Importance Weighted Retrieval for Few-shot Imitation Learning

Amber Xie
Stanford

Rahul Chand
Stanford

Dorsa Sadigh
Stanford

Joey Hejna
Stanford

***Abstract*—While large-scale robot datasets have propelled recent progress in imitation learning, learning from smaller task specific datasets remains critical for deployment in new environments and unseen tasks. One such approach to few-shot imitation learning is retrieval-based imitation learning, which extracts relevant samples from large, widely available prior datasets to augment a limited demonstration dataset. To determine the relevant data from prior datasets, retrieval-based approaches most commonly calculate a prior data point's minimum distance to a point in the target dataset in latent space. While retrieval-based methods have shown success using this metric for data selection, we demonstrate its equivalence to the limit of a Gaussian kernel density (KDE) estimate of the target data distribution. This reveals two shortcomings of the retrieval rule used in prior work. First, it relies on high-variance nearest neighbor estimates that are susceptible to noise. Second, it does not account for the distribution of prior data when retrieving data. To address these issues, we introduce Importance Weighted Retrieval (IWR), which estimates importance weights, or the ratio between the target and prior data distributions for retrieval, using Gaussian KDEs. By considering the probability ratio, IWR overcomes the bias of previous selection rules, and by using reasonable modeling parameters, IWR effectively smooths estimates using all data points. Across both simulation environments and real-world evaluations on the Bridge dataset we find that our method, IWR, consistently improves performance of existing retrieval-based methods, despite only requiring minor modifications.**

I. Introduction

Data has been integral to the performance of deep learning-based methods across a wide variety of domains [22, 23]. Unsurprisingly, the same has found to be true for imitation learning (IL) methods in robotics, for which the most compelling examples often require hundreds to thousands of collected demonstrations in order to learn a single task [30]. Unfortunately, this makes scaling IL difficult. When trying to learn a new task, one needs to collect a large number of demonstrations to achieve a reasonable level of in-domain success while simultaneously ensuring sufficient diversity for generalization to out-of-distribution scenarios.

One approach for learning from limited demonstrations is *retrieval* from prior datasets. Retrieval-based methods augment small demonstration datasets with relevant samples taken from large, widely available robotics datasets [21]. This is typically done by learning a representation of all state-action pairs, and "retrieving" those from the prior dataset that are most similar to the target demonstration data according to some metric, e.g. distance in latent space [6, 14, 20]. By providing the policy with additional relevant state-action pairs, retrieval reduces the need for collecting additional expert demonstrations for the target task.

Though this has held in practice, the derivation of retrieval-based methods has largely hailed from intuition. For example, if a data point in the prior dataset is close to that of a target demonstration in latent space, intuitively we can hope that adding it to the training dataset might help the learned policy. While this may help justify the performance boost afforded by retrieval-based methods, their design choices are still largely heuristic. In particular, the use of the nearest-neighbor L2 distance metric for scoring prior data is often chosen arbitrarily without principled justification. This begs the question: mathematically, how should we interpret retrieval? Moreover, the possibility remains that a more grounded understanding of retrieval could address the shortcomings of existing heuristic approaches. In this work, we develop such an understanding through the probabilistic lens of importance sampling, and propose a new method for retrieval.

At their core, retrieval-based methods aim to leverage a broader data distribution, denoted as p_{prior}, to estimate the loss of a learned policy on the distribution defined by a set of target demonstrations, p_{t}. Usually, to estimate the expectation of a random variable under a target distribution p with samples from an easier-to-sample-form distribution q, one would weight samples from q by the ratio of probability densities, p/q. Crucially, when dividing by q, these "importance weights" overcomes the bias introduced by using samples from q instead of p. Though retrieval parallels this framework by leveraging samples from p_{prior} to improve behavior cloning on the task defined by p_{t}, existing retrieval methods can be viewed as only approximating the numerator of this ratio p_{t}, leading to inherent bias. Moreover, the use of the aforementioned nearest-neighbor distance metric is of high variance because of its susceptibility to noise. This leads to two avenues for improvement, which we address through our method, Importance Weighted Retrieval, or IWR.

IWR simultaneously addresses both the bias and high-variance of prior retrieval methods by applying Gaussian kernel density estimation (KDE) to estimate the full importance weights $p_{\text{t}}/p_{\text{prior}}$. Using Gaussian KDEs produces smoother estimates by considering all data points within a dataset instead of just the nearest-neighbor (see Fig. 2). Then, by using KDEs to approximate both p_{t} and p_{prior}, IWR uses importance weights to retrieve data, mathematically ensuring

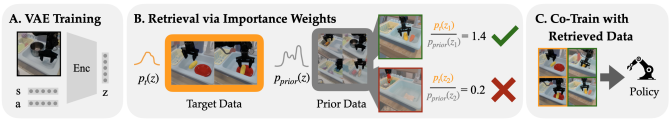


Fig. 1: IWR consists of three main steps: (A) Learning a latent space to encode state-action pairs, (B) Estimating a probability distribution over the target and prior data, and using importance weights for data retrieval, and (C) Co-training on the target data and retrieved prior data. By augmenting our high-quality but much smaller target dataset with diverse, relevant prior samples, we learn more robust and performant policies.

a more accurate approximation of the expectation under p_t (Fig. 1).

Practically, we find that these choices allow IWR to retrieve higher quality data, in terms of relevancy to the target task and diversity across task phases. Moreover, these benefits are not limited to a single retrieval-based IL method – we find that IWR consistently improves the quality of retrieved data when applied to a number of different prior works by simply replacing nearest-neighbor distance queries with estimated importance weights. For example, on the LIBERO benchmark, using IWR increases average success rates by 5.8% on top of SAILOR [20], 4.4% on top of Flow Retrieval [14], and 5.8% on top of Behavior Retrieval [6]. On real-world tasks with the Bridge V2 Dataset [27], we find the performance improvement afforded by IWR to be more significant, increasing success rate by 30% on average, in comparison to Behavior Retrieval.

II. Related Work

Retrieval Retrieval enables few-shot imitation learning by augmenting a target dataset, consisting of a handful of demonstrations for a task, with additional prior data from pre-existing datasets such as DROID [12], OpenX [21], and Bridge [27]. Typically, the retrieved prior data is used to augment the dataset for imitation learning [6, 14, 19], with the focus of these works primarily about retrieval, not algorithmic improvements [20, 14]. BehaviorRetrieval [6] and FlowRetrieval [14] both learn a latent embedding space of state-action pairs and optical flow respectively, and they retrieve prior data points that are closest to the latent embeddings of the target data points. SAILOR [20] learns latent embeddings through skill-based representation learning, and they also retrieve data points that are closest to the target data in latent space. STRAP [19] uses foundation model latent embeddings and retrieves sub-trajectories based on dynamic time warping and proximity in latent space. While most of these works focus on altering latent representations, our work instead focuses on improving the Euclidean distance metric typically used for retrieval.

Importance Sampling Outside of few-shot imitation learning, importance sampling, from which IWR is motivated, has been used for data selection in a variety of domains. Works in imitation learning use importance weights to discard sub-optimal transitions [29] or for off-policy evaluation [9]. In reinforcement learning, importance weights have been used for prioritized sampling [26]. Perhaps most similar to the retrieval problem, importance weights have been used to select relevant documents for training language models [28]. While more complicated techniques for estimating importance weights have been developed, from telescoping classifiers [3, 24] to generative models [5], we find simple Gaussian KDEs to be effective.

