2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

HAND Me the Data: Fast Robot Adaptation via Hand Path Retrieval

Matthew M. Hong*, Anthony Liang*, Kevin Kim, Harshitha Rajaprakash,
Jesse Thomason†, Erdem Bıyık†, Jesse Zhang†
Thomas Lord Department of Computer Science,
University of Southern California

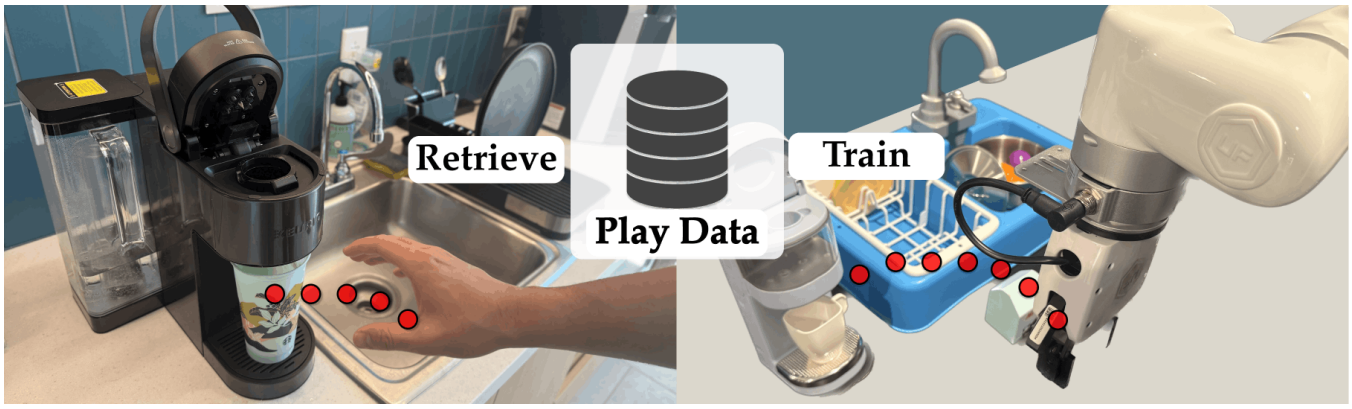

Fig. 1: HAND learns a policy from as little as 1 human hand demonstration and 4 minutes of real-world time by *retrieving* from robot play data.

***Abstract*— We present HAND, a *simple* and *time-efficient* method for teaching robots new manipulation tasks through human hand demonstrations. Instead of relying on task-specific robot demonstrations collected via teleoperation, HAND uses easy-to-provide hand demonstrations to retrieve relevant behaviors from task-agnostic robot play data. Using a visual tracking pipeline, HAND extracts the motion of the human hand from the hand demonstration and retrieves robot sub-trajectories in two stages: first filtering by visual similarity, then retrieving trajectories with similar behaviors to the hand. Fine-tuning a policy on the retrieved data enables *real-time learning of tasks* in under four minutes, without requiring calibrated cameras or detailed hand pose estimation. Experiments also show that HAND outperforms retrieval baselines by over $2\times$ in average task success rates on real robots. Videos can be found at our project website: https://liralab.usc.edu/handretrieval/.**

I. INTRODUCTION

For robots to operate seamlessly in human-centric settings, they should be able to *rapidly* learn new tasks with *minimal human supervision*. This requires learning algorithms that (1) scale across many tasks and (2) adapt quickly to new ones.

Imitation learning has produced capable multi-task robot policies [1, 2, 3, 4, 5], but scaling is hindered by its reliance on vast amounts of expert, task-specific teleoperation data [6]. In contrast, *task-agnostic play data* is far easier to collect, without requiring constant environment resets or task-specific labeling [7, 8, 9]. The challenge is in making such unstructured data usable for teaching robots new tasks quickly.

*Equal Contribution, † Equal Advising

Therefore, we propose HAND, a *simple and time-efficient* approach to adapt pre-trained play policies to specific tasks using a single human hand demonstration (see Figure 1). Unlike prior retrieval methods [10, 11, 12, 13, 14, 15] that require robot demonstrations of the target task, HAND extracts 2D relative hand motion paths from the provided human hand demonstration to inform retrieval, enabling even non-experts to teach robots without teleoperation.

HAND enables both *scalability* and *speed*. Towards *scalability*, HAND avoids the need for calibrated depth cameras [16, 17], specialized eye-in-hand setups [18], or detailed hand-pose estimation [18, 19]. Instead, it labels a robot play dataset with 2D gripper positions relative to the RGB camera frame, tracked using a visual point-tracking model [20]. When a human hand demonstration is provided, HAND tracks the hand trajectory with the same simple pipeline. The hand positions are then converted into 2D *relative* sub-trajectories, capturing motion agnostic to the starting point [21]. After an initial filtering step that removes unrelated behaviors using a visual foundation model [22], HAND retrieves matching sub-trajectories from the play dataset based on the 2D relative hand path. Finally, towards *speed*, a policy pre-trained on the play dataset is LoRA-fine-tuned on the retrieved sub-trajectories, encouraging the policy to specialize in the demonstrated task. Because HAND retrieves primarily based on hand motion, it is robust to irrelevant visual features such as background clutter and lighting changes compared to purely visual retrieval methods.

Our experiments, across **10 tasks** and **550 total evaluations** in the real world on a WidowX robot demonstrate that HAND enables quick adaptation even to long-horizon tasks,

979-8-3315-8160-2/26/$31.00 ©2026 IEEE

outperforming the best baseline by 3× in task completion. We also demonstrate that HAND is effective with hand demonstrations collected from *completely different scenes* from the robot scene and across significant camera angle changes. Finally, we perform a *real-time learning* experiment, where HAND learns a challenging long-horizon task in **under 4 minutes** of experiment time, from providing the hand demonstration to the trained policy, while being on average **5**× faster to collect data for than robot teleoperation demonstrations on our WidowX arm.

