Retrieve, Don't Retrain: Extending Vision-Language-Action Models to New Tasks at Test Time

Anonymous Author(s)
Affiliation
Address
email

Abstract: Extending a vision-language-action (VLA) policy to a new task typically requires task-specific teleoperated demonstrations and per-task fine-tuning, making adaptation costly in both data collection and compute. In this paper, we show that this target-side per-task adaptation cost can be replaced by **retrieval**. Our retrieval augmented policy is trained once on paired demonstrations from the target embodiment (**query**) and a cheaper embodiment (**pool**, *e.g.*, human-hand video), then frozen. New tasks are added at deployment by appending pool-side demonstrations to a retrieval pool. The frozen policy conditions on retrieved trajectories at every control step, so new tasks are absorbed by indexing data rather than updating parameters. Fine-tuning is needed only to take on a new, unseen embodiment, not for each new task. We show that retrieval improves policies beyond a specific backbone, including standard VLA policies, but its effect is especially pronounced in Cosmos Policy, a video-generation-based world-action model (WAM). In this setting, retrieval supplies coarse task progression, while the WAM's future-image objective provides an additional visual consistency signal that strengthens the retrieval-conditioned actions. On PushT, we study how retrieval provides a reusable high-level motion prior for cross-embodiment generalization to unseen goal angles, while on RoboTwin 2.0 our method outperforms cross-embodiment baselines on unseen tasks, and we additionally demonstrate the method on a real robot.

Keywords: Robot foundation models, World-action models, Retrieval-augmented policies, Vision-language-action models

1 Introduction

General-purpose robot policies [1, 2, 3, 4, 5, 6, 7, 8] aim to execute open-ended manipulation behaviors from natural-language instructions while generalizing across diverse environments, tasks, and embodiments. Yet a new embodiment still requires its own teleoperated demonstrations and per-task fine-tuning, so cost grows with each new task added. We argue that this per-task cost is avoidable: behavioral knowledge from a cheap, data-rich source (*e.g.*, human-hand video demonstrations) can transfer to the target embodiment through a **retrieval rather than retraining** paradigm.

The cost of the previous approach is twofold. On the data side, target-embodiment demonstrations would be collected through teleoperation, which is slow, hardware-bound, and roughly $18\times$ slower to acquire than equivalent human-hand demonstrations [9, 10]. On the compute side, modern vision-language-action (VLA) models and robot foundation models operate over high-dimensional visual and action sequences, so per-task fine-tuning of recent world-action models (WAM) [7, 11, 8] costs roughly 24 GPU-hours per task and continues to scale with model size, context length, and action horizon. Both costs compound with every new task introduced.

Submitted to the 10th Conference on Robot Learning (CoRL 2026). Do not distribute.

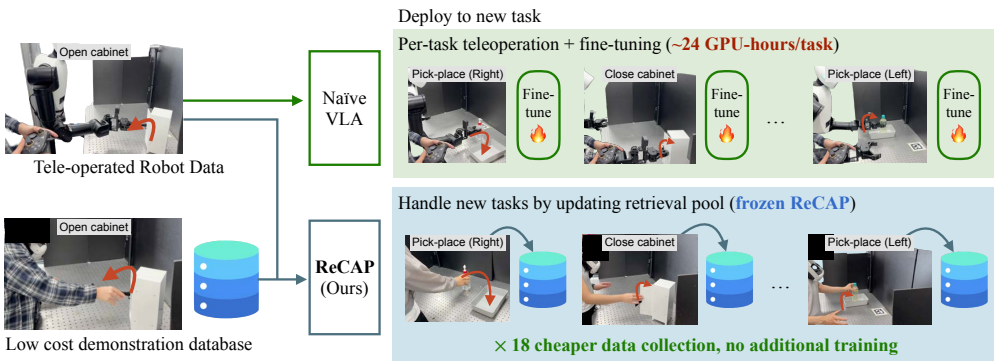


Figure 1: **RECAP overview.** Instead of teleoperating each new task and fine-tuning the policy (top, ∼24 GPU-hours/task for Cosmos Policy [7]), ReCAP appends cheap human-hand demonstrations to a retrieval pool while keeping the policy frozen (bottom), 18× cheaper [9, 10], no additional training.

We propose RECAP (**Re**trieval-**C**onditioned **A**ction **P**olicy), which shifts adaptation from repeated optimization to retrieval over a reusable pool of source-embodiment demonstrations. The policy is trained *once* to bridge the gap between source and target embodiments and is then frozen; behavioral coverage expands by *simply appending new demonstrations to the retrieval memory*.

RECAP builds on a world-action model (WAM) [7, 8, 12, 11, 13, 14, 15], specifically Cosmos Policy [7]. We parameterize the action latents as a *residual* over retrieved trajectories: retrieval supplies the coarse high-level motion and task progression, while the policy learns only the embodiment-specific dynamics needed to execute the behavior on the target robot. Crucially, the WAM's future-image prediction objective enforces consistency between the retrieved trajectory and the predicted evolution of the scene, a visual alignment signal that becomes informative *only* when paired with retrieval in unseen tasks, and that we find especially beneficial for long-horizon behaviors where high-level motion structure dominates.