Data for Imitation Learning The paradigm of co-training policies with large amounts of additional, diverse [21, 12] or even simulated [16] data has proven effective in imitation learning. While these works take a blanket approach, retrieval seeks to identify the most relevant data for a particular tasks. Other works have sought to identify relevant data, but with different goals in mind. Hejna et al. [10] identify group weights in large robot datasets to accelerate behavior cloning. Others filter data based on various quality metrics [2] using mutual information [11], preferences [13], or demonstrator expertise [1]. Instead of selecting data, other works opt to generate similar data leveraging simulation [8, 18].

III. Preliminaries

A. Problem Setup

The objective of imitation learning (IL) methods is to learn a policy $\pi(a|s)$, which mimics the expert behavior within an environment with states s and actions a. Typically, the policy is learned using a dataset $\mathcal{D}$ of expert trajectories $\tau = \{s_0, a_0, \ldots, s_T, a_T\}$, where s_t, a_t correspond to states and actions at timestep t, and T is the length of the trajectory. While IL has shown success in simulated and real tasks, collecting the requisite expert demonstrations for IL is expensive and has to be repeated for each new task one wishes to learn.

Retrieval-based methods have sought to address this shortcoming by leveraging existing prior data when learning a new task. Specifically, we consider a few-shot setting in which we only have a handful of expert demonstrations for a new target task, our target dataset $\mathcal{D}_t$. Retrieval-based methods assume access to a much larger prior dataset $\mathcal{D}_{\text{prior}}$, consisting of more diverse tasks and scenes. To decrease the number of demonstrations needed in $\mathcal{D}_t$, retrieval-based methods carefully select data from $\mathcal{D}_{\text{prior}}$, which is then used to co-train the policy. We denote this dataset of retrieved state-action pairs as $\mathcal{D}_{\text{ret}} \subset \mathcal{D}_{\text{prior}}$. Mathematically, this leads to the following weighted behavior cloning objective for a parameterized policy π_θ:

$$\max_\theta \quad \alpha \frac{1}{|\mathcal{D}_t|} \sum_{(s,a)\in\mathcal{D}_t} \log \pi_\theta(a|s) + (1-\alpha) \frac{1}{|\mathcal{D}_{\text{ret}}|} \sum_{(s,a)\in\mathcal{D}_{\text{ret}}} \log \pi_\theta(a|s) \tag{1}$$

where α (typically 0.5) is the weighting coefficient between the target and retrieved data. By training on additional data, retrieval-based methods aim to increase the robustness of the learned policy.

Though retrieval methods often differ in the specific representations they learn, most share a common selection mechanism: L2 distance between embedding representations z of the target and prior data. To learn latent embeddings, BehaviorRetrieval [6] trains a variational autoencoder (VAE) over state-action pairs (s_t, a_t), FlowRetrieval [14] learns a VAE over the optical flow for each state s_t, and SAILOR [20] encodes a sub-trajectories $(s_t, a_t, \ldots, s_{t+k}, a_{t+k})$ using

skill-based representation learning. Despite differences in how these latent spaces are learned, all of these methods select data based on the L2 distance. Denoting the learned encoders as f_ϕ and the representations they produce as $z = f_\phi(s, a)$ the selection rule can be written as:

$$\mathcal{D}_{\text{ret}} \coloneqq \left\{(s,a) \in \mathcal{D}_{\text{prior}} \middle| \min_{(s',a') \in \mathcal{D}_{\text{t}}} \|f_\phi(s,a) - f_\phi(s',a')\|_2^2 < \zeta \right\} \quad (2)$$

where ζ is the retrieval threshold, often chosen such that $\mathcal{D}_{\text{ret}}$ is small percentage of $\mathcal{D}_{\text{prior}}$. This intuitively makes sense, as data with representations that are close to those of $\mathcal{D}_{\text{t}}$ are likely useful for training. In the next section, we re-examine retrieval through a probabilistic lens.

B. From samples to densities

To characterize retrieval probabilistically, we define marginal state-action distributions p_{t}, p_{prior}, and p_{ret}, from which we assume $\mathcal{D}_{\text{t}}$, $\mathcal{D}_{\text{prior}}$, and $\mathcal{D}_{\text{ret}}$ are sampled. Then, the previous IL objective (Eq. (1)) becomes

$$\max_\theta \quad \alpha \mathbb{E}_{(s,a)\sim p_{\text{t}}} [\log \pi_\theta(a|s)] + (1-\alpha) \mathbb{E}_{(s,a)\sim p_{\text{ret}}} [\log \pi_\theta(a|s)] . \quad (3)$$

However, for a given target task, we are primarily interested in maximizing the policy likelihood under the target task distribution, i.e. $\mathbb{E}_{(s,a)\sim p_{\text{t}}} [\log \pi_\theta(a|s)]$. In order to maximize this policy likelihood when optimizing Eq. (3), we would like retrieval-based methods to align the distribution of retrieved data such that it is equivalent to that of the target task, i.e. $p_{\text{t}} \approx p_{\text{ret}}$. Then, Eq. (3) would amount to behavior cloning on the target task distribution, which is our desired objective, while using the additional samples from $\mathcal{D}_{\text{ret}}$, which we would expect to improve performance.

Examining the selection rule Eq. (2) used in prior work, we find that it considers the minimum squared distance to a data point in $\mathcal{D}_{\text{t}}$, corresponding to an approximation of p_{t}, as it does not leverage samples from $\mathcal{D}_{\text{prior}}$ (we formalize this connection in in Section IV-A). First, we note that this approximation of p_{t} is imprecise as it only uses the nearest neighbor in $\mathcal{D}_{\text{t}}$, resulting in high variance and susceptibility to noise. Second, even if Eq. (2) were able to perfectly recover p_{t}, it still only uses estimates of p_{t} to retrieve from $\mathcal{D}_{\text{prior}}$. The resulting retrieved distribution p_{ret} is thus closer to the product of densities $p_{\text{t}} \cdot p_{\text{prior}}$ than p_{t}, as the prior data is first sampled according to p_{prior}, and then retrieved according to a condition of p_{t}. We address these limitations through IWR, ensuring that we can retrieve a more accurate approximation of the desirable distribution.

IV. Importance Weighted Retrieval

In this section, we address the shortcomings of the standard retrieval selection rule by introducing our method Importance Weighted Retrieval (IWR). Similar to other retrieval methods, we assume access to an embedding function f_ϕ, typically taken from a VAE. Given the resulting embeddings from f_ϕ, we first discuss how we can better model the distributions used in retrieval with Gaussian KDEs. Second, we discuss how modeling the prior distribution p_{prior}, which we use to compute importance weights $p_{\text{t}}/p_{\text{prior}}$, allows us to retrieve data from the desired distribution p_{t}. Our approach can be applied in conjunction with a broad set of retrieval-based methods to improve performance.

A. Improved Density Modeling

As discussed in Section III-B, standard retrieval methods select data points from $\mathcal{D}_{\text{prior}}$ that minimize the L2 distance from their nearest neighbors in $\mathcal{D}_{\text{t}}$, which may suffer from high variance. To address this, we use lower variance estimates of the probability density function (pdf), which considers all data points (Fig. 2).

