T-N1.6	0.51	0.03	0.21	0.67
$\pi_{0.5}$	0.73	0.09	0.80	0.85
UniSkill	0.75	0.07	0.27	0.59
XSkill	0.79	0.10	0.35	0.73
DP/DINOv2	0.67	0.07	0.13	0.54
VQ-BeT/DINOv2	0.52	0.13	0.35	0.23
ACT/DINOv2	0.81	0.02	0.76	0.84
MIRAL	**0.84**	**0.35**	**0.81**	**0.91**

4.1 Training data and configuration

Figure 3 illustrates the human, real-robot, and simulated-robot domains of The Imitator Game. Our IG-10K training catalog below uses the human and simulated-robot domains. Section 4.3 describes our separately collected real-robot tasks.

The following describes the available training catalog and standard optimization configuration. The training catalog contains 9,994 simulated robot episodes and 4,053 real human episodes from IG-10K, including 800 additional human-level episodes. Robot query coverage spans 2,494 episodes at L0 and 2,500 at each of L1–L3. We use one short instruction per task. The simulator covers 50 tasks; the instruction catalog contains 53 entries. Training uses one main-view RGB image per observation at 224×224.

We initialize MIRAL from $\pi_{0.5}$ base and train all components for a budget of 100,000 optimizer steps. The global batch contains 128 queries on eight GPUs, each with matched and mismatched branches. The learning rate reaches 5×10^{-5} after 1,000 warmup steps and follows cosine decay. The supplementary material lists the remaining hyperparameters.

Robot records use 30 Hz and derived human records use 10 Hz before visibility filtering. A 64-step chunk spans different physical durations across modalities. The reference-start ablation tests sensitivity to relative episode progress.

4.2 Main benchmark comparison

The Imitator Game evaluates trajectory correspondence (L0), rearranged layouts (L1), within-category substitutions (L2), and affordance substitutions (L3) (Zhou et al., 2026b). Table 1 reports success rates for seen-task imitation, zero-shot transfer, few-shot training from scratch, and pretraining followed by finetuning.

Table 2: Real-robot evaluation. Task columns report successes out of 10 trials; Avg. is the mean standard-task success rate. OOD tests tableware arrangement with a bowl absent from training, which the robot must place on a plate.

Method	Soybean pouring	Tableware arrangement	Avg. (%)	OOD
π_0	2 / 10	6 / 10	40%	0 / 10
$\pi_{0.5}$	7 / 10	9 / 10	80%	1 / 10
MIRAL	9 / 10	10 / 10	95%	6 / 10

MIRAL exceeds the strongest listed baseline by 3 percentage points on seen tasks and 22 points in zero-shot transfer. Compared with $\pi_{0.5}$, few-shot success improves by 1 point from scratch and 6 points after pretraining. The largest advantage appears in zero-shot transfer; the smaller scratch difference suggests that the benefit depends on the learning regime. These aggregate results motivate controlled tests of reference use.

Published baseline recipes differ from ours. For example, the benchmark's $\pi_{0.5}$ uses a $\pi_{0.5}$-DROID initialization, frozen backbone, 50-step horizon, batch size 16, and a dense demonstration caption (Zhou et al., 2026b). The comparison therefore measures the full policy recipe. Table 3 compares CEM training with positive-only contextual training and varies reference content at inference.

4.3 Real-robot evaluation

We evaluate soybean pouring and tableware arrangement after local post-training. All policies use the same robot demonstrations from an ARX bimanual platform, three Intel RealSense cameras, state, instruction, and controller. Contextual policies receive two human reference windows from a disjoint bank. We partition demonstrations, references, and test initializations before adaptation. The supplementary material illustrates human–robot demonstration pairs.

Table 2 reports 10 trials per policy for each standard task and 10 additional trials for out-of-distribution (OOD) tableware arrangement. The OOD condition requires placing a bowl absent from the training set onto a plate. Avg. is the mean success rate across the two standard tasks. We record robot demonstrations, sample local post-training sequences, and run the deployed policy at 30 Hz. A soybean-pouring trial succeeds when the robot transfers at least one soybean from the source bowl into the target cup. A tableware-arrangement trial succeeds when the robot places the plate at the center, the cup at the upper right, and the spoon on the right. Each rollout has a 180 s timeout. Timeouts and safety interventions count as failed trials. Figure 4 shows the task setups and the start and end of an OOD policy execution. The supplementary material shows its full progression.