The main contribution of this paper is threefold: a paradigm that adapts a policy to new tasks entirely at test time, absorbing each new task by extending a retrieval pool with cheap pool-embodiment demonstrations while the policy stays frozen with no parameter updates; a retrieval-conditioned residual policy on a WAM (i.e., Cosmos Policy [7]) in which retrieval supplies the high-level motion so the policy learns only the embodiment-specific correction, reinforced by the WAM's future-image objective that is informative only when paired with retrieval; and consistent gains over cross-embodiment baselines on PushT [16] (34.9% vs. 6.0% on seven unseen angles) and RoboTwin 2.0 [17] (31.5% vs. 26.0% on five unseen tasks), with a pool-progression study confirming monotonic coverage growth without parameter updates and a further real-robot validation.

2 Related Work

World action models. Recent VLA policies inherit either a pretrained language backbone with an added action head (OpenVLA [1], $\pi_{0.5}$ [3], GR00T N1.6 [4]) or a pretrained video model that folds actions into the same generative process (DreamZero [8], Cosmos Policy [7], mimic-video [12], Fast-WAM [11]). We call the latter family world-action models (WAMs); their video backbone is pretrained on internet-scale data and already encodes semantics and physical dynamics, so the policy learns only control on top, with action and future-observation prediction emerging from one shared video generation. These WAMs train on target-robot demonstrations alone, leaving their dynamics priors unpaired with cheaper cross-embodiment supervision, which we address by building a WAM on Cosmos Policy that conditions on retrieved pool-embodiment trajectories at training and deployment (Section 4.1).

Retrieval and Cross-embodiment policy transfer. Retrieval-based imitation [18] adapts a policy with relevant demonstrations rather than training from scratch on each task. Within a single em-

bodiment, FlowRetrieval [19] co-trains the policy on optical-flow-retrieved data, and STRAP [20] trains a task-specific specialist at deployment from DTW-matched sub-trajectories. To cut data cost, attention has turned to cheaper human demonstrations [21, 22, 23]: Hong et al. [24] uses a single human-hand demonstration to retrieve matching robot sub-trajectories and efficiently fine-tunes a policy on them for fast adaptation. These methods still pay for each new task with training, whether by co-training or a test-time fine-tune. Other methods reduce this training cost by using human data more directly. R+X [25] retrieves everyday human videos and executes them directly without training, but it stays in the human domain and does not learn an embodiment-specific correction. In-context methods such as MimicDroid [9] condition on a human prompt, which a user must hand-pick for each task. We instead learn the cross-embodiment gap once on paired (query, pool) data and freeze it; the user then grows the retrieval pool with new pool-embodiment demonstrations at test time, which broadens the policy's task coverage without retraining.

3 Problem Formulation

Setting. We study a cross-embodiment imitation setting with two sides: a *query* side that we want to control autonomously at deployment, carrying a *target embodiment* (e.g., a robot arm), and a *pool* side carrying a *pool embodiment* that is easier to collect demonstrations from but is never deployed (e.g., a human hand). The two embodiments differ in geometry, contacts, and dynamics, and differ sharply in data cost, since a query demonstration requires a teleoperation rig and operator while a pool demonstration only needs a lightweight tracker on a human. Transfer between them rests on two assumptions: a shared state/action representation (we use $\mathrm{SE}(3)$ end-effector pose plus a gripper signal where applicable), and motions that are semantically similar at the trajectory level, so a coarse plan derived from a pool trajectory is informative for the query.

Train and test access. At training time we assume that the model is given paired demonstrations $\mathcal{D}^{\text{query}}_{\text{train}}$ and $\mathcal{D}^{\text{pool}}_{\text{train}}$ on a fixed task distribution, each a set of state-action pairs $\{(s_t, a_t)\}$ collected on the corresponding embodiment. At test time the model faces *new* tasks outside this distribution; only pool demonstrations $\mathcal{D}^{\text{pool}}_{\text{test}}$ are provided, no additional query data is collected, and model parameters are not updated. The model is rolled out on the target embodiment, so the current query state s^{query}_t is observed at every control step.

Retrieval-conditioned action prediction. Let $\mathcal{D}^{\text{pool}}$ denote the active retrieval pool, $\mathcal{D}^{\text{pool}}_{\text{train}}$ at training and $\mathcal{D}^{\text{pool}}_{\text{test}}$ at deployment. At each step t, we select $t' = \arg\min_{t'} d(s^{\text{query}}_t, s^{\text{pool}}_{t'})$ over $\mathcal{D}^{\text{pool}}$, where d is a feature-space distance specified in Section 4.2. The retrieved chunk $(s^{\text{pool}}_{t':t'+H}, a^{\text{pool}}_{t':t'+H})$ of action chunk length H steps together with s^{query}_t is fed to the policy, which predicts the query action chunk $a^{\text{query}}_{t:t+H}$. The same rule applies at training and deployment with only $\mathcal{D}^{\text{pool}}$ changing; crucially, $\mathcal{D}^{\text{pool}}_{\text{test}}$ can be extended at any time without touching θ.