The condition from Eq. (2) can equivalently be written as follows using a max instead of a min: $\max_{(s',a')\in\mathcal{D}_{\text{t}}} -\|f_\phi(s,a) - f_\phi(s',a')\|_2^2 > -\zeta$. To smooth this retrieval rule, we replace the hard maximum with a log-sum-exp parameterized by temperature h, resulting in a soft approximation that aggregates contributions from all data points. The retrieved dataset then becomes:

$$\mathcal{D}_{\text{ret}} \coloneqq \left\{(s,a) \in \mathcal{D}_{\text{prior}} \middle| \tfrac{1}{h^2} \log \textstyle\sum_{(s',a')\in\mathcal{D}_{\text{t}}} \right. \quad (4)$$

$$\left. \exp\left(-\|f_\phi(s,a) - f_\phi(s',a')\|_2^2 \Big/ h^2\right) > -\zeta. \right\} \quad (5)$$

Here, the exponential term within the sum is proportional to the multivariate Gaussian pdf $\mathcal{N}$, with mean $f_\phi(s', a')$ and covariance matrix $h^2 I$, evaluated at $f_\phi(s, a)$. This implies that the sum over all such Gaussians – each centered at each data-point in $\mathcal{D}_{\text{t}}$ – is proportional to a Gaussian kernel density estimate (KDE) of $\mathcal{D}_{\text{t}}$, assuming isotropic covariance I and bandwidth h. In the limit as the bandwidth $h \to 0$, the KDE becomes sharply peaked at each data point in $\mathcal{D}$, and the density at a point is dominated by its nearest neighbor – recovering the original retrieval rule from Eq. (3). Under this view, the original retrieval rule can be interpreted as a limiting case of a KDE estimate of p_{t} implying that prior retrieval methods implicitly rely on overly restrictive modeling assumptions.

Instead, we directly model distributions with Gaussian KDEs using well-calibrated parameters that smooth across data points to obtain lower variance estimates (See Fig. 2). Specifically, we employ bandwidths h set to multiplicative factor of Scott's rule [25] and use the sample covariance matrix Σ, giving us

$$p^{\text{KDE}}(z) = \tfrac{1}{|\mathcal{D}|} \textstyle\sum_{z' \in f_\phi(\mathcal{D})} \left((2\pi)^d |h^2\Sigma|\right)^{-1/2} \quad (6)$$

$$\exp\left\{-\tfrac{1}{2}(z-z')^\top (h^2\Sigma)^{-1}(z-z')\right\} \quad (7)$$

where z and z' are the representations of state-action pairs from f_ϕ and $f_\phi(\mathcal{D})$ denotes an encoded dataset. In comparison to the original retrieval rule, using Eq. (6) to model p_{t} prefers retrieving data points near multiple targets and better handles dependencies among features via Σ.

B. Importance Weighting

Our ultimate goal in retrieval is to estimate the expectation of the loss under the target distribution p_{t} using samples from the prior distribution p_{prior}. This bears a striking resemblance

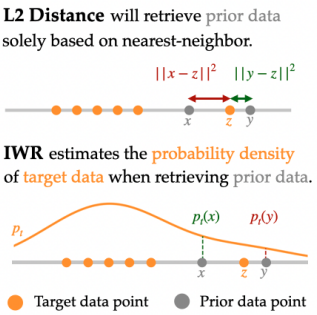


Fig. 2: In this toy example, using L2 distance in latent space leads to the left point discarded, and the right point retrieved. However, when using IWR to estimate the probability density of the target data, the left point is retrieved. This is because IWR has a smoothing effect and uses many target points for retrieval. In this example, we may expect the left point to be relevant, as it is close to many target points, as IWR correctly determines.

to importance sampling. Instead of selecting data according to p_t, we select data according to the importance weight p_t/p_{prior} as

$$\mathbb{E}_{p_{\text{prior}}}\left[p_t/p_{\text{prior}} \log \pi(a|s)\right] = \mathbb{E}_{p_t}\left[\log \pi(a|s)\right] \tag{8}$$

which ensures that the expectation under samples from p_{prior} is the same as that of p_t. Practically, we can use the aforementioned KDE estimators to fit importance weights as $p_t^{\text{KDE}}/p_{\text{prior}}^{\text{KDE}}$ following Eq. (6). Doing so overcomes the bias in the retrieval distribution introduced by the prior dataset $\mathcal{D}_t$.

Then, given a particular threshold we select data points from $\mathcal{D}_{\text{prior}}$ which have the highest estimated importance weights $p_t^{\text{KDE}}/p_{\text{prior}}^{\text{KDE}}$. Though consistent with prior retrieval works, it is not an unbiased estimate. Alternatively, we could obtain an unbiased estimate by following an importance resampling procedure [7], where K data points are sampled from $\mathcal{D}_{\text{prior}}$ based on estimated importance weights. However, such an approach could be higher variance due to the nature of importance sampling procedures, and we thus follow the simple thresholding procedure that easily integrates with prior methods.

C. Putting It All Together

Beginning with a handful of demos in $\mathcal{D}_t$, the final recipe for IWR involves the following steps:

1. Representation Learning. First, we train a model f_ϕ to produce low-dimensional representations z of state-action pairs or sequences. Most prior works in retrieval address this step. For example, SAILOR uses "skill" representation learning while Flow Retrieval uses VAEs on learned from visual flow. IWR is compatible with all types of representation learning, so long as the learned latent dimension is sufficiently small for a Gaussian KDE. In this manner, IWR can be combined with several prior works in retrieval.

2. Importance Weight Estimation. Following Eq. (6), we fit Gaussian KDEs to embedding representations of $\mathcal{D}_t$ and $\mathcal{D}_{\text{prior}}$ to estimate both $p_t \approx p_t^{\text{KDE}}$ and $p_{\text{prior}} \approx p_{\text{prior}}^{\text{KDE}}$. Then, we query the KDEs at z for embedded state-action chunks or sequences in $\mathcal{D}_{\text{prior}}$ to estimate their importance weights $p_t^{\text{KDE}}/p_{\text{prior}}^{\text{KDE}}$. In practice, $\mathcal{D}_{\text{prior}}$ is often too big to fit with a single KDE, so we estimate $p_{\text{prior}}^{\text{KDE}}$ using random batches from $\mathcal{D}_{\text{prior}}$.

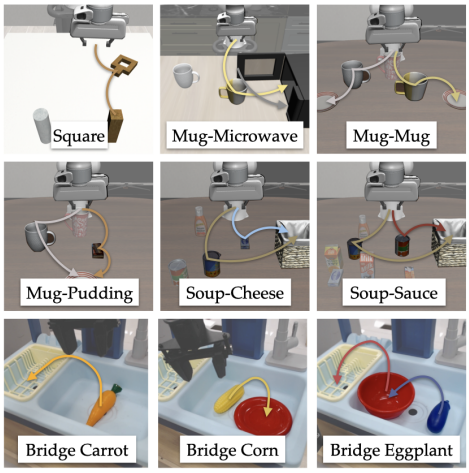


Fig. 3: We evaluate on simulated environments: Robomimic Square, a suite of 5 LIBERO-10 tasks, which each task consisting of two subtasks. For our real experiments, we consider 3 Bridge tasks, with Eggplant being a long-horizon task.

3. Data Retrieval. We then select data with the highest importance weights to train on, following a similar rule to that of Eq. (2) except with importance weights; e.g. $p_t^{\text{KDE}}/p_{\text{prior}}^{\text{KDE}} > \eta$. Similar to prior work, the threshold η is determined experimentally or by examining the distribution of scores.