MIRAL improves average standard-task success by 15 percentage points over $\pi_{0.5}$, with gains on both soybean pouring and tableware arrangement. The larger difference occurs with the unseen bowl: OOD success improves by 50 percentage points. This condition retains the tableware task's goal while changing the manipulated object, testing whether the policy can apply the demonstrated placement behavior beyond the training objects. Together, the standard-task and OOD results support human-reference-conditioned control on hardware as well as in simulation.

At execution time, task-level matching restricts the reference bank, and DINO image similarity selects windows from the eligible episodes (Section 3.2). Contextual policies use the same bank and selection rule. The selector uses current observations, without access to future query actions or a known rollout duration.

4.4 Ablation study

The ablations separate the training objective from reference information. Table 3 crosses CEM training with three inference conditions. Table 4 tests four component changes. Both tables report Seen, Zero-shot, Scratch, and P+FT success rates, following Table 1.

CEM strengthens useful conditioning. With matched references, CEM training improves seen-task success by 10 percentage points and zero-shot success by 22 points over positive-only con-

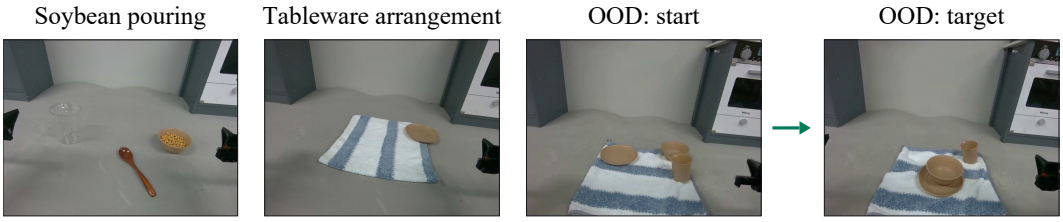


Figure 4: Real-robot task setups and OOD execution. The first two images show initial observations from robot demonstrations. The last two show the start and end of a policy rollout placing a bowl on a plate. This bowl is absent from the training set.

Table 3: CEM and inference-context ablation. Each training setting uses matched references; inference uses matched, absent, or random references. All entries are task success rates.

Method		Seen	Zero-shot	Scratch	P+FT
CEM	Inference context				
Enabled	**With context**	0.84	0.35	0.81	0.91
	Without context	0.70	0.08	0.76	0.88
	Random context	0.29	0.01	0.42	0.45
Disabled	With context	0.74	0.13	0.81	0.87
	Without context	0.74	0.10	0.79	0.85
	Random context	0.62	0.03	0.77	0.82

Table 4: Component ablations with matched references at inference. The last row is full MIRAL; preceding rows disable one component (✓: enabled, ×: disabled). Progress alignment refers to training reference starts. Entries are task success rates.

Language dropout	Progress alignment	Reference actions	Separate projectors	Seen	Zero-shot	Scratch	P+FT
×	✓	✓	✓	**0.85**	0.23	0.80	**0.91**
✓	×	✓	✓	0.81	0.30	0.79	0.90
✓	✓	×	✓	0.79	0.22	0.76	0.85
✓	✓	✓	×	0.83	0.34	**0.81**	**0.91**
✓	✓	✓	✓	0.84	**0.35**	**0.81**	**0.91**

textual training (Table 3). The two objectives tie in Scratch; CEM adds 4 points in P+FT. Within the CEM model, removing references reduces zero-shot success by 27 percentage points, compared with 3 points for positive-only training. Correct references therefore contribute to the gain, beyond the model's response to an incorrect reference.

Random references also cause larger losses in success after CEM training: seen-task success falls by 55 percentage points, compared with 12 points for positive-only training. The objective strengthens reference dependence in both directions. This makes reference selection consequential at deployment, and mismatch sensitivity alone would overstate the benefit.

Prediction-error diagnostics. On 16 held-out H2R query windows, we evaluate matched, absent, and mismatched references using mean normalized action L1 and sample standard deviation. Both training objectives use identical query windows, references, normalization, masks, and initial diffusion noise. We generate complete chunks with ten Euler steps and exponential moving average (EMA) parameters, then average valid-element errors within each window before computing statistics across windows. The supplementary material details the sampling protocol. "Random context" in the plots denotes the mismatched condition: a different-task reference that preserves modality and real/simulation domain. "With CEM" and "Without CEM" name the training objectives; both models use the same inference-time denoising procedure.