4 Proposed Method

We propose RECAP to adapt a policy to new tasks at test time without retraining, by retrieving relevant demonstrations rather than updating weights. The intuition is that the target and pool embodiments largely agree on *what* a task requires and differ mainly in *how* to execute it. A retrieved pool trajectory therefore supplies the shared high-level plan cheaply, leaving the policy to learn only the embodiment-specific correction, so adapting to a new task becomes a matter of indexing data rather than updating parameters, much as retrieval-augmented generation externalizes knowledge into a searchable store (Fig. 2). Section 4.1 details the backbone and its retrieval conditioning, and Section 4.2 specifies how the retrieved chunk is selected and extended at test time.

4.1 Retrieval-Augmented World Action Model

Because the policy stays frozen and conditions on an external pool, new tasks can be absorbed at test time by extending that pool rather than by retraining; we expect coverage to grow with the pool,

and cheap pool-embodiment data, such as human-hand video, to partly substitute for target-robot teleoperation.

Backbone and retrieval-conditioned input. The backbone is the Cosmos Policy formulation [7], which emits query-side actions and future image observations as one denoised video sequence.

We extend its conditioning with the retrieved pool-embodiment chunk:

$$\pi_\theta\Big(s_t^{\text{query}},\ \big(s^{\text{pool}}_{t':t'+H},\ a^{\text{pool}}_{t':t'+H}\big)\Big) \longmapsto \hat{a}^{\text{query}}_{t:t+H},\ \hat{s}^{\text{query}}_{t+H}. \quad (1)$$

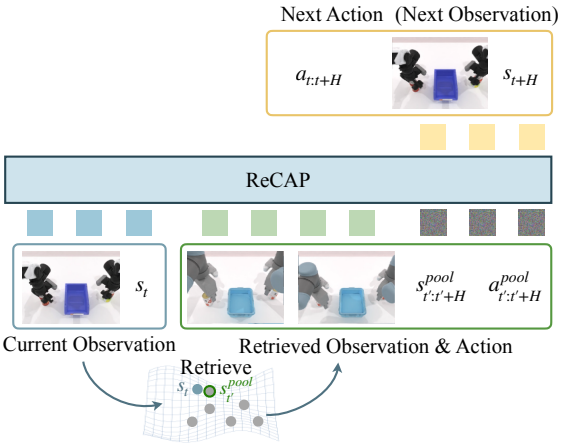


Figure 2: **RECAP framework.** The current observation retrieves a matching state-action chunk from the pool database; the retrieved chunk and the current observation then condition a world action model that denoises the next action and next observation in one video sequence.

The retrieved chunk and the observed query frame are encoded into clean latent frames and prepended along the temporal axis as conditioning, while the query-side future actions and observations are denoised from noise (Fig. 2); the language instruction enters via cross-attention. The retrieved chunk thus extends the standard I2V conditioning, a single clean frame, to a clean state-action subsequence, with no architectural modification.

Joint training objective (World action model). Action and future-image latents are supervised jointly with a single flow-matching loss:

$$\mathcal{L}(\theta) \;=\; \lambda\,\mathcal{L}_{\text{act}}\big(\hat{a}^{\text{query}}_{t:t+H},\, a^{\text{query}}_{t:t+H}\big) \;+\; \mathcal{L}_{\text{state}}\big(\hat{s}^{\text{query}}_{t+H},\, s^{\text{query}}_{t+H}\big) \quad (2)$$

Joint training yields actions aligned with the predicted next state, producing more grounded action outputs. For a standard VLA, we lack $\mathcal{L}_{\text{state}}$ and $\hat{s}^{\text{query}}_{t+H}$.

Residual action parameterization. Because the retrieved pool action chunk $a^{\text{pool}}_{t':t'+H}$ already encodes a coarse motion the target should execute, we let the action latents represent only the embodiment-specific correction [26, 27] on top:

$$\hat{a}^{\text{query}}_{t:t+H} \;=\; a^{\text{pool}}_{t':t'+H} \;+\; \Delta a_{t:t+H}. \quad (3)$$

This narrows what the action latents must encode to "how the query action differs from the pool's," the variation actually caused by the embodiment gap. This variation is weakly reflected in action labels but clearly visible in pixels, e.g., how contact happens, how the gripper closes. Residual focuses the action latents on this correction, and state prediction provides the dense visual signal to learn it.