4. Policy Learning. Finally, we co-train our policy with the retrieved data following Eq. (1).

In comparison to contemporary methods, IWR has minimal overhead as it only modifies the latter steps in retrieval for importance weight estimation.

V. Experiments

In this section we answer the following questions: 1) How much does IWR improve performance? 2) Is IWR broadly applicable to all retrieval methods? 3) What contributes to IWR's performance?

A. Experimental Setup

Simulated Tasks. We evaluate on two simulated domains used in prior retrieval work: Robomimic Square [17] and LIBERO [15].

- **Robomimic Square:** We select the Square Assembly task from Robomimic, a popular imitation learning benchmark. We use the same datasets as Behavior Retrieval [6], where $\mathcal{D}_t$ consists of 10 demonstrations lifting the square nut into the goal peg, and $\mathcal{D}_{\text{prior}}$ consists of 400 trajectories, with 200 placing the square nut on the goal peg, and 200 adversarial episodes with the wrong peg. Similar to prior work [6, 14], we retrieve 30% of $\mathcal{D}_{\text{prior}}$.
- **LIBERO:** We use the LIBERO benchmark [15], and select the 5 tasks from LIBERO-10 with the lowest non-trivial success from [19], where each $\mathcal{D}_t$ consists of 5 demos. Following [19], we use LIBERO-90 as $\mathcal{D}_{\text{prior}}$ and condition our policy on one-hot task vectors. For each task we retrieve 2.5% of the prior data.

Real World Tasks. We further instantiate experiments in the real world using the Bridge setup [27]. We include 3 Bridge tasks: *Corn* with 5 demos, where the robot is tasked with

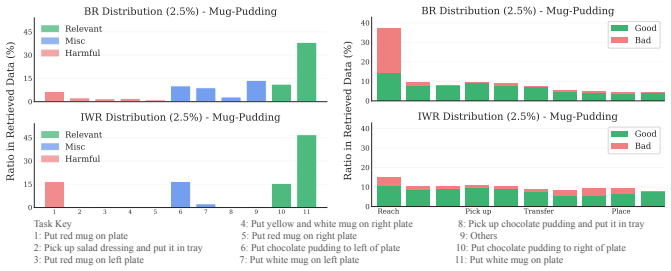


Fig. 4: Difference in retrieval distributions between BR and IWR for the Mug-Pudding task in terms of both tasks (left) and timesteps (right). (Left) Prior tasks which form exact sub-tasks of the target are marked as Relevant, tasks with at least one common object with target as Misc, and the rest as Harmful. (Right) Demonstrations are divided into 10 bins. Green represent samples from either relevant or temporally-appropriate portions of partially-relevant Misc tasks. More details can be found in the Appendix.

moving a corn onto a plate; *Carrot* with 10 demos, where the robot moves a carrot into the sink, and *Eggplant* with 20 demos; a long-horizon task consisting of grasping the eggplant, placing it into the bowl, and transferring the bowl with the eggplant into the dish rack. These three tasks are unseen in the Bridge-V2 dataset. However, the prior dataset consists of similar tasks performed in many toy sinks, including the specific toy sink for our tasks, and thus we may expect transfer from this prior dataset.

Our $\mathcal{D}_{\text{t}}$ consists of VR-teleoperated demonstrations on a WidowX, and our $\mathcal{D}_{\text{prior}}$ consists of a subset of Bridgev2 Dataset [27], where we take trajectories with keyword sink, leading to 130k transitions.

Baselines We compare IWR to popular retrieval baselines:

- **Behavior Cloning (BC)** is only trained to imitate $\mathcal{D}_{\text{t}}$.
- **Behavior Retrieval (BR)** learns a VAE over state-action pairs, and it retrieves data from $\mathcal{D}_{\text{prior}}$ based on L2 distance in the latent space.
- **Flow Retrieval (FR)** learns a VAE over optical flow between frames and actions. Similar to BR, it retrieves data via L2 distance in latent space.
- **SAILOR (SR)** learns a skill-based latent space by compressing state-action chunks. SAILOR retrieves data via L2 distance in latent space.

We evaluate the performance of different methods by training Diffusion Policy [4] on the retrieved and target data following Eq. (1). By default, we use the representations from BR for IWR, as we found them to perform best overall. More details can be found in the Appendix.

B. How much does IWR improve performance?

Simulated Experiments. The left half of Table I provides results for simulated tasks. In Robomimic Square, retrieval is crucial, as training only on $\mathcal{D}_{\text{t}}$ leads to extremely poor performance. Here, where retrieving incorrect data is detrimental, IWR consistently outperforms other methods, likely due to its lower variances estimates of p_{t}. Only FR exhibits similar performance, likely because the prior data includes only two motions (peg to left or right) which is readily captured by visual flow. In LIBERO, where $\mathcal{D}_{\text{prior}}$ is significantly larger and more diverse, SR and FR perform considerably worse, often under-performing BC. For LIBERO, BR faces two challenges: (1) object similarity across tasks results in retrieval of irrelevant demonstrations, and (2) tasks often share similar starting configurations, which bias retrieval towards initial samples instead of more informative later-stage actions. Though these issues are correlated, even if we only retrieve from relevant tasks, the bias towards starting samples can still compromise performance.

IWR addresses these limitations by upweighting samples containing underrepresented objects or occurring later in the demonstrations. Consider the Mug-Pudding task, which requires placing a white mug on a plate and a pudding on the left of the plate. In this case, Fig. 4 shows that BR retrieves irrelevant tasks containing either "chocolate pudding" or "white mug" and also disproportionately samples from the initial phase ($\sim$40%). IWR corrects both issues, retrieving a higher percentage of relevant tasks and a more balanced distribution across timesteps as importance weights correct for the bias in $\mathcal{D}_{\text{prior}}$.

On the other hand, for Mug-Microwave (place mug inside microwave), BR and IWR perform similarly because the critical subtask of "inserting into microwave" is absent from all priors, resulting in a consistent failure mode where objects collide with the microwave. This failure mode cannot be overcome with better or more sampling. In contrast, for Soup-Cheese, BR already achieves high performance due to the task's simplicity and distinctive priors - one of the task's component involves a cheese box that is visually different from other objects in similar task setting (cans, ketchup). With both BR and IWR retrieving over 50% from directly relevant tasks, IWR's improved retrieval offers minimal additional benefit.

Real World Experiments. We evaluate IWR and baselines on a real-world Bridge sink environment, with performance on the three tasks reported in Table I. For all of these tasks, using retrieved prior data improves performance. Qualitatively, the BC policies often early grasp or miss the object, which additional pick-and-place retrieved data helps mitigate. IWR consistently leads to the largest improvements, especially for the long-horizon Eggplant task, where IWR is able to achieve partial success on 100% of rollouts, compared to the best performing retrieval method, which only achieves partial success on 50% of rollouts. For this longer horizon task, the effect of IWR's data retrieval leads to a drastic improvement, as additional data may alleviate issues with accumulating errors across subtasks.

Among other retrieval methods, FR performs poorly across most tasks, whereas SR is particularly successful for Corn. We hypothesize that for tasks in more cluttered scenes, such as Corn, FR's embeddings are not precise enough, because the optical flow may be noisy. SR's performance in Real may be because Bridge tasks have lower-frequency control, so its skill-based, chunking representations consist of more cohesive motion than in simulated tasks.