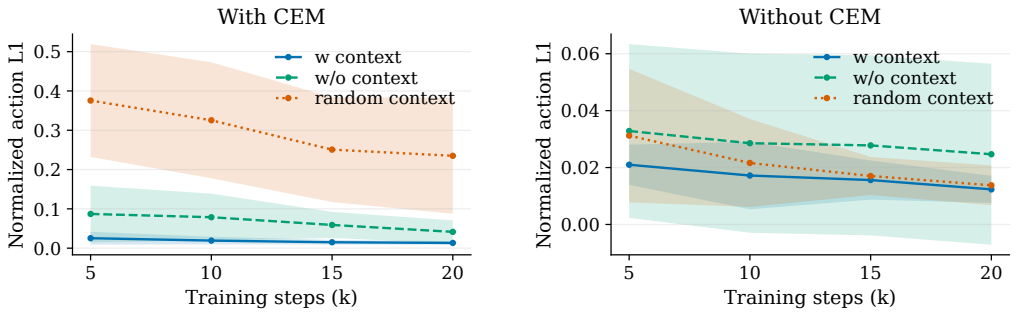


Figure 5: Simulation-query L1 with CEM (left) and without CEM (right), each with matched, absent, and mismatched references. Lines connect four evaluated checkpoints; shading shows mean $\pm$ one sample standard deviation across the same 16 fixed windows from six held-out episodes covering three tasks. The panels use different vertical scales.

Figure 5 shows clear separation between matched and mismatched references after CEM training. Without CEM, positive-only training produces similar errors for matched and mismatched references. At 20k steps, the matched and mismatched errors are 0.013 and 0.235 with CEM, compared with 0.012 and 0.014 without CEM. Removing references raises the error from 0.013 to 0.041 with CEM and from 0.012 to 0.025 without CEM. Both objectives use reference presence, while only CEM separates matched from mismatched content. These curves distinguish sensitivity to reference content from absolute prediction accuracy; the simulation tables evaluate closed-loop task success.

Motion and temporal alignment aid transfer. Removing reference actions reduces zero-shot success by 13 percentage points and P+FT success by 6 points (Table 4). Thus the motion channel contributes beyond the remaining reference inputs. Uniform training reference starts reduce success in all four regimes, with the largest drop of 5 points in zero-shot transfer. Training-time relative progress provides a useful temporal cue despite differences in execution speed.

Language dropout helps most in zero-shot transfer. Removing language dropout raises seen-task success by 1 percentage point but lowers zero-shot success by 12 points. Scratch changes by 1 point and P+FT remains unchanged. The result favors language dropout for transfer, without a uniform benefit across regimes. Sharing the reference-action projector changes success by at most 1 point and leaves Scratch and P+FT unchanged. These results provide limited evidence for an advantage from separate projectors, while reference motion and temporal selection account for larger changes.

5 Conclusion

We introduced MIRAL, a cross-embodiment in-context control framework trained on cross-modality demonstrations and conditioned on demonstrations from either modality. MIRAL combines a shared, modality-partitioned action tensor with reference-action projectors and CEM, which encourages reference-dependent prediction by enforcing a matched-versus-mismatched prediction-error margin without changing inference. This design enables a policy to learn relationships between human behavior and robot actions while preserving each modality's native action representation.

On The Imitator Game, MIRAL reaches 84% seen-task and 35% zero-shot success, with its largest gain in zero-shot transfer. Simulation analyses attribute these improvements to the combination of reference actions, CEM, and progress-aligned reference selection, demonstrating that the policy learns to use contextual demonstrations rather than relying only on query observations. Real-robot evaluations on soybean pouring and tableware arrangement, including transfer to an unseen bowl, further support human-reference-conditioned control.

MIRAL provides a step toward scalable robot learning from diverse human and robot experiences without task-specific post-training. Future deployment requires reliable reference retrieval, validation across broader task and embodiment distributions, and stronger safety mechanisms for long-horizon physical interaction. We hope this framework motivates further exploration of cross-embodiment in-context learning as a practical route toward more adaptive robotic systems.

AI USE STATEMENT

Generative AI tools assisted manuscript writing and editing, experimental planning, figure preparation, literature searches, and formatting checks. The authors reviewed all AI-assisted outputs and take responsibility for the final text, claims, citations, code, and figures. We did not use generative AI to generate experimental data or numerical results.

ETHICS STATEMENT

This work uses human demonstrations and physical robot evaluations. Human data use must follow applicable dataset licenses and consent conditions. Real-robot deployment carries physical risk; we count safety interventions as failed trials and confine deliberately mismatched-reference diagnostics to simulation and open-loop evaluation. These considerations motivate reliable reference retrieval and conservative safety procedures in deployment.