II. Related Works

Robot Data Retrieval. Prior work has demonstrated *retrieval* as an effective mechanism for extracting relevant on-robot data for training robots [10, 11, 12, 13, 14, 15, 23, 24]. For example, SAILOR [10] and Behavior Retrieval [11] pre-train variational auto-encoders (VAEs) on prior robot images and actions to learn a latent embedding. This latent embedding is used to retrieve states and actions from an offline dataset similar to ones provided in expert demonstration trajectories. However, retrieving based on learned full image encodings or even raw pixel values [14] can be noisy; Flow-Retrieval [12] instead trains a VAE to encode *optical flows* indicating movement of objects and the robot arm in the scene. Similar to Flow-Retrieval, HAND also retrieves based on robot arm motion. However, rather than training a dataset-specific VAE model that may not be robust to large visual differences, we retrieve from our offline robot data by primarily matching motions of a human hand demonstration using *relative 2D paths* of the robot end-effector in the prior data. This hand path retrieval helps us robustly retrieve relevant robot arm *behaviors*.

STRAP [13] addresses visual retrieval robustness issues of prior work by using features from DINO-v2 [22], a large pre-trained image-input foundation model for retrieval. However, STRAP, along with all aforementioned retrieval work, assumes access to expert robot demonstrations for the target tasks. In contrast, HAND only requires a *single*, easier-to-collect human hand demonstration that results in more *time-efficient* learning of demonstrated tasks compared to methods requiring robot teleoperation data. Furthermore, our experiments demonstrate that HAND retrieves more task-relevant trajectories and therefore attains higher success rates compared to these methods.

Learning From Human Hands. Similar to HAND, a separate line of work proposes using human hands to learn robot policies. One approach is to train models on human video datasets to predict future object flows [25, 26] or human affordances [27, 28]. These intermediate affordance and flow representations are then used to either train a policy conditioned on this representation [25] on robot data or control a heuristic policy [26, 27, 28]. Other works focus on learning directly from human hands [16, 17, 18, 19, 29]. These works generally use hand-pose detection models aided by multiple cameras or calibrated depth cameras to convert hand poses directly to robot gripper keypoints [16, 17, 19]. However, works that exclusively retrieve human data are restricted to constrained policy representations as they must match human hand poses to robot gripper poses. Kim et al. [18] instead use an eye-in-hand camera mounted on a human demonstrator's forearm to train an imitation learning policy conditioned on robot eye-in-hand camera observations. Unlike these prior works, HAND only requires a single RGB camera from which the robot gripper can be seen. Also, we focus on retrieving robot play data, allowing us to train arbitrarily expressive policies without constrained policy representations [16, 17, 19] or intermediate representations [25, 26, 27, 28].

III. HAND: Fast Robot Adaptation via Hand Path Retrieval

A. Preliminaries and Formulation

We assume access to a dataset of task-agnostic robot play data, $\mathcal{D}_{\text{play}}$, consisting of trajectories $\tau_i = \{(o_t, a_t)\}_{i=1}^{T}$, where each o_t is a per-timestep observation that includes RGB images of the robot gripper and robot proprioceptive information, and a_t is the robot action. These trajectories may span many scenes, tasks, and episode horizons. We do not assume task labels (e.g., language labels), as data collection is easier to scale without labeling each sub-trajectory in a long-horizon play trajectory.[1]

In contrast to retrieval methods that rely on robot demonstrations for each target task [10, 11, 12, 13], we assume access to easy-to-provide human hand demonstrations. For each task, a human records their hand movement without teleoperating the robot. On our real-world setup, these hand demonstrations, $\mathcal{D}_{\text{hand}}$, are on average $5\times$ faster to collect than robot teleoperation data. Moreover, hand demonstrations are generally easier to provide than robot teleoperation [30, 31]. Each video in $\mathcal{D}_{\text{hand}}$ consists of RGB frames $o_1, \ldots, o_H$, captured such that the human hand occupies a similar viewpoint in the frame as the robot gripper does in $\mathcal{D}_{\text{play}}$.[2]

Given $\mathcal{D}_{\text{play}}$ and $\mathcal{D}_{\text{hand}}$, we aim to train a policy $\pi_\theta(a \mid o)$ to perform the target task demonstrated by the human in $\mathcal{D}_{\text{hand}}$. Since we do not assume task labels in $\mathcal{D}_{\text{play}}$ and we are provided no expert robot teleoperation demonstrations, we must *retrieve* sub-trajectories indicating how to perform the behavior demonstrated in $\mathcal{D}_{\text{hand}}$ from $\mathcal{D}_{\text{play}}$ for training π. We denote this retrieved dataset, later used for imitation learning, as $\mathcal{D}_{\text{retrieved}}$. Moreover, following our motivation in Section I, we aim for our method to be *fast*, so that non-expert end-users can easily train the robot for many downstream tasks.

The key challenges HAND addresses are: (1) designing a representation that can unify the behaviors in robot sub-trajectories and human hand demonstrations (Section III-B), (2) retrieving relevant sub-trajectories based on a suitable distance metric between these representations (Section III-C), and (3) quickly training a policy that can perform various unseen target tasks with a high success rate without expert demonstrations (Section III-D). See Figure 2 for an overview.

[1] Section IV demonstrates that HAND can also incorporate language labels as extra policy conditioning.

[2] Section IV demonstrates HAND works under large camera angle shifts.