TABLE I: We evaluate across a suite of *simulated* (Square, Mug-Microwave, .., Soup-Sauce) and *real* (Corn, .., Eggplant) tasks. We find that IWR consistently outperforms previous retrieval baselines. We report success rates in % over 3 seeds for simulated tasks. For real-world tasks, we report success rate over 20 trials. For the long-horizon Eggplant task, we also record Partial Success (PS) for completing subtasks. We bold the best-performing method.

Method	Square	Mug-Microwave	Mug-Mug	Mug-Pudding	Soup-Cheese	Soup-Sauce	Corn	Carrot	Eggpl. Partial	Eggpl. Full
BC	1 ± 0.9	72 ± 0.9	54 ± 3.3	21 ± 2.4	58 ± 6.8	32 ± 2.5	4/20	2/20	9/20	2/20
BR	69 ± 5.0	**81** ± 0.5	81 ± 2.4	33 ± 3.3	**83** ± 4.5	43 ± 2.7	2/20	8/20	8/20	3/20
SR	40 ± 4.9	67 ± 2.4	67 ± 4.7	14 ± 0.9	76 ± 4.3	51 ± 2.2	**12**/20	3/20	10/20	2/20
FR	79 ± 5.0	79 ± 2.2	59 ± 4.3	17 ± 2.9	37 ± 3.8	45 ± 5.5	2/20	3/20	6/20	0/20
IWR	**84** ± 2.8	**81** ± 3.6	**87** ± 2.0	**45** ± 1.4	**83** ± 3.3	**54** ± 5.7	9/20	**14**/20	**20**/20	**11**/20

TABLE II: We ablate existing retrieval embeddings with importance weights for retrieval (IWR).

Method	Square	Mug-Micro.	Mug-Mug	Mug-Pudding	Soup-Cheese	Soup-Sauce	Corn	Carrot	Eggpl. Partial	Eggpl. Full
SR	40 ± 4.9	67 ± 2.4	67 ± 4.7	14 ± 0.9	76 ± 4.3	51 ± 2.2	12/20	3/20	10/20	2/20
SR-IWR	57 ± 9.4	81 ± 3.0	70 ± 3.4	20 ± 0.9	85 ± 1.4	48 ± 3.4	9/20	13/20	16/20	1/20
FR	79 ± 5.0	79 ± 2.2	59 ± 4.3	17 ± 2.9	37 ± 3.8	45 ± 5.5	2/20	3/20	6/20	0/20
FR-IWR	67 ± 1.9	81 ± 4.8	69 ± 1.1	18 ± 0.9	49 ± 3.6	42 ± 3.3	3/20	11/20	8/20	0/20

C. IWR with Other Retrieval Methods

Though we use the latent space from BR for IWR in Table I, IWR can be combined with other learned representations such as those from FR and SR with minimal modification; we simply estimate importance weights using KDE's for retrieval instead of using L2 distance. In Table II, we combine IWR with SR and FR for simulated and real tasks. Across simulated tasks, we generally find that IWR augmented retrieval leads to stronger policy performance. In real, adding IWR to FR improves performance for all tasks. For SR, using IWR is generally helpful with the exception of the Corn task, in which the SR base policy performed exceptionally well. For Eggplant, we found the SR-IWR policy to achieve only one fewer full completion than SR, but it achieved 6 more partial successes, showing its robust real-world performance. Overall, our results suggest that IWR can be a simple but effective way to improve retrieval across different latent representations.

D. Ablations

Importance Weights. While IWR computes importance weights p_t/p_{prior} as described in Eq. (8), prior works only consider p_t. In Table IV, we ablate whether this normalization is important for IWR. Across simulated tasks, we generally find the normalization by p_{prior} to be helpful, suggesting that using importance weights for retrieval leads to effective policy performance.

Bandwidth Parameters. We set the bandwidth factor for all Gaussian KDEs to multiplicative factor of Scott's rule, e.g. $h = c \times |\mathcal{D}|^{-1/(d+4)}$ where d is the dimension of the embedding vectors and c is the multiplicative constant. The bottom row of Table IV shows the performance of IWR in LIBERO when we halve the multiplicative factor c from 4 to 2. While performance is relatively robust, we find that smoother KDEs are slightly better on average.

Retrieval Thresholds. The amount of data retrieved can affect the policy performance. In Table III, we show that IWR's performance across different thresholds. Consistent with prior work, choosing the correct threshold is important. We see this effect in Square where a threshold of more than 50% forces the retrieval of prior data from the incorrect task execution.

VI. Conclusion

We introduce Importance Weighted Retrieval, an importance sampling-inspired method for retrieval. We find that IWR is able to better select data for retrieval, as evidenced by improved performance across both simulated and real tasks, including a long horizon task. Moreover, as IWR can easily be used with several retrieval works, we hope our insights will become standard practice for few-shot imitation learning.

Limitations. While IWR can be effectively used on top of existing latent spaces, we do not tackle the question of what makes an effective latent space, leaving this direction for future work. Moreover, due to the constraints of existing large scale prior datasets, our evaluation is largely limited to pick-place like tasks. Future work may explore retrieval for more complex and dexterous tasks. Finally, IWR assumes the use of Gaussian KDEs, which can become computational intractable and numerically unstable in higher dimensions, ultimately restricting the size of the latent representation that can be used for retrieval. Future work could seek to use more advanced methods for estimating importance weights.

References

[1] Mark Beliaev, Andy Shih, Stefano Ermon, Dorsa Sadigh, and Ramtin Pedarsani. Imitation learning by estimating expertise of demonstrators. In *International Conference on Machine Learning*, pages 1732–1748. PMLR, 2022.

[2] Suneel Belkhale, Yuchen Cui, and Dorsa Sadigh. Data quality in imitation learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Charles H Bennett. Efficient estimation of free energy differences from monte carlo data. *Journal of Computational Physics*, 22(2):245–268, 1976. ISSN 0021-9991. doi: https://doi.org/10.1016/0021-9991(76)

90078-4. URL https://www.sciencedirect.com/science/article/pii/0021999176900784.

[4] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[5] Kristy Choi, Chenlin Meng, Yang Song, and Stefano Ermon. Density ratio estimation via infinitesimal classification. In *International Conference on Artificial Intelligence and Statistics*, pages 2552–2573. PMLR, 2022.

[6] Maximilian Du, Suraj Nair, Dorsa Sadigh, and Chelsea Finn. Behavior retrieval: Few-shot imitation learning by querying unlabeled datasets. *arXiv preprint arXiv:2304.08742*, 2023.

[7] Andrew Gelman and Xiao-Li Meng. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*. John Wiley & Sons, 2004.

[8] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Proceedings of the 2023 Conference on Robot Learning*, 2023.

[9] Josiah Hanna, Scott Niekum, and Peter Stone. Importance sampling policy evaluation with an estimated behavior policy. In *International Conference on Machine Learning*, pages 2605–2613. PMLR, 2019.

[10] Joey Hejna, Chethan Anand Bhateja, Yichen Jiang, Karl Pertsch, and Dorsa Sadigh. Remix: Optimizing data mixtures for large scale imitation learning. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=fIj88Tn3fc.