REPRODUCIBILITY STATEMENT

Section 3 specifies the unified action interface, reference construction and retrieval, and CEM objective. Section 4 reports the data composition, optimization settings, evaluation counts, and ablations. The supplementary material records the action coordinates, remaining hyperparameters, normalization, temporal alignment, and the prediction-error sampling protocol.

REFERENCES

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control, 2024. URL https://arxiv.org/abs/2410.24164.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision*, 2021. URL https://arxiv.org/abs/2104.14294.

Letian Fu, Huang Huang, Gaurav Datta, Lawrence Yunliang Chen, William Chung-Ho Panitch, Fangchen Liu, Hui Li, and Ken Goldberg. In-context imitation learning via next-token prediction, 2024. URL https://arxiv.org/abs/2408.15980.

Vidhi Jain, Maria Attarian, Nikhil J. Joshi, Ayzaan Wahid, Danny Driess, Quan Vuong, Pannag R. Sanketi, Pierre Sermanet, Stefan Welker, Christine Chan, Igor Gilitschenski, Yonatan Bisk, and Debidatta Dwibedi. Vid2Robot: End-to-end video-conditioned policy learning with cross-attention transformers, 2024. URL https://arxiv.org/abs/2403.12943.

Yunfan Jiang, Yevgen Chebotar, Ruijie Zheng, Fengyuan Hu, Yunhao Ge, Jimmy Wu, Tianyuan Dai, Scott Reed, Fei-Fei Li, Yuke Zhu, and Linxi Fan. RoboTTT: Context scaling for robot policies, 2026. URL https://arxiv.org/abs/2607.15275.

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

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model, 2024. URL https://arxiv.org/abs/2406.09246.

Zhao Mandi, Fangchen Liu, Kimin Lee, and Pieter Abbeel. Towards more generalizable one-shot visual imitation learning, 2021. URL https://arxiv.org/abs/2110.13423.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3M: A universal visual representation for robot manipulation, 2022. URL https://arxiv.org/abs/2203.12601.

Open X-Embodiment Collaboration. Open X-Embodiment: Robotic learning datasets and RT-X models, 2023. URL https://arxiv.org/abs/2310.08864.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, et al. $\pi_{0.5}$: A vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025. URL https://arxiv.org/abs/2504.16054.

Rolandos Alexandros Potamias, Jinglei Zhang, Jiankang Deng, and Stefanos Zafeiriou. WiLoR: End-to-end 3d hand localization and reconstruction in-the-wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. URL https://arxiv.org/abs/2409.12259.

Yuzhe Qin, Yueh-Hua Wu, Shaowei Liu, Hanwen Jiang, Ruihan Yang, Yang Fu, and Xiaolong Wang. DexMV: Imitation learning for dexterous manipulation from human videos, 2021. URL https://arxiv.org/abs/2108.05877.

Rutav Shah, Shuijing Liu, Qi Wang, Zhenyu Jiang, Sateesh Kumar, Mingyo Seo, Roberto Martín-Martín, and Yuke Zhu. MimicDroid: In-context learning for humanoid robot manipulation from human play videos, 2025. URL https://arxiv.org/abs/2509.09769.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. RICL: Adding in-context adaptability to pre-trained vision-language-action models. In *Conference on Robot Learning*, 2025. URL https://arxiv.org/abs/2508.02062.

Vitalis Vosylius and Edward Johns. Instant Policy: In-context imitation learning via graph diffusion, 2024. URL https://arxiv.org/abs/2411.12633.

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

Yida Yin, Zekai Wang, Yuvan Sharma, Dantong Niu, Trevor Darrell, and Roei Herzig. In-context learning enables robot action prediction in LLMs, 2024. URL https://arxiv.org/abs/2410.12782.

Jiaming Zhou, Qihang Zhang, Gangwei Xu, Cunxin Fan, Yujie Zhao, Ruilin Wang, Yiming Luo, Shuai Yang, Xing Zhu, Yujun Shen, Junwei Liang, and Yinghao Xu. Zero-WAM: In-context world-action modeling from human videos for open-ended task generalization, 2026a. URL https://arxiv.org/abs/2608.26103.

Xunzhe Zhou, Yiyang Cai, Fengyi Wang, Ran Ju, Hanxiang Ren, Ruizhe Liu, Yu Zhang, Qian Luo, Feng Chen, Pei Zhou, Yi Ma, and Yanchao Yang. The imitator game: Benchmarking robot imitative ability beyond action prediction. *arXiv preprint arXiv:2608.22301*, 2026b. URL https://arxiv.org/abs/2608.22301v1. Version 1.