4.2 Retrieval

At each control step t, the policy retrieves a pool-embodiment index t' from $\mathcal{D}^{\text{pool}}$ whose surrounding chunk best matches the current query context. We first form a candidate set $\mathcal{C}_t^{\text{traj}}$ by taking the top-K trajectories closest to the query under a composite initial-frame descriptor ψ_0, a language embedding of the goal, initial task-relevant object positions (via SAM 3 [28]), and initial proprioception. Within $\mathcal{C}_t^{\text{traj}}$, the index distance d in Section 3 is a weighted sum of L_2 distances over object pose, proprioception, and the upcoming action chunk (training only; dropped at inference), and a cosine distance over a DINOv3 [29] image feature.

At inference, new pool-embodiment demonstrations $\mathcal{D}^{\text{pool}}_{\text{test}}$ replace the active pool and are reindexed under ψ_0 and the features above; retrieval re-runs every step, so $\mathcal{D}^{\text{pool}}_{\text{test}}$ can grow within a session.

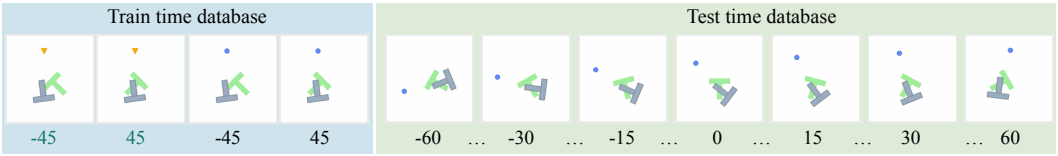


Figure 3: **PushT cross-embodiment pool database setting.** The training set pairs the *triangle* (target) and *disc* (pool) at $\pm 45^\circ$. The test set is a pool database of disc-pusher demonstrations spanning all goal angles, which the frozen triangle policy retrieves from on the seven unseen angles.

5 Experiments

We evaluate the proposed RECAP policy in three challenging cross-embodiment settings. PushT variant [16] (§5.2) provides a controlled environment for analyzing how retrieval improves generalization and why retrieval-conditioned WAMs are effective. RoboTwin 2.0 [17] (§5.3) evaluates whether the same paradigm scales to multi-task dual-arm manipulation, while real-robot experiments (§5.4) test whether unseen tasks can be absorbed through retrieval alone using human-hand demonstrations without additional robot training. Across all setups, we study the same hypothesis: retrieval supplies coarse task progression, while the policy learns only the embodiment-specific dynamics needed to execute the behavior on the target robot.

5.1 Experiment Setup

PushT Environment. In the 2D PushT benchmark [16, 30] an agent pushes a T-shaped block to a goal pose; we make it cross-embodiment with two pushers of different contact dynamics, a *triangle* (target) and a *disc* (pool), and take the goal rotation angle as the task axis. Training uses 100 paired $\{$*triangle*, *disc*$\}$ demonstrations at $\pm 45^\circ$; the triangle is then evaluated on nine different angles ranging from -60° to $+60^\circ$ in 15° steps, seven of them unseen. The test-time pool holds disc-pusher demonstrations at 5° resolution over $[-60^\circ, +60^\circ]$, added without retraining.

RoboTwin Simulation Environment. On RoboTwin 2.0 [17], we take Aloha-Agilex [31] as the target and UR5 as the pool, training on five paired $\{$target, pool$\}$ tasks and evaluating on five unseen ones (Table 1). A test-time pool-progression eval grows the pool through five levels (i.e., 11, 17, 23, 29, 35 tasks), each a strict superset of the previous, with the policy frozen throughout.

Real Robot Setup. On a physical robot, the pool embodiment is a human hand (video with wrist-pose tracked in VR) and the target is the teleoperated robot. We fine-tune on a single task, *open-cabinet*, with 25 paired demonstrations. Then we freeze the policy and evaluate on three tasks: the seen *open-cabinet* and two held out from fine-tuning, *place-bottle-in-plastic-box* and *close-cabinet* (Fig. 8a). The only test-time exposure to the held-out behaviors is 10 human-hand demonstrations per task added to the pool.

5.2 PushT Experiments

In this section, we study the following aspects of retrieval-conditioned generalization through PushT experiments [16]: (1) whether expanding the retrieval pool at test time improves unseen-task coverage without retraining, (2) whether retrieval benefits more from a WAM backbone than from an action-only policy, and (3) whether the retrieved trajectory acts as a reusable high-level motion prior that the policy adapts to the target embodiment. PushT is a controlled testbed whose generalization axis (i.e., the goal angle) is one-dimensional and densely measurable, which enables exploring these questions.

Test-time pool progression. Expanding the retrieval pool at test time without parameter updates steadily recovers the unseen angles. We grow the pool with disc-pusher demonstrations at intermediate goal angles and track per-angle success across five snapshots (Fig. 4). Specifically, the unseen-angle average rises monotonically from 6.0% without retrieval to 34.9% at the full pool.