[11] Joey Hejna, Suvir Mirchandani, Ashwin Balakrishna, Annie Xie, Ayzaan Wahid, Jonathan Tompson, Pannag Sanketi, Dhruv Shah, Coline Devin, and Dorsa Sadigh. Robot data curation with mutual information estimators. *arXiv preprint arXiv:2502.08623*, 2025.

[12] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[13] Sachit Kuhar, Shuo Cheng, Shivang Chopra, Matthew Bronars, and Danfei Xu. Learning to discern: Imitating heterogeneous human demonstrations with preference and representation learning. In *7th Annual Conference on Robot Learning*, 2023.

[14] Li-Heng Lin, Yuchen Cui, Amber Xie, Tianyu Hua, and Dorsa Sadigh. Flowretrieval: Flow-guided data retrieval for few-shot imitation learning. In *8th Annual Conference on Robot Learning*.

[15] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023.

[16] Abhiram Maddukuri, Zhenyu Jiang, Lawrence Yunliang Chen, Soroush Nasiriany, Yuqi Xie, Yu Fang, Wenqi Huang, Zu Wang, Zhenjia Xu, Nikita Chernyadev, Scott Reed, Ken Goldberg, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation, 2025. URL https://arxiv.org/abs/2503.24361.

[17] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

[18] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*, 2023.

[19] Marius Memmel, Jacob Berg, Bingqing Chen, Abhishek Gupta, and Jonathan Francis. STRAP: Robot sub-trajectory retrieval for augmented policy learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4VHiptx7xe.

[20] Soroush Nasiriany, Tian Gao, Ajay Mandlekar, and Yuke Zhu. Learning and retrieval from prior data for skill-based imitation learning. In *Conference on Robot Learning (CoRL)*, 2022.

[21] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024.

[22] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[23] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=1ikK0kHjvj. Featured Certification, Outstanding Certification.

[24] Benjamin Rhodes, Kai Xu, and Michael U. Gutmann. Telescoping density-ratio estimation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4905–4916. Curran Associates, Inc., 2020. URL

https://proceedings.neurips.cc/paper_files/paper/2020/file/33d3b157ddc0896addfb22fa2a519097-Paper.pdf.

[25] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[26] Samarth Sinha, Jiaming Song, Animesh Garg, and Stefano Ermon. Experience replay with likelihood-free importance weights. In *Learning for Dynamics and Control Conference*, pages 110–123. PMLR, 2022.

[27] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[28] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227, 2023.

[29] Sheng Yue, Jiani Liu, Xingyuan Hua, Ju Ren, Sen Lin, Junshan Zhang, and Yaoxue Zhang. How to leverage diverse demonstrations in offline imitation learning. *arXiv preprint arXiv:2405.17476*, 2024.

[30] Tony Z. Zhao, Jonathan Tompson, Danny Driess, Pete Florence, Seyed Kamyar Seyed Ghasemipour, Chelsea Finn, and Ayzaan Wahid. ALOHA unleashed: A simple recipe for robot dexterity. In *8th Annual Conference on Robot Learning*, 2024. URL https://openreview.net/forum?id=gvdXE7ikHI.

TABLE IV: We ablate removing the denominator p_{prior} in IWR and halving the Gaussian KDE bandwidth parameter.

Method	Square	Mug-Micro.	Mug-Mug	Mug-Pudding	Soup-Cheese	Soup-Sauce
IWR	84 ± 2.8	81 ± 3.6	87 ± 2.0	45 ± 1.4	83 ± 3.3	54 ± 5.7
IWR (w/o norm)	61 ± 1.0	79 ± 2.1	93 ± 0.5	41 ± 5.4	83 ± 3.8	52 ± 4.7
IWR (1/2 bw)	–	79 ± 1.4	92 ± 1.6	40 ± 0.9	79 ± 1.4	56 ± 1.6

APPENDIX

Please find videos of sample task rollouts and more on our site: https://sites.google.com/view/iwr-corl/home

Hyperparameters. We provide hyper-parameters for all retrieval methods in Table V and for Diffusion Policy [4], which was used for all policy learning evaluations, in Table VI. While we found the original hyper-parameters from Lin et al. [14], Du et al. [6] to work well for Robomimic and Bridge, we found that they did not perform well for LIBERO, likely due to the use of large action chunks. We thus modified the VAE to accept action chunks, use state history, and down-weight image reconstruction which lead to overall better performance for all methods and baselines. This obviated the need to additionally append the action to the learned representation z as done by Lin et al. [14], Du et al. [6].

TABLE III: IWR across retrieval thresholds.

Square	
% Ret	Success
20	71 ± 4.1
30	84 ± 2.8
50	88 ± 1.9
60	54 ± 5.9

Architectures. For all VAEs we use a ResNet18 encoder-decoder architecture, with MLPs in between to process the concatenated image embeddings, robot proprioceptive state, and actions. For LIBERO only, we additionally use a small 2 layer transformer for the action encoder and decoder instead of projection layers from the MLP.

Simulation Evaluation Procedure. For evaluating policies in sim we run three seeds for 100k timesteps. We evaluate each policy every 25K steps for 50 episodes. Following the evaluation procedure of Mandlekar et al. [17], Chi et al. [4], we take the average of the best performing checkpoint across all seeds.

Real Evaluation Procedure. For real world evaluations we train a single policy for a fixed number of timesteps and run 20 evaluation trials.

TABLE V: Hyperparameters used for retrieval methods.

Method	Parameter	Robomimic	LIBERO	Bridge
Shared	Optimizer		Adam	
	Learning Rate		0.0001	
	Batch Size		256	
	Training Steps	200,000	200,000	400,000
	Image Resolution	(84, 84)	(128, 128)	(224, 224)
	Augmentations		None	
	β		0.0001	
	z dim	16	32	32
BR	Image Recon Weight	1	0.01	1
	Action Chunk	1	16	4
	Append Action	TRUE	FALSE	TRUE
	State History	1	2	1
FR	Steps between Flow Frames	8	8	8
	Image Recon Weight	1	0.01	1
	Action Chunk	1	16	4
	Append Action	TRUE	FALSE	TRUE
	State History	1	2	1
SAILOR	Obs & Action Chunk		10	
	Time Loss Weight		0.000001	
	Training Steps		200,000	
	Max Seq Offset		50	

TABLE VI: Diffusion Policy hyperparameters used for policy learning evaluations.

Parameter	Value
Optimizer	Adam
Learning Rate	0.0001
Batch Size	256
Training Steps	100,000
Obs History	2
Action Chunk	16
Image Resolution	See Table V
Augmentations	Random Scale and Crop (0.85, 1.0)

In Fig. 5 - Fig. 9, we provide additional visualizations of the retrieved data across BR and IWR for all the Libero simulated tasks. In Fig. 10 we provide visualization for the Robomimic Square task.

Retrieval Distribution Across Tasks For the task-based retrieval visualization (plotted on left in Fig. 5 - Fig. 9), we classify each retrieved task as "Relevant" (green), "Mixed" (blue) or "Harmful" (red). We now explain how we determine whether the task is Relevant, Mixed, or Harmful. First, a task is Relevant if it corresponds exactly to the target task. For example, the target task Mug-Pudding Fig. 7 (Put white mug on the plate and put chocolate pudding to right of the plate) has two relevant tasks: "Put chocolate pudding to right of the plate" and "Put white mug on the plate".