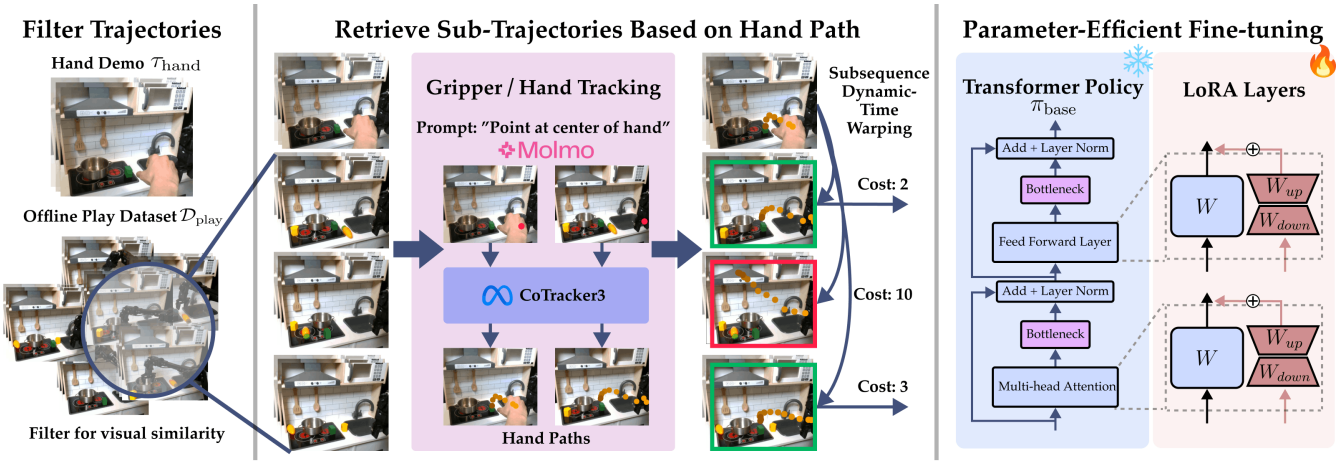

Fig. 2: **HAND** enables fast-adaptation to a new target task by using an easy-to-provide hand demonstration of the target task (Left). We propose a two-step retrieval procedure where we first filter the trajectories in the offline play dataset, $\mathcal{D}_{\text{play}}$, for visually similar trajectories based on features from a pretrained vision model. We use off-the-shelf, pretrained hand detection and point tracking to construct 2D paths of the motion for both the human hand and robot end-effector. We use these paths as a distance metric to retrieve relevant trajectories from the play dataset (Middle) for quickly fine-tuning a pretrained transformer policy on the target task (Right).

B. Path Distance as a Unifying Representation for Retrieval

Prior robot retrieval methods assume access to expert demonstrations from which they extract proprioceptive information (e.g., joint angles and actions) alongside visual features for retrieval [10, 11, 12, 13, 14]. However, since $\mathcal{D}_{\text{hand}}$ contains only visual data and no robot actions, retrieval based purely on appearance can be noisy—especially due to the visual domain gap between hand demonstrations in $\mathcal{D}_{\text{hand}}$ and robot demonstrations in $\mathcal{D}_{\text{play}}$ (see Figure 2, left). To address these issues, we propose an embodiment-agnostic, behavior-centric retrieval metric that enables matching between $\mathcal{D}_{\text{hand}}$ and $\mathcal{D}_{\text{play}}$ based on demonstrated behaviors rather than appearance.

Using 2D Paths for Retrieval. The movement of the robot end-effector over time provides rich information about its behavior [4]. We represent behaviors in both datasets using the paths traced by the human hand or the gripper. Because we assume access only to an RGB camera from which the hand or the gripper is visible (i.e., no depth), we construct these paths in 2D relative to the camera viewpoint for both $\mathcal{D}_{\text{play}}$ and $\mathcal{D}_{\text{hand}}$.[3]

Obtaining Paths from Data. To extract paths, we use CoTracker3 [20], an off-the-shelf point tracker capable of tracking 2D points across video sequences, even under occlusion. CoTracker3 only requires a single point on the gripper or hand to track motion across the full sequence. We use Molmo-7B [32], an open-source 7B image-to-point foundation model, to automatically select this point by prompting it at the *midpoint* of each trajectory with either "Point at the center of the hand" or "Point to the robot gripper." Using the middle frame ensures a higher chance of visibility in case the gripper or hand is not yet in frame at the beginning or occluded at the end.[4]

Given the 2D point $(x, y)_{\text{hand}}$ or $(x, y)_{\text{play}}$ from the middle frame, we use CoTracker3 to perform bidirectional point tracking, resulting in a 2D path $p_{\text{hand}} = \{(x_t, y_t)_{\text{hand}}\}_{t=1}^{H}$ or $p_{\text{play}} = \{(x_t, y_t)_{\text{play}}\}_{t=1}^{T}$ for each trajectory. See the Gripper/Hand Tracking block of Figure 2 for a visualization of this pipeline. Next, we describe how we use 2D paths to retrieve sub-trajectories from $\mathcal{D}_{\text{play}}$.

C. Retrieving Relevant Sub-Trajectories using Path Distance

Background. For identifying relevant sub-trajectories in $\mathcal{D}_{\text{play}}$, we use Subsequence Dynamic Time Warping (S-DTW) [33], an algorithm for aligning a shorter sequence to a portion of a longer reference sequence prior work has demonstrated effective for sub-trajectory retrieval [13]. Given a query sequence $Q = \{q_1, q_2, \ldots, q_H\}$ and a longer reference sequence $R = \{r_1, r_2, \ldots, r_T\}$, where $T > H$, the goal of S-DTW is to find a contiguous subsequence of R that minimizes the total cumulative distance between elements of both sequences. In HAND, the query sequences are the 2D hand demonstration paths $\{(x_t, y_t)_{\text{hand}}\}_{t=1}^{H}$ and the reference sequences are the 2D paths generated from long-horizon robot play data $\{(x_t, y_t)_{\text{play}}\}_{t=1}^{T}$.