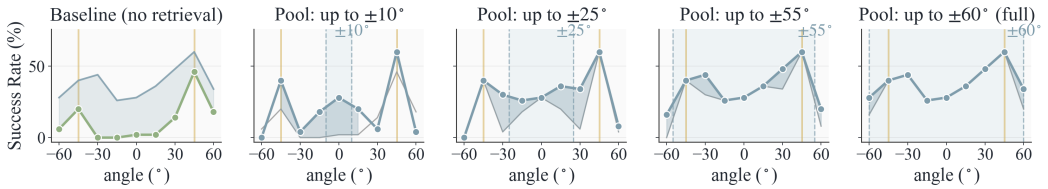
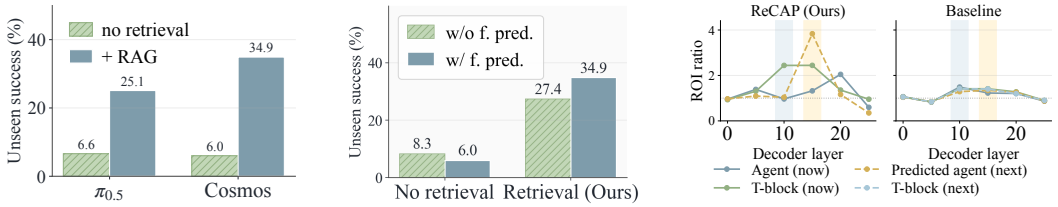


Figure 4: **Test-time pool progression on PushT.** The leftmost panel is the no-retrieval baseline with our full-pool curve overlaid (shaded gap). The other panels show per-angle success as the pool grows with no retraining, with the previous snapshot in gray and the incremental gain shaded.



(a) Backbone comparison. (b) Joint training (Cosmos). (c) ROI ratio across layers.

Figure 5: **Comparative analyses of RECAP and baseline on PushT.** (a) Unseen-angle success with and without retrieval on a $\pi_{0.5}$ and a Cosmos (WAM) backbone; retrieval helps both, and the WAM benefits more. (b) The future-image objective improves unseen success only when paired with retrieval. (c) Action-slot attention across decoder layers, which peaks on the T-block and then on the predicted next position under retrieval but stays near uniform without it.

Notably, each angle reaches much of its final success *before* its matching pool angle is added, so the policy interpolates over neighboring demonstrations rather than memorizing the nearest one.

Retrieval and the role of the WAM objective. If retrieval already supplies the coarse motion plan, then the remaining learning problem is primarily adapting that plan to the target embodiment dynamics. We therefore hypothesize that a WAM should benefit more from retrieval than an action-only policy, since its future-image objective encourages the retrieved trajectory to remain consistent with the predicted future scene, providing a stronger learning signal for the embodiment-specific dynamics adaptation. Figure 5a reveals that retrieval improves both backbones, raising the unseen-angle success of the action-only $\pi_{0.5}$ [3] from 6.6% to 25.1%. However, the gain is larger with a WAM backbone. In Fig. 5b, removing the image-prediction objective reduces our model to 27.4%, comparable to retrieval-augmented $\pi_{0.5}$ (25.1%), while restoring it improves performance to 34.9%.

How the retrieval-conditioned policy works. The retrieved chunk acts as a coarse motion prior, while the policy adapts it to the target embodiment rather than planning from scratch. Decoder cross-attention analysis in Fig. 5c reveals a two-stage routing behavior: early layers attend to the manipulated object and retrieved trajectory, whereas later layers shift attention toward the policy's own predicted next state, adapting the retrieved plan to the target embodiment dynamics. This structure does not emerge without retrieval, where the ROI ratio remains near 1.0 across layers, indicating near-uniform attention. Masking the action-to-retrieval attention further confirms that the retrieved trajectory causally influences the generated actions.

5.3 RoboTwin Simulation Experiments

In this section, we evaluate the proposed method in a multi-task, dual-arm manipulation simulation environment, RoboTwin 2.0 [17]. We compare against standard cross-embodiment baselines on seen and held-out unseen tasks, then test whether the test-time pool growth seen on PushT carries over to this multi-task regime.

Baseline comparisons. We compare against three cross-embodiment baselines that share our backbone but differ in how they incorporate the UR5 (pool) data. *Baseline*, Cosmos Policy [7] is trained

Table 1: **Quantitative analysis on RoboTwin.** We report per-task success rate (%) on RoboTwin, with Aloha-Agilex as the target embodiment and UR5 as the retrieval pool. The left block shows seen tasks, and the right block shows unseen tasks.