We classify tasks from the prior dataset as Mixed if part of the trajectory is similar to the target task. For instance, learning how to pick up a target object is useful, even if the prior task is otherwise different. For instance, for Mug-Pudding, "Put chocolate pudding to left of plate" can be useful if we retrieve from the Reach/Pick-up portion of the trajectory, and thus, we label this prior task as Mixed. Similarly, for Mug-Pudding, "Put red mug on left plate" is classified as Mixed, because the action of placing an object on the plate is useful for the target task.

The remaining tasks are marked as harmful since they have nothing in common with the target task.

Retrieval Distribution Across Timesteps For timesteps (plotted on the right in Fig. 5 - Fig. 9), demonstrations are divided into 10 equal bins. Green bars represent samples from either relevant tasks or temporally-appropriate portions of partially-relevant mixed tasks (e.g. initial "Reach" and "Pickup" steps from "Put chocolate pudding to left of plate" are relevant to the target task even though the final portion is not).

Across all plots, we see that IWR consistently helps in both (1) retrieving a higher portion of directly relevant tasks and (2) retrieves a more balanced distribution across timesteps. For Robomimic Square Fig. 10, since there are only two prior tasks, both of which are visually very similar, the performance gains in IWR are likely due to retrieving more samples from the reach/pick-up section (grasping the square), which is where most of the failure cases are present.

Note: The visualizations in the Appendix include the following minor changes compared to the figure in the main paper: (1) The legend "Misc" is replaced with "Mixed" since this labeling better captures the tasks listed under it. (2) The "Others" bin is now marked as "Harmful" (red) instead of "Misc" (blue) since all tasks in the "Others" category share neither a common object nor an ending configuration and are therefore adversarial if retrieved. (3) Tasks with similar ending configurations (such as the earlier example,"Put red mug on left plate") were originally marked as "Harmful" instead of "Misc"/"Mixed" in the main paper plot, which we have now updated. In order to adhere to these stricter definitions, we have corrected these plots, including an updated version of Figure 4 in the main paper. We plan to update the main paper when possible. Note that these changes do not affect the retrieved values or the conclusions, but instead, more rigorously characterize retrieved tasks.

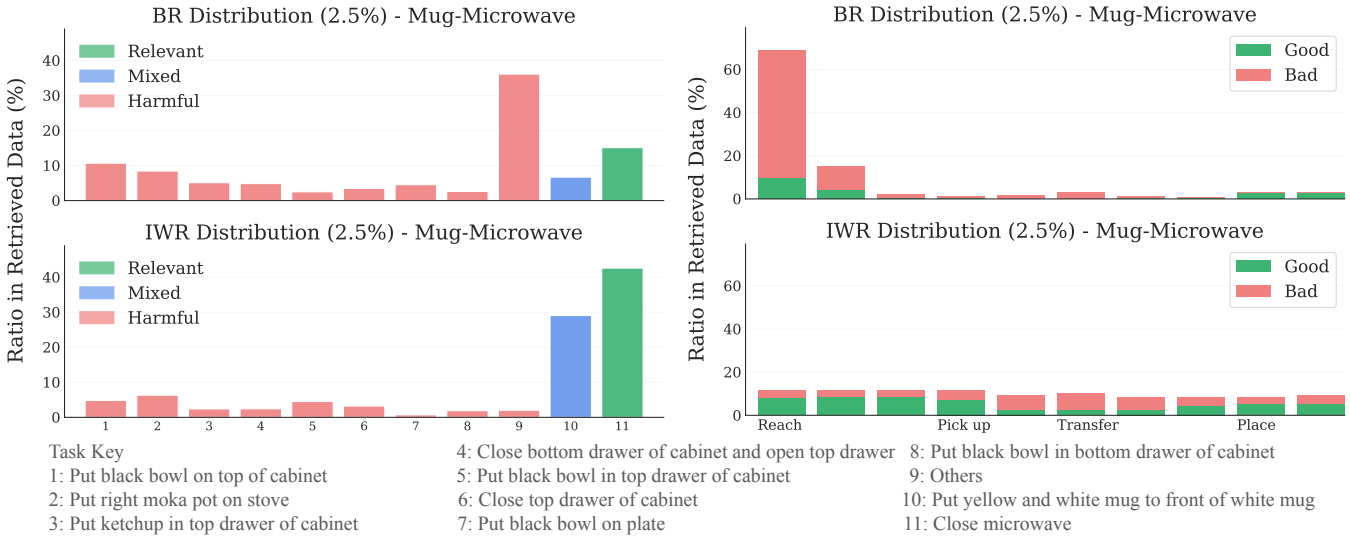


Fig. 5: Mug-Microwave LIBERO Task. (Left) Retrieval distribution across tasks. **(Right)** Retrieval distribution across timesteps.

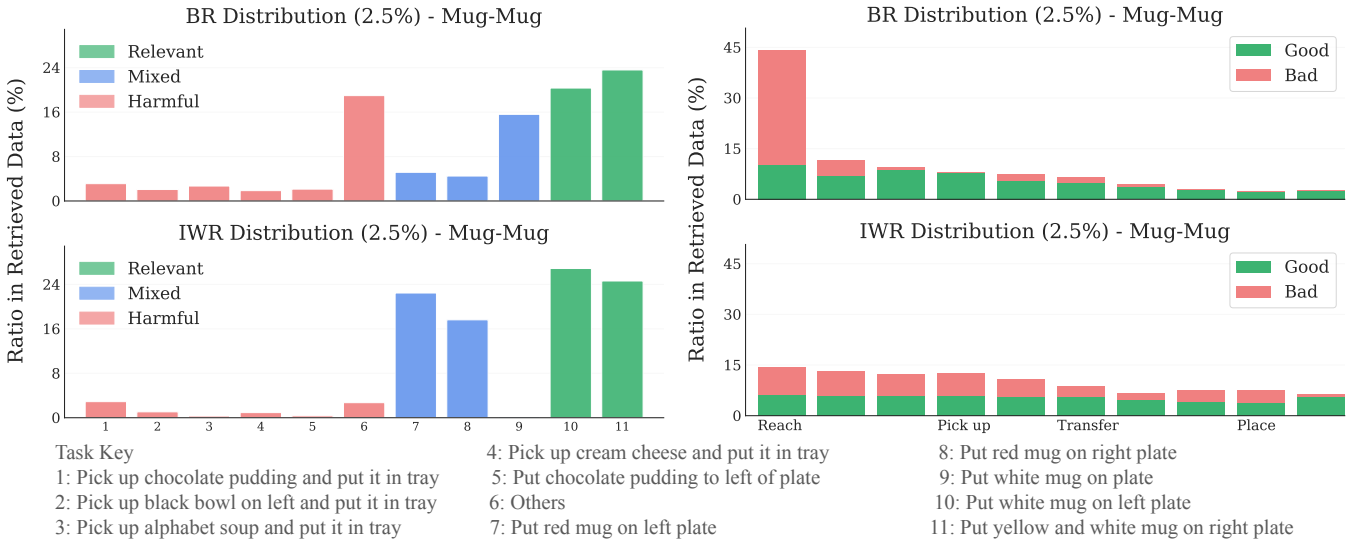


Fig. 6: Mug-Mug LIBERO Task. (Left) Retrieval distribution across tasks. **(Right)** Retrieval distribution across timesteps.