Sub-Trajectory Preprocessing. To preprocess the datasets for S-DTW, we first segment the offline play dataset, $\mathcal{D}_{\text{play}}$, into variable-length sub-trajectories using a simple heuristic based on proprioception proposed in several prior works [13, 34]. In particular, we split the trajectories whenever the acceleration or velocity magnitude (depending on what proprioception data is available) drops below a predefined ϵ value, corresponding to when the teleoperator switches between tasks. We find that this simple heuristic can reasonably segment trajectories into

[3] If both datasets have additional calibrated depth information, HAND can also operate on 3D paths.

[4] Points can also be obtained heuristically, e.g., if the robot starts from the same position in each $\mathcal{D}_{\text{play}}$ traj.

atomic components resembling lower-level primitives. We also split the hand demonstrations evenly into smaller sub-trajectories based on how many subtasks the human operator determined they have completed. After sub-trajectory splitting, we have two sub-trajectory datasets, $\mathcal{T}_{\text{hand}} = \{t^i_{1:a}, t^i_{a:b}, \ldots, t^i_{H_i - |p^i_{\text{hand}}|:H_i} \forall\, \tau^i_{\text{hand}} \in \mathcal{D}_{\text{hand}}\}$ and $\mathcal{T}_{\text{play}} = \{t^j_{1:a}, t^j_{a:b}, \ldots, t^j_{T_j - |p^j_{\text{play}}|:T} \forall\, \tau^j_{\text{play}} \in \mathcal{D}_{\text{play}}\}$ where $|p^i_{\text{hand}}|$ and $|p^j_{\text{play}}|$ are the lengths of the last sub-trajectory paths of trajectories i, j from $\mathcal{D}_{\text{hand}}$ and $\mathcal{D}_{\text{play}}$, respectively. Finally, each sub-trajectory is represented in *relative 2D coordinates*, i.e., $p_t = [x_{t+1} - x_t, y_{t+1} - y_t]$. Relative coordinates ensure retrieval invariance to the initial positions of the hand or gripper [21].

Visual Filtering. A key limitation of path distance retrieval is that distinct tasks can exhibit similar movement patterns. For example, tasks like "pick up the mug" and "pick up the cube" can appear nearly identical in 2D path space [4]. But, the retrieved trajectories for one task may not benefit learning of the other; since we do not assume task labels in $\mathcal{D}_{\text{play}}$, a policy directly trained on "pick up the cube" retrieved sub-trajectories may still fail to pick up a mug. Therefore, before retrieving sub-trajectories with paths, we first run a visual filtering step to ensure that the sub-trajectories we retrieve will be task-relevant. We use an object-centric visual foundation model, namely DINOv2 [22], to first filter out sub-trajectories performing unrelated tasks with different objects. Specifically, we use the DINOv2 first and final frame embedding differences, representing visual object movement from the first to last frame, between human hand demonstrations and robot play data to filter $\mathcal{T}_{\text{play}}$. In practice, this simple filtering step removes the majority of irrelevant sub-trajectories. For a given image sequence $o^{\text{hand}}_{1:H}$ from a hand sub-trajectory and image sequence $o^{\text{play}}_{1:T}$ from a robot play sub-trajectory, we define the cost as:

$$\begin{aligned} \text{C}_{\text{visual}}(o^{\text{hand}}_{1:H}, o^{\text{play}}_{1:T}) = &\underbrace{||\text{DINO}(o^{\text{hand}}_1) - \text{DINO}(o^{\text{play}}_1)||^2_2}_{\text{first frame DINO embedding difference}} \\ &+ \underbrace{||\text{DINO}(o^{\text{hand}}_H) - \text{DINO}(o^{\text{play}}_T)||^2_2}_{\text{last frame DINO embedding difference}}. \quad (1) \end{aligned}$$

We take the M trajectories with lowest cost as possible retrieval trajectories from $\mathcal{D}_{\text{play}}$ for each human demonstration sub-trajectory in $\mathcal{T}_{\text{hand}}$. The rest are ignored for those hand demonstrations.

Retrieving Sub-Trajectories. We then employ S-DTW to match the target sub-trajectories, $\mathcal{T}_{\text{hand}}$, to the set of visually filtered segments $\in \mathcal{T}_{\text{play}}$. Given two sub-trajectories, $t_i \in \mathcal{T}_{\text{play}}$ and $t_j \in \mathcal{T}_{\text{hand}}$, S-DTW returns the cost along with the start and end indices of the subsequence in t_j that minimizes the path cost (see Figure 2). We select the K matches from $\mathcal{D}_{\text{play}}$ with the lowest cost to construct our retrieval dataset, $\mathcal{D}_{\text{retrieved}}$.

D. Putting it All Together: Fast-Adaptation with Parameter-Efficient Policy Fine-tuning

We aim to enable fast, data-efficient learning of the task demonstrated in $\mathcal{D}_{\text{hand}}$. To this end, we first pretrain a task-agnostic base policy π_{base} on $\mathcal{D}_{\text{play}}$ with standard behavior cloning (BC) loss. While our approach is compatible with any policy architecture, we use action-chunked transformer policies [35] due to their suitability for low-parameter fine-tuning and strong performance in long-horizon imitation learning [3, 35, 36, 37].

Adapting to $\mathcal{D}_{\text{retrieved}}$. To rapidly adapt to a task with minimal data, we leverage parameter-efficient fine-tuning using *task-specific adapters*—small trainable modules that modulate the behavior of the frozen base policy. Adapter-based methods have shown promise in few-shot imitation learning [38, 39], making them ideal for our limited retrieved dataset $\mathcal{D}_{\text{retrieved}}$. Specifically, we insert LoRA layers [40] into the transformer blocks of π_{base}. These are low-rank trainable matrices (about 0.1%–2% of the parameters of π_{base}) added to the attention projections of π_{base} (see Figure 2, LoRA Layers). During fine-tuning, we update only the parameters of these LoRA layers, θ, using $\mathcal{D}_{\text{retrieved}}$.