	Seen tasks						*Unseen tasks*					
Method	PCB	OM	DB	MP	GR	Avg ↑	MPP	PBS	CB	HM	LP	Avg ↑
Baseline [7]	47.5	10.0	25.0	30.0	50.0	32.5	0.0	0.0	20.0	0.0	0.0	4.0
Retrieval Only	30.0	7.5	22.5	7.5	60.0	25.5	0.0	10.0	42.5	37.5	40.0	26.0
Co-training	0.0	5.0	7.5	50.0	72.5	27.0	0.0	0.0	40.0	0.0	10.0	10.0
RECAP (Ours)	**60.0**	**12.5**	**40.0**	35.0	70.0	**43.5**	**5.0**	**12.5**	**47.5**	**47.5**	**45.0**	**31.5**

PCB = Place Cans Plasticbox OM = Open Microwave DB = Pick Dual Bottles MP = Move Can Pot GR = Grab Roller
MPP = Move Pillbottle Pad PBS = Place Bread Skillet CB = Click Bell HM = Hand-over Mic LP = Lift Pot

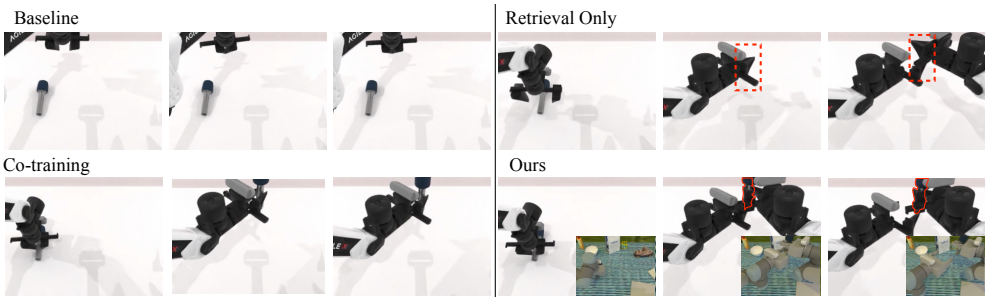


Figure 6: **Qualitative comparison on the held-out hand-over-mic task.** Baseline (top-left) and Co-training (bottom-left) fail to grasp the microphone; Retrieval Only (top-right) knocks it over (red box); Ours (bottom-right) grasps it successfully. Each inset shows the retrieved UR5 chunk that the policy conditions on.

on Aloha-Agilex (target) demonstrations alone, with no access to the pool. *Retrieval Only* executes the action sequence of the nearest pool demonstration without learning. *Co-training* is a common cross-embodiment baseline, also used in EgoBridge [21] and STRAP [20], that jointly trains a single policy on the union of target and pool trajectories. Table 1 shows that RECAP leads on both splits, at 43.5% seen and 31.5% unseen versus 32.5% and 26.0% for the strongest baseline. Replaying the nearest pool trajectory (i.e., Retrieval Only) is competitive only where that trajectory already approximates the target action. Otherwise, the learned residual on top of retrieval is what closes the gap. As Fig. 6 illustrates, the nearest UR5 trajectory collides and dislodges the object, while our policy produces the finer grip orientation the task needs.

Test-time pool progression. Figure 7 shows that growing the retrieval pool at test time with the policy frozen raises unseen-task success monotonically, from 9.0% to 31.5% at the full pool. Each increase coincides with a held-out task becoming retrievable, and once all five are in the pool, the frozen policy matches its supervised unseen-task average. Cheap pool-embodiment data at deployment can therefore stand in for new target-embodiment demonstrations on tasks unseen during fine-tuning.

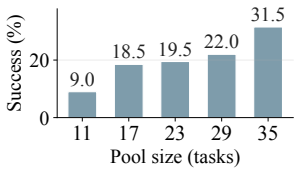
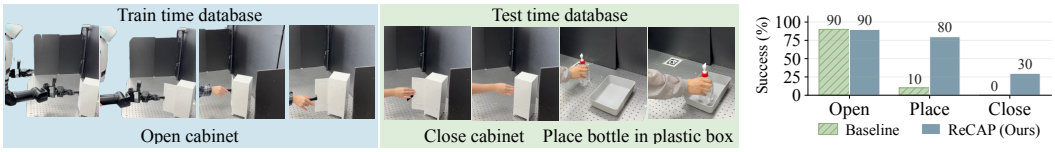


Figure 7: Test-time pool progression on RoboTwin.

5.4 Real Robot Experiments

We test whether the protocol transfers to real-world robots, despite the large embodiment gap between the human-hand demonstrations in the retrieval pool and the target robot (Fig. 8a). On two held-out tasks, the no-retrieval baseline collapses to the trained open-cabinet motion regardless of the target task, reaching only 10% and 0%, while retrieval enables the frozen policy to follow the conditioned human trajectory and reach 80% and 30% on closing the cabinet and placing the bottle (Fig. 8b). This indicates that human-hand demonstrations in the pool can partly substitute for additional target-robot teleoperation, even across a substantial embodiment gap. The qualitative results are shown in Figure 9.



(a) Query, Pool database setting. (b) Success rate.

Figure 8: **Real-robot experiment.** (a) The training-time database pairs the robot (query) with a human-hand pool; the test-time database adds human-hand demonstrations for the held-out tasks. (b) Per-task success rate over 10 rollouts, Baseline vs RECAP (Ours).