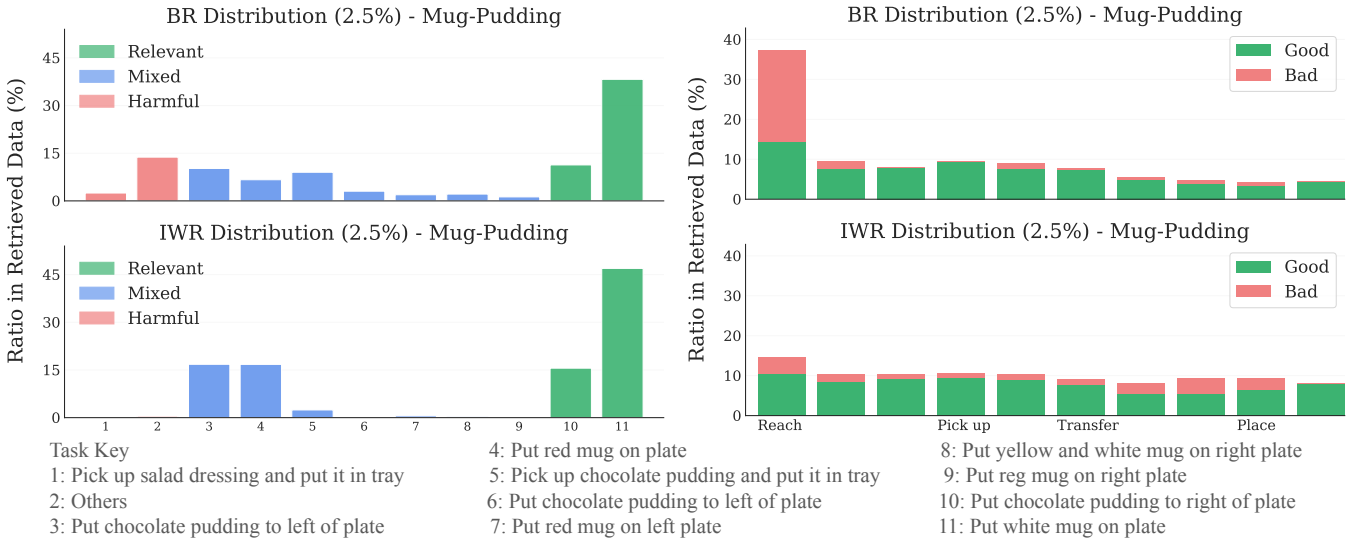
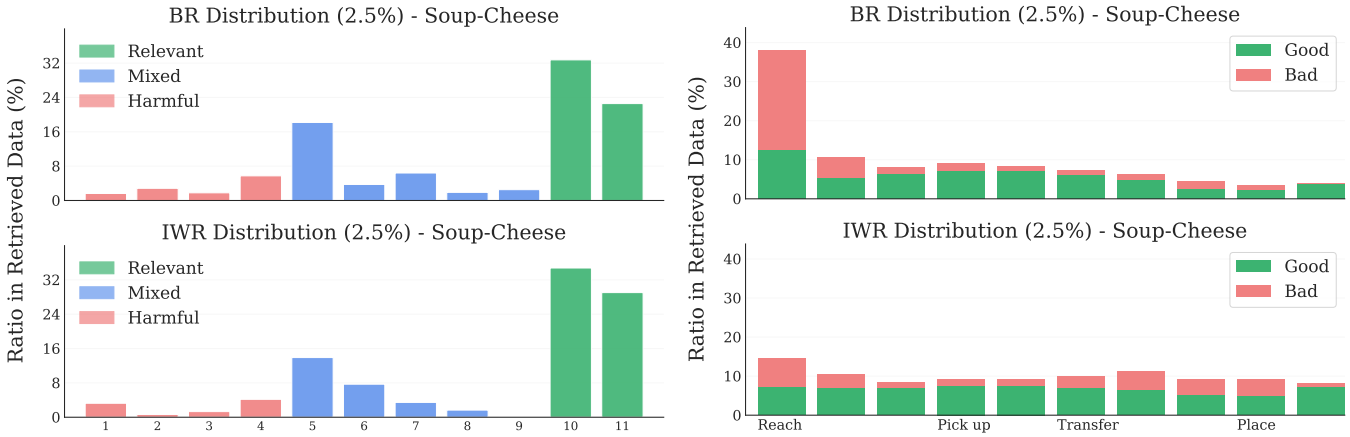


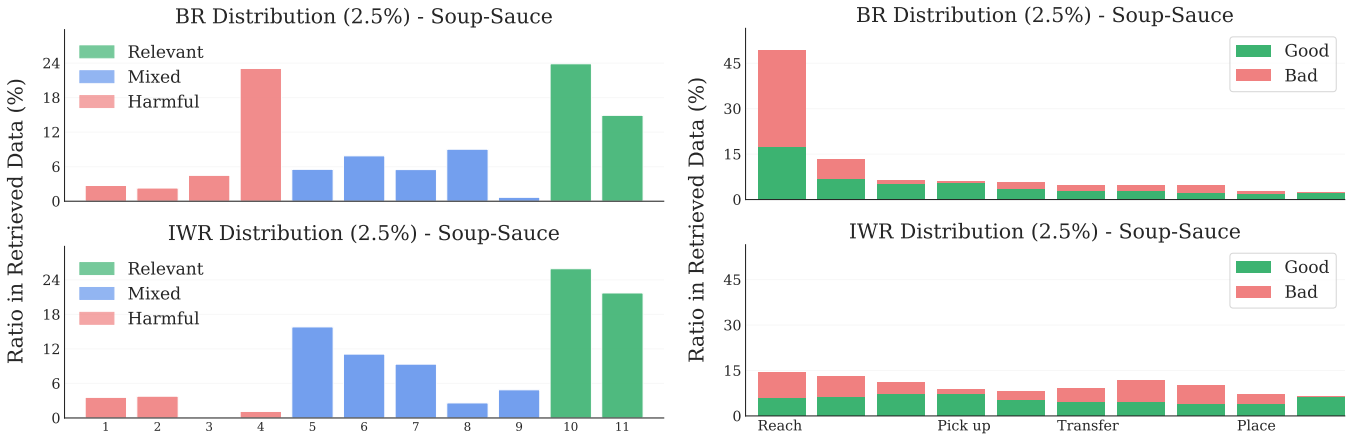
Fig. 7: Mug-Pudding LIBERO Task. (Left) Retrieval distribution across tasks. **(Right)** Retrieval distribution across timesteps.



Task Key

1: Pick up cream cheese and put it in tray
2: Pick up butter and put it in tray
3: Pick up ketchup and put it in tray
4: Others
5: Pick up tomato sauce and put it in basket
6: Pick up ketchup and put it in basket
7: Pick up milk and put it in basket
8: Pick up alphabet soup and put it in tray
9: Pick up orange juice and put it in basket
10: Pick up alphabet soup and put it in basket
11: Pick up cream cheese box and put it in basket

Fig. 8: Soup-Cheese LIBERO Task. (Left) Retrieval distribution across tasks. **(Right)** Retrieval distribution across timesteps.



Task Key

1: Pick up cream cheese and put it in tray
2: Pick up butter and put it in tray
3: Pick up ketchup and put it in tray
4: Others
5: Pick up milk and put it in basket
6: Pick up orange juice and put it in basket
7: Pick up tomato sauce and put it in tray
8: Pick up alphabet