Loss Reweighting. While our retrieval mechanism identifies sub-trajectories relevant to the target task, not all will be equally useful. Following prior work [14, 15, 24], we reweight the BC loss with an exponential term $\in (0, \infty)$ (similar to AWR [41]), where each sub-trajectory is weighted based on its S-DTW similarity to the hand demonstration. Intuitively, this upweights the loss of the most relevant examples in $\mathcal{D}_{\text{retrieved}}$ and downweights those that are less relevant. Finally, because trajectory cost scales vary depending on the task being retrieved and the features being used for S-DTW, we rescale the S-DTW costs $C_{i,\text{path}}$ to a fixed range. For each $\tau_i \in \mathcal{D}_{\text{retrieved}}$, its weight $e^{-C_{i,\text{path}}}$ is scaled to between $[0.01, 100]$, where the normalization term comes from the sum of costs of all trajectories in $\mathcal{D}_{\text{retrieved}}$. Let the normalized weight for a trajectory be $w_i = \exp(-C_{i,\text{path}})$ and the behavioral cloning loss be $L_i(a, o) = -\log \pi_\theta(a \mid o)$. The total loss is then the weighted average over the dataset $\mathcal{D}_{\text{retrieved}}$:

$$\mathcal{L}_{\text{BC};\theta} = \frac{1}{|\mathcal{D}_{\text{retrieved}}|} \sum_{\tau_i \in \mathcal{D}_{\text{retrieved}}} w_i \times L_i(a, o). \quad (2)$$

We summarize HAND in the pseudocode in Algorithm 1.

IV. EXPERIMENTS

We evaluate HAND both as a retrieval pipeline and as a method for quickly learning downstream tasks. To this end, we organize our experiments to answer the following questions:

- **(Q1)** How well can HAND retrieve *task-relevant* behaviors?
- **(Q2)** Does HAND support hand demonstrations from *unseen scenes* and is it *robust* to visual shifts?
- **(Q3)** How does HAND perform in policy learning?
- **(Q4)** Can HAND enable *real-time* adaptation?

A. Experimental Setup

We evaluate HAND on a real-world multi-task kitchen environment using the WidowX robot arm. Our robot environment setup is shown in Figure 3. We use an Intel

Algorithm 1 HAND PSEUDOCODE

Require: $\mathcal{D}_{\text{hand}}$, $\mathcal{D}_{\text{play}}$, threshold ϵ, # visual-filtered trajectories M, # retrieved sub-trajectories K

Train base policy π_{base} on $\mathcal{D}_{\text{play}}$ via behavior cloning
Segment both $\mathcal{D}_{\text{play}}$ and $\mathcal{D}_{\text{hand}}$ into sub-trajectory datasets w/ threshold ϵ: $\mathcal{T}_{\text{play}}$, $\mathcal{T}_{\text{hand}}$
for $\tau^{\text{hand}} \in \mathcal{T}_{\text{hand}}$ **do**
    Filter top-M visually similar $\tau^{\text{play}} \in \mathcal{T}_{\text{play}}$ via DINO-based C_{visual}
    for each filtered τ^{play} **do**
        Track 2D hand paths with Molmo + CoTracker3
        Retrieve K best-matching segments via S-DTW on relative path similarity
Fine-tune π_{base} on retrieved data with adapter layers θ to obtain π_θ with $\mathcal{L}_{\text{BC};\theta}$ Equation (2)
return π_θ

Realsense D435 camera as an external camera and a Logitech C920 as an over-the-shoulder camera.

Evaluation Tasks: We evaluate on 10 total tasks: three standard tasks—REACH GREEN BLOCK, PRESS BUTTON, and CLOSE MICROWAVE—and three challenging long-horizon tasks—PUT K-CUP IN COFFEE MACHINE, BLEND CARROT, and COOK CARROT. The latter tasks demand high precision and span more than 150 timesteps at a 5 Hz control frequency. In particular, COOK CARROT is composed of four shorter tasks, SLIDE POT $\rightarrow$ PUT OBJECT IN POT $\rightarrow$ PUT LID ON POT $\rightarrow$ TURN STOVE KNOB, including non-prehensile tasks (e.g., slide pot) and taking $\sim$ 300 steps to complete even for expert teleoperators. For our long-horizon tasks, we provide one hand demonstration to perform retrieval for each subtask. Partial success is provided for tasks composed of multiple subtasks.

Play Dataset Collection: We collect a task-agnostic play dataset containing a total of 50k transitions, each trajectory having an average of 230 timesteps and covering multiple tasks, collected at 5 Hz. The full dataset required roughly four hours to collect. To prevent the play data from mirroring evaluation tasks, we place distractor objects in the environment for teleoperators to interact with during collection. The dataset is split into two, with about 1 hour corresponding to the scene for COOK CARROT. To evaluate language-conditioned methods, we manually annotate the COOK CARROT scene of the dataset with language, which takes an additional 87 minutes. During data collection and evaluation, movable task objects are randomized in a 5" x 7" region within the workspace.

Baselines: We compare HAND to the following baselines:

- π_{base}: the base policy pre-trained on play data;
- LCBC: π_{base} with language-conditioning;
- CLIP-L: retrieves based on cosine similarity between language embedding of target task (rather than hand demo) and language embedding of the play data;
- CLIP-I: retrieves based on cosine similarity between language embedding of target task and image embedding of the play data;

Fig. 3: **WidowX Robot Arm Setup.** We evaluate the scalability of HAND on 10 manipulation tasks on a WidowX robot arm in a kitchen setup [43].

	Reach Green Block	Push Button	Close Microwave
Flow	7/25	0/25	0/25
STRAP	5/25	0/25	2/25
HAND (-VF)	9/25	13/25	9/25
HAND	**15/25**	**18/25**	**11/25**

TABLE I: **Number of retrieved sub-trajectories performing the demonstrated task.** HAND retrieves more task-performing sub-trajectories than Flow and STRAP.