6 APPENDIX

A IMPLEMENTATION DETAILS

Table 5: Training configuration. Batch size counts queries; each query contributes two reference branches.

Component	Setting
Initialization	$\pi_{0.5}$ base
Action chunk	64×32
Reference windows	2, independent episode draws
Reference-action projections	Human and robot, each $32 \to 2048$
Images	One main RGB view per observation, 224×224
Normalization	Fixed training q01/q99; no clipping
Supervision mask	Valid timesteps and active modality coordinates
Trainable components	Vision–language backbone, action expert, projectors
Global query batch / GPUs	128 / 8
Branch evaluations per update	128 matched + 128 mismatched
Training budget	100,000 updates
Peak / terminal learning rate	5×10^{-5} / 2.5×10^{-6}
Warmup / decay	1,000 steps / cosine
AdamW $\beta_1, \beta_2, \epsilon$	$0.9,\ 0.95,\ 10^{-8}$
Weight decay / gradient norm clip	10^{-10} / 1.0
Parameter EMA decay	0.99
CEM λ, α	$1.0,\ 0.10$
Language dropout	0.4
No-reference training probability	0.0
Diffusion time	$0.999\,\mathrm{Beta}(1.5, 1) + 0.001$
Inference	EMA parameters, 10 Euler steps

Offline temporal alignment. For training pairs, let n be the query start index and let T_q, T_c be the query and reference episode lengths. Relative progress gives

$$u = \frac{n}{\max(T_q - 1, 1)}, \qquad j = \mathrm{round}\big(u(T_c - 1)\big). \tag{9}$$

We take the reference window starting at j, computing progress after human-frame visibility filtering. This cue approximates action phase across episodes. At deployment, we instead use the image-based selector in the main paper.

Action coordinates. Unified indices 0–6 contain left robot joint actions, index 7 the left gripper, indices 8–14 the right joint actions, and index 15 the right gripper. These values follow the native joint-control dataset. Indices 16–31 contain left then right human descriptors. Each hand uses wrist position (3), palm rotation vector (3), openness (1), and pinch (1). The rotation vector differs from Euler angles. We derive each 8-D hand descriptor from WiLoR reconstructions; wrist positions and palm rotation vectors use the camera coordinate frame. If WiLoR misses one hand, we set its eight coordinates to zero; if it misses both hands, we discard the frame.

Token capacity. The text/state budget is 200 tokens per observation. The complete reference and query sequence has a configured capacity of 1,560 tokens. Each valid reference timestep contributes one continuous action token. We omit missing wrist views instead of allocating black-image tokens.

Normalization and randomness. Training statistics include the catalog's padding convention. Forward and inverse transforms share the same fixed assets; EMA applies only to policy parameters. The present sampler uses a NumPy generator without a single end-to-end seed for all workers and sampling operations. Consequently, the configuration alone cannot reproduce exact episode/window draws. The reproduction of sampling sequences requires explicit worker, augmentation, and diffusion seeding.

Gradient implementation. The numerical value of CEM does not fully specify its update. Stop-gradient on p prevents a second gradient through the positive loss, and stop-gradient on w gives the bounded negative coefficient in the CEM gradient described in the main paper. Removing either

operation changes the update. Even when the negative branch has zero weight, the model must evaluate it to determine whether it satisfies the margin.

Prediction-error evaluation. We compare reference conditions through open-loop action prediction. We generate complete chunks from Gaussian noise using ten Euler integration steps and an exponential moving average (EMA) of model parameters. Each query window uses the same initial noise with matched (C^+), absent ($\varnothing$), and mismatched (C^-) references. We measure normalized action L1. The with-versus-without comparison measures useful conditioning; the random-versus-with comparison measures mismatch sensitivity. Removing references from a model that always trains with them introduces a distribution shift. The success-rate ablation separately compares training with and without CEM.

We evaluate 16 distinct H2R query windows from six held-out robot episodes at L0 across three tasks for both training objectives, using identical query windows, references, normalization, masks, and initial noise. Each sample contributes its mean L1 over valid action elements; we report the mean and sample standard deviation of these 16 values. The windows share episodes, so the spread describes window-level variation. This subset supports seen-task, episode-held-out diagnostics. Scene levels L0–L3 describe reference correspondence; they differ from the four evaluation regimes in the main paper's benchmark table.