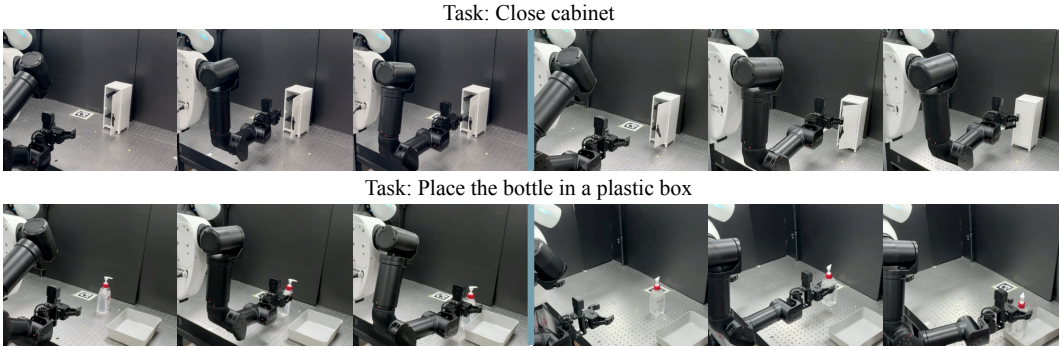


Figure 9: **Real-robot generalization to unseen tasks.** We show rollouts on the two held-out tasks (three frames each; baseline left, ours right). Trained only on *open-cabinet*, the baseline replays that trajectory and fails to close the cabinet (top) or grasp the bottle (bottom), whereas our policy follows the commanded behavior by conditioning on retrieved human-hand chunks.

6 Discussions

Summary. We extend a world-action-model policy to new tasks without retraining. Trained once to condition on a retrieval pool and predict an embodiment-specific residual on the retrieved trajectory, the frozen policy absorbs a new task by adding cheap pool-embodiment demonstrations at deployment. Across a cross-embodiment PushT variant, RoboTwin [17], and a physical robot, this improves generalization to unseen angles and tasks over cross-embodiment baselines, with success growing as the pool expands, and our analysis ties the gain to the WAM's future-image objective acting together with retrieval. Indexing cheaper pool-embodiment data at deployment can thus stand in for collecting new target-robot demonstrations.

Limitations and Future Work. Several constraints point to future work. The target and pool embodiments must share an end-effector action space, since the residual refines a retrieved chunk in a common low-level representation; structurally different action spaces (e.g., a dexterous hand versus a parallel gripper) would require an embodiment-agnostic interface or learned action translator. The pool must also contain trajectories rather than video alone, so video-only sources such as raw human or web video would first need to be lifted into a state-action representation. What constitutes an effective representation for cross-embodiment retrieval also remains an open question; our current descriptor combines language, object pose, proprioception, and visual features, but more scalable or embodiment-invariant representations may further improve transfer.

In addition, the residual formulation becomes less reliable when retrieved motions differ substantially in execution speed or temporal scale, particularly for larger chunks where errors can accumulate over time. Developing retrieval and adaptation mechanisms that remain robust under significant temporal or dynamical mismatch is an important direction for future work. Finally, scaling retrieval beyond curated trajectory pools to in-the-wild video sources such as raw YouTube video remains a promising step toward broadly reusable robot experience.

References

[1] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. OpenVLA: An open-source vision-language-action model. In *Proc. of the 8th Annual Conference on Robot Learning (CoRL)*, 2024. URL https://openreview.net/forum?id=ZMnD6QZAE6.

[2] J. Lee, J. Duan, H. Fang, Y. Deng, S. Liu, B. Li, B. Fang, J. Zhang, Y. R. Wang, S. Lee, et al. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.

[3] Physical Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

[4] J. Bjorck et al. Gr00t n: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[5] J. Zheng, J. Li, Z. Wang, D. Liu, X. Kang, Y. Feng, Y. Zheng, J. Zou, Y. Chen, J. Zeng, T. Wang, Y.-Q. Zhang, J. Liu, and X. Zhan. X-VLA: Soft-prompted transformer as scalable cross-embodiment vision-language-action model. In *Proc. of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=kt51kZH4aG.

[6] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu. RDT-1b: a diffusion foundation model for bimanual manipulation. In *Proc. of the Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=yAzN4tz7oI.

[7] M. J. Kim, Y. Gao, T.-Y. Lin, Y.-C. Lin, Y. Ge, G. Lam, P. Liang, S. Song, M.-Y. Liu, C. Finn, and J. Gu. Cosmos policy: Fine-tuning video models for visuomotor control and planning. In *Proc. of the Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=wPEIStHxYH.

[8] S. Ye, Y. Ge, K. Zheng, S. Gao, S. Yu, G. Kurian, S. Indupuru, Y. L. Tan, C. Zhu, J. Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

[9] R. Shah, S. Liu, Q. Wang, Z. Jiang, S. Kumar, M. Seo, R. Martín-Martín, and Y. Zhu. Mimicdroid: In-context learning for humanoid robot manipulation from human play videos. *arXiv preprint arXiv:2509.09769*, 2025.