- Flow [12]: trains a VAE on pre-computed optical flows for $\mathcal{D}_{\text{play}}$ from GMFlow [42] and retrieves individual states-action pairs based on latent motion similarity; and
- STRAP [13]: also uses S-DTW for sub-trajectory retrieval but uses S-DTW distance based solely on Euclidean distance between pre-trained DINO-v2 image embeddings.

STRAP and Flow assume access to expert *robot* demonstrations for both retrieval and fine-tuning. In our setting, we do not assume such demonstrations, therefore, unless otherwise noted, we adopt them without expert fine-tuning. While STRAP and Flow originally propose training policies from scratch, we instead apply LoRA fine-tuning—as with HAND —which we found to yield better performance for these baselines.

Policy Architecture: To ensure fair comparison, all methods use a three-layer action-chunking transformer (similar to ACT [35]) decoder policy where applicable. The input to the transformer policy is a sequence of image tokens corresponding to the external and over-shoulder camera views. Conditioned on the current image observation, the model predicts an action chunk corresponding to a second of execution.

B. Experimental Evaluation

(Q1): HAND retrieves more task-relevant data. We analyze the quality of retrieved sub-trajectories between Flow, STRAP, and HAND. STRAP and HAND both use S-DTW-based trajectory retrieval, but STRAP relies purely on visual DINO-v2 embeddings for retrieval. We provide a single hand demonstration of three real robot tasks and retrieve the top $K = 25$ matches from $\mathcal{D}_{\text{play}}$. As shown in Table I, HAND retrieves more task-relevant trajectories than both STRAP and Flow. STRAP relies exclusively on

Fig. 4: **Qualitative retrieval results on OOD scene.** We visualize the top sub-trajectory match of Flow, STRAP, HAND (−VF), and HAND on two OOD domain demonstrations recorded from an iPhone camera, showing approaching a K-Cup and putting it into the machine. Only HAND's top match is relevant for both hand demonstrations.

Method	10° Horiz.	20° Horiz.	30° Horiz.	10° Vert	20° Vert	30° Vert
Flow	0 / 25	2 / 25	5 / 25	1 / 25	0 / 25	0 / 25
STRAP	1 / 25	10 / 25	13 / 25	12 / 25	11 / 25	11 / 25
HAND	**21 / 25**	**18 / 25**	**19 / 25**	**16 / 25**	**13 / 25**	**14 / 25**

TABLE II: **Camera angle robustness results**. Number of relevant retrieved trajectories for PUT LID ON POT if we change the camera angle vertically and horizontally by 10° increments. HAND retrieves +18% more relevant trajectories compared to STRAP even in the extreme case of 30° shift.

visual similarity, while Flow relies exclusively on motion similarity. Both methods struggle to bridge the domain gap between human hand demonstrations and robot play data. In particular, for PUSH BUTTON, STRAP is unable to retrieve any relevant trajectories in its top matches.

We also observe that **visual filtering is necessary** to retrieve trajectories where the target object is interacted with, as demonstrated by HAND (−VF), an ablation of HAND without visual filtering (Section III-C), having 30% worse retrieval performance than HAND in Table I.

(Q2): HAND supports hand demonstrations from unseen environments and is robust to camera angle shifts. Because HAND retrieves based on *relative hand motions*, it is also effective with hand demonstrations from out-of-distribution (OOD) scenes. To illustrate, we collect hand demonstrations in a new environment using a handheld iPhone camera and a real coffee machine, while retrieving from robot play data recorded in a completely different scene with a toy coffee machine. In Figure 4, we show the lowest cost retrieved sub-trajectory of STRAP and Flow compared to HAND and a HAND ablation without the visual filtering step, HAND (−VF). Both of the retrieved trajectories for STRAP and Flow, along with the top trajectory for HAND (−VF) are irrelevant to the demonstrated task. For the first task, STRAP is able to retrieve the initial reaching motion toward the K-Cup but misses the crucial grasping segment, as it does not leverage motion for retrieval. Only HAND retrieves relevant robot trajectories for both hand demonstrations because it focuses on the *motion* demonstrated by the human hand after *visual filtering*.

Table II shows that **HAND is also robust to shifts in camera angle** for the PUT LID ON POT task, far more than Flow and STRAP. We measure the number of relevant

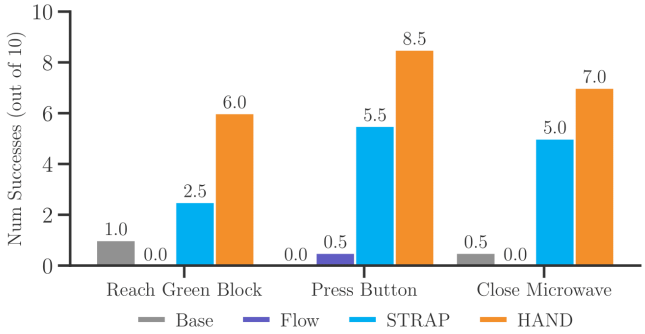

Fig. 5: **Real Robot Results.** Task completion (including partial completion) out of 10 of π_{base}, STRAP, Flow, and HAND.

Method	Slide Pot	Put Obj in Pot	Put Lid on Pot	Turn Knob	Long Horiz.
LCBC	2	0	0	0	0
CLIP-L	0	4	1	0	0
STRAP	0	0	2	0	0
HAND w/o Pre-training	1	1	1	2	0
HAND (−VF)	2	0	0	5	0
HAND (−CW)	2	5	3	5	0
HAND	**5**	**6**	**4**	**6**	**3**
HAND + LCBC	**8**	**7**	**5**	**7**	**5**

TABLE III: **Long horizon COOK CARROT task results**. We show success rates on each subtask and on the full task execution. Successes are out of 10.

retrieved trajectories of different methods after vertical and horizontal camera angle shifts of 10° increments. In the most extreme setting of 30° shift, HAND retrieves +18% more relevant trajectories compared to STRAP. These camera angle shifts emulate head rotations on humanoid robots or camera movement on mobile manipulators, suggesting that HAND can even work in such settings where camera viewpoint change may occur.