B Additional OOD policy visualization

Figure 6 shows a real-robot policy execution for OOD tableware arrangement with a bowl absent from training. The task is to place it on the plate. We sample 16 frames uniformly over the video, including its first and last frames, and retain only the main camera view.

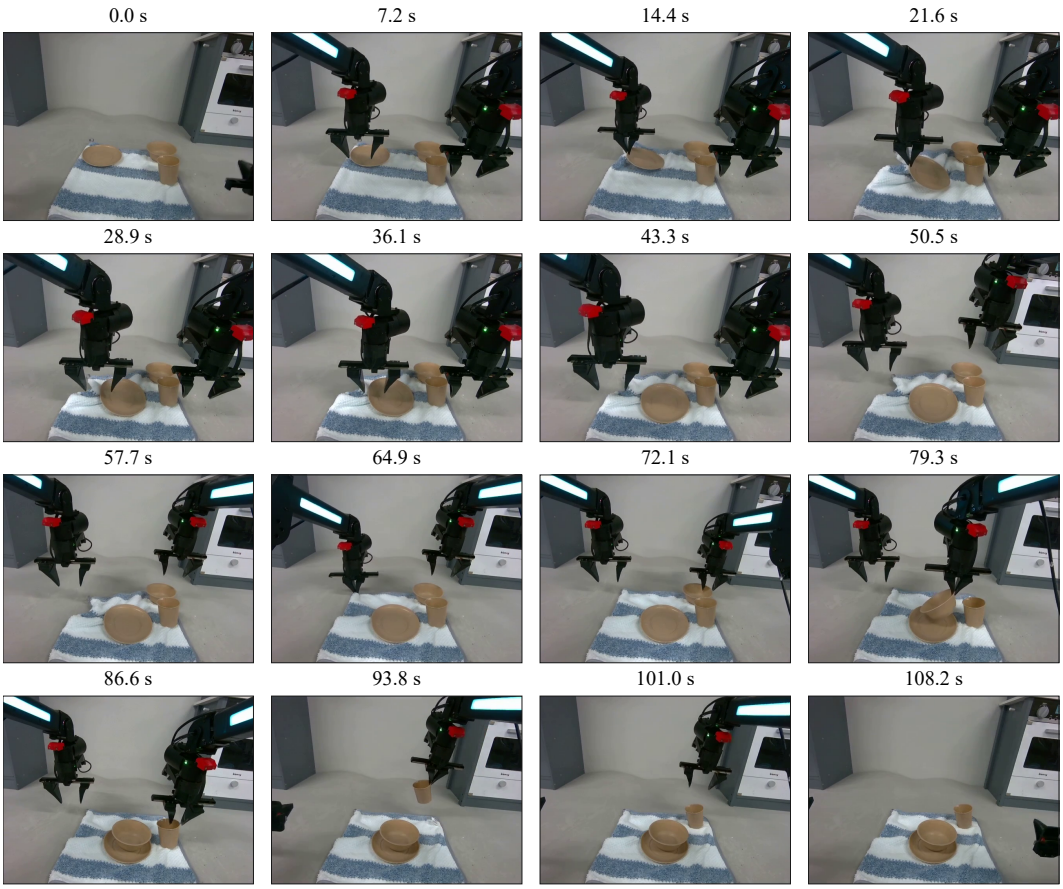


Figure 6: OOD tableware arrangement with an unseen bowl. Frames follow execution time from left to right and top to bottom; timestamps denote elapsed video time.

C HUMAN–ROBOT REFERENCE EXAMPLES

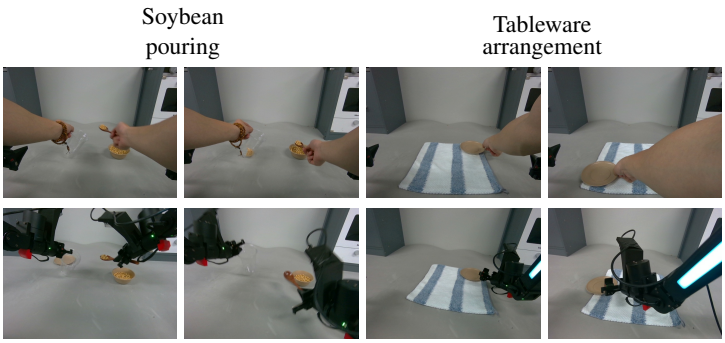


Figure 7: Human–robot demonstration pairs for the two real-robot tasks. Each column pairs a human episode with a robot episode; selected frames show similar object layouts and manipulation stages.