[10] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar. Mimicplay: Long-horizon imitation learning by watching human play. In *Proc. of the 7th Annual Conference on Robot Learning (CoRL)*, 2023. URL https://openreview.net/forum?id=hRZ1YjDZmTo.

[11] T. Yuan, Z. Dong, Y. Liu, and H. Zhao. Fast-wam: Do world action models need test-time future imagination? *arXiv preprint arXiv:2603.16666*, 2026. URL https://arxiv.org/abs/2603.16666.

[12] J. Pai, L. Achenbach, V. Montesinos, B. Forrai, O. Mees, and E. Nava. mimic-video: Video-action models for generalizable robot control beyond vlas. *arXiv preprint arXiv:2512.15692*, 2025.

[13] L. Li, Q. Zhang, Y. Luo, S. Yang, R. Wang, F. Han, M. Yu, Z. Gao, N. Xue, X. Zhu, et al. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

[14] S. Wang, J. Shi, Z. Fu, X. He, F. Liu, C. Yang, Y. Zhou, Z. Fei, J. Gong, J. Fu, et al. World action models: The next frontier in embodied ai. *arXiv preprint arXiv:2605.12090*, 2026.

[15] S. Li, Y. Gao, D. Sadigh, and S. Song. Unified video action model. In *Proc. of the Robotics: Science and Systems (RSS), 2025*, 2025.

[16] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

[17] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Z. Li, Q. Liang, X. Lin, Y. Ge, Z. Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025.

[18] M. Du, S. Nair, D. Sadigh, and C. Finn. Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets. In *Proc. of the Robotics: Science and Systems (RSS), 2024*, 2024.

[19] L.-H. Lin, Y. Cui, A. Xie, T. Hua, and D. Sadigh. Flowretrieval: Flow-guided data retrieval for few-shot imitation learning. In *Proc. of the 8th Annual Conference on Robot Learning (CoRL)*, 2024. URL https://openreview.net/forum?id=FHnVRmeqxf.

[20] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis. STRAP: Robot sub-trajectory retrieval for augmented policy learning. In *Proc. of the Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=4VHiptx7xe.

[21] R. Punamiya, D. Patel, P. Aphiwetsa, P. Kuppili, L. Y. Zhu, S. Kareer, J. Hoffman, and D. Xu. Egobridge: Domain adaptation for generalizable imitation from egocentric human data. In *Proc. of the Thirty-ninth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2026. URL https://openreview.net/forum?id=FGMBxzpgis.

[22] V. Jain, M. Attarian, N. J. Joshi, A. Wahid, D. Driess, Q. Vuong, P. R. Sanketi, P. Sermanet, S. Welker, C. Chan, I. Gilitschenski, Y. Bisk, and D. Dwibedi. Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers. In *Proc. of the Robotics: Science and Systems (RSS), 2024*, 2024.

[23] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. In *Proc. of the Robotics: Science and Systems (RSS), 2024*, 2024.

[24] M. Hong, A. Liang, K. Kim, H. Rajaprakash, J. Thomason, E. Bıyık, and J. Zhang. Hand me the data: Fast robot adaptation via hand path retrieval. *arXiv preprint arXiv:2505.20455*, 2025.

[25] G. Papagiannis, N. Di Palo, P. Vitiello, and E. Johns. R+ x: Retrieval and execution from everyday human videos. In *Proc. of the 2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8284–8290. IEEE, 2025.

[26] S. Sha, Y. Wang, B. Huang, A. Loquercio, and Y. Li. Efficient and reliable teleoperation through real-to-sim-to-real shared autonomy. *arXiv preprint arXiv:2603.17016*, 2026.

[27] C. Schaff and M. R. Walter. Residual policy learning for shared autonomy. In *Proc. of the Robotics: Science and Systems (RSS), 2020*, 2020.

[28] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, J. Lei, T. Ma, B. Guo, A. Kalla, M. Marks, J. Greer, M. Wang, P. Sun, R. Rädle, T. Afouras, E. Mavroudi, K. Xu, T.-H. Wu, Y. Zhou, L. Momeni, R. Hazra, S. Ding, S. Vaze, F. Porcher, F. Li, S. Li, A. Kamath, H. K. Cheng, P. Dollár, N. Ravi, K. Saenko, P. Zhang, and C. Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL https://arxiv.org/abs/2511.16719.

[29] O. Siméoni, H. V. Vo, M. Seitzer, F. Baldassarre, M. Oquab, C. Jose, V. Khalidov, M. Szafraniec, S. Yi, M. Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

[30] R. Cadene, S. Alibert, A. Soare, Q. Gallouedec, A. Zouitine, S. Palma, P. Kooijmans, M. Aractingi, M. Shukor, D. Aubakirova, M. Russi, F. Capuano, C. Pascal, J. Choghari, J. Moss, and T. Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

[31] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.