(Q3): HAND enables efficient policy learning in the real world. We evaluate four methods, including ours, across three standard tasks with ten trials each, for a total of **120 evaluations**. Real-world experiments in Figure 5 on our three standard tasks demonstrate that fine-tuning with HAND improves success rates by **+45%** over the next best baseline, STRAP. In contrast, Flow fails to learn a policy that achieves reasonable success rates in any of the tasks. We also report the performance of π_{base}, trained on all of $\mathcal{D}_{\text{play}}$ and note that the pre-trained policy struggles to perform the tasks without any task-specific fine-tuning.

We next evaluate eight methods, including several ablations of HAND, across four base tasks and one long-horizon task constructed from these base tasks, with ten trials each for a total of **400 evaluations**. Results on the more challenging long-horizon tasks (Table III) demonstrate that retrieval using **hand demonstrations outperforms language-based retrieval** (CLIP-L) by a factor of **3×** in success rate. Language-based retrieval suffers from the lack of spatial awareness, often retrieving trajectories that are semantically correct but spatially misaligned—similar

Fig. 6: **Fast Adaptation Study.** We conduct a small-scale user study to demonstrate HAND's ability to learn robot policies in real-time. From providing the hand demonstration (Left), to retrieval and fine-tuning a base policy (Middle), to evaluating the policy (Right), we show HAND can learn to put a carrot in the blender with 7.5/10 task completion in less than 4 minutes.

	Time (Min) ↓		Success Rate ↑	
Method	**User 1**	**User 2**	**User 1**	**User 2**
HAND (Hand Demo)	**3**	**2**	**5/10**	**4/10**
STRAP (Robot Demo)	10	14	3/10	2.5/10

TABLE IV: **Hand vs. Robot Teleoperation.** HAND uses a single hand demonstration while STRAP uses robot teleoperation demonstrations. HAND achieves higher success rates in significantly less data collection time.

to STRAP —which makes policy fine-tuning more difficult. In contrast, **directly conditioning on language performs poorly** compared to retrieval (LCBC), despite the fact that annotating sub-trajectories with language more than doubles data collection and annotation time.

Ablation Study: We observe that each component of HAND, namely pretraining, visual filtering (VF), and cost weighting (CW), are critical for task performance. Cost weighting helps bias the resulting policy towards behaviors that are most relevant to the downstream task, and reduces the effect of potentially noisy retrievals that may not directly aid in learning the target task. Without any of these components, the resulting policy is unable to learn the task. Only HAND completes COOK CARROT, succeeding in 3 out of 10 trials. We also show that given access to a language-annotated dataset, one could add language-conditioning on top of HAND to further improve the task performance (HAND + LCBC).

(Q4): HAND enables real-time, data-efficient policy learning of long-horizon tasks. Finally, we performed two small-scale user studies with IRB approval from our institution to demonstrate real-time learning. In the first study shown in Figure 6, a participant familiar with HAND iteratively demonstrated each part of a long-horizon BLEND CARROT task and trained a HAND policy ***with over 70% success rate in under four minutes*** from providing a single hand demonstration to deploying the fine-tuned policy. A video of a similar experiment can be found on our website.

Hand vs Robot Demonstration Comparison: In the second study, two users with prior teleoperation experience—but not affiliated with this research—each collected a total of 20 demonstrations: 10 using hand demonstrations and 10 using robot teleoperation, to train the robot for PUT K-CUP IN COFFEE MACHINE. We employ HAND retrieval for hand-collected demonstrations and STRAP retrieval for robot teleoperation demonstrations. For a direct comparison, we additionally fine-tune STRAP with the human-collected teleoperated demonstrations as per [13].

As reported in Table IV, teleoperated demonstrations required over 3× more time to collect than hand demonstrations. Remarkably, with just a single hand demonstration per user, we fine-tuned a policy achieving over 40% task completion compared to STRAP which reaches only 25% using a single *robot teleoperation* demonstration. Interestingly, we observed that increasing the number of expert demonstrations for STRAP degraded downstream performance likely due to lower quality retrieved trajectories. These results demonstrate that HAND enables *fast* adaptation to downstream tasks with as few as one easy-to-provide hand demonstration.

V. CONCLUSION AND LIMITATIONS

We presented HAND, a simple and time-efficient framework for adapting robots to tasks using easy-to-provide human hand demonstrations. We demonstrated that HAND enables *real-time* task adaptation with a *single* hand demonstration in under four minutes.

Extending to 3D paths for retrieval. While HAND uses 2D paths for retrieval, one future direction could extend HAND to estimate the hand trajectory in 3D using foundation depth prediction models. Another direction future work could consider is a mixture of features for improving retrieval for tasks that require more dexterous control, i.e., cloth folding or deformable object manipulation.

Severe Camera Viewpoint Changes. Future work could address issues from severe camera viewpoint shifts between the collected hand demonstrations and robot play data via the use of 3D information, multiple camera viewpoints, or scene re-rendering with virtual cameras [44].

ACKNOWLEDGEMENTS

We thank Yigit Korkmaz and Yutai Zhou for their participation and assistance in our user study. We also thank the USC Center for Advanced Research Computing (CARC) for providing us with compute resources.

REFERENCES

[1] Octo Model Team et al., "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[2] M. J. Kim et al., "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[3] K. Black et al., "*pi*_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[4] Y. Li et al., "HAMSTER: Hierarchical action models for open-world robot manipulation," in *International Conference on Learning Representations*, 2025.

[5] G. R. Team et al., *Gemini robotics: Bringing ai into the physical world*, 2025.

[6] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao, "Data scaling laws in imitation learning for robotic manipulation," in *The Thirteenth International Conference on Learning Representations*, 2025.

[7] C. Lynch et al., "Learning latent plans from play," in *Conference on Robot Learning*, 2020.

[8] S. Young, J. Pari, P. Abbeel, and L. Pinto, "Playful interactions for representation learning," in *International Conference on Intelligent Robots and Systems*, IEEE, 2022.

[9] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *Robotics and Automation Letters (RA-L)*, 2022.

[10] S. Nasiriany, T. Gao, A. Mandlekar, and Y. Zhu, "Learning and retrieval from prior data for skill-based imitation learning," in *Conference on Robot Learning*, 2022.

[11] M. Du, S. Nair, D. Sadigh, and C. Finn, "Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets," in *Robotics: Science and Systems*, 2023.

[12] L.-H. Lin, Y. Cui, A. Xie, T. Hua, and D. Sadigh, "Flowretrieval: Flow-guided data retrieval for few-shot imitation learning," in *Conference on Robot Learning*, 2024.

[13] M. Memmel, J. Berg, B. Chen, A. Gupta, and J. Francis, "STRAP: Robot sub-trajectory retrieval for augmented policy learning," in *International Conference on Learning Representations*, 2025.

[14] K. Sridhar, S. Dutta, D. Jayaraman, and I. Lee, "REGENT: A retrieval-augmented generalist agent that can act in-context in new environments," in *International Conference on Learning Representations*, 2025.

[15] A. Xie, R. Chand, D. Sadigh, and J. Hejna, "Data retrieval with importance weights for few-shot imitation learning," in *9th Annual Conference on Robot Learning*, 2025.

[16] G. Papagiannis, N. Di Palo, P. Vitiello, and E. Johns, "R+ x: Retrieval and execution from everyday human videos," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2025, pp. 8284–8290.

[17] S. Haldar and L. Pinto, "Point policy: Unifying observations and actions with key points for robot manipulation," *arXiv preprint arXiv:2502.20391*, 2025.

[18] M. J. Kim, J. Wu, and C. Finn, "Giving robots a hand: Learning generalizable manipulation with eye-in-hand human video demonstrations," *CoRR*, 2023.

[19] M. Lepert, J. Fang, and J. Bohg, "Phantom: Training robots without robots using only human videos," in *Conference on Robot Learning (CoRL)*, Seoul, Korea, 2025.

[20] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht, "Cotracker: It is better to track together," in *European Conference on Computer Vision*, 2025.

[21] J. Zhang et al., "EXTRACT: Efficient policy learning by extracting transferrable robot skills from offline data," in *Conference on Robot Learning*, 2024.

[22] M. Oquab et al., *Dinov2: Learning robust visual features without supervision*, 2024.

[23] K. Kedia, P. Dan, A. Chao, M. A. Pace, and S. Choudhury, "One-shot imitation under mismatched execution," in *International Conference on Robotics and Automation (ICRA)*, 2025.

[24] K. Sridhar, S. Dutta, D. Jayaraman, and I. Lee, "Ricl: Adding in-context adaptability to pre-trained vision-language-action models," in *Conference on Robot Learning (CoRL)*, PMLR, 2025.

[25] M. Xu et al., "Flow as the cross-domain manipulation interface," in *Conference on Robot Learning*, 2024.

[26] C. Yuan, C. Wen, T. Zhang, and Y. Gao, "General flow as foundation affordance for scalable robot learning," *Conference on Robot Learning*, 2024.

[27] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 778–13 790.

[28] Y. Kuang et al., "Ram: Retrieval-based affordance transfer for generalizable zero-shot robotic manipulation," *Conference on Robot Learning*, 2024.

[29] S. Kareer et al., "Egomimic: Scaling imitation learning via egocentric video," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2025, pp. 13 226–13 233.

[30] J. Xie, Z. Xu, J. Zeng, Y. Gao, and K. Hashimoto, "Human–robot interaction using dynamic hand gesture for teleoperation of quadruped robots with a robotic arm," *Electronics*, 2025.

[31] H. Li, Y. Cui, and D. Sadigh, "How to train your robots? the impact of demonstration modality on imitation learning," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2025, pp. 1113–1120.

[32] M. Deitke et al., "Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models," *arXiv preprint arXiv:2409.17146*, 2024.

[33] M. Müller, *Fundamentals of music processing: Using Python and Jupyter notebooks*. Springer, 2021, vol. 2.

[34] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning*, 2023.

[35] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Robotics: Science and Systems*, K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, Eds., 2023.

[36] T. Z. Zhao et al., *Aloha unleashed: A simple recipe for robot dexterity*, 2024.

[37] S. Haldar, Z. Peng, and L. Pinto, "Baku: An efficient transformer for multi-task policy learning," *Neural Information Processing Systems*, 2024.

[38] A. Liang, I. Singh, K. Pertsch, and J. Thomason, "Transformer adapters for robot learning," in *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022.

[39] Z. Liu et al., "TAIL: Task-specific adapters for imitation learning with large pretrained models," in *International Conference on Learning Representations*, 2024.

[40] E. J. Hu et al., "Lora: Low-rank adaptation of large language models.," *International Conference on Learning Representations*, 2022.

[41] X. B. Peng, A. Kumar, G. Zhang, and S. Levine, "Advantage-weighted regression: Simple and scalable off-policy reinforcement learning," *arXiv preprint arXiv:1910.00177*, 2019.

[42] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, and D. Tao, "Gmflow: Learning optical flow via global matching," in *Conference on Computer Vision and Pattern Recognition*, 2022.

[43] H. Walke et al., "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning*, 2023.

[44] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox, "Rvt2: Learning precise manipulation from few demonstrations," *Robotics: Science and Systems*, 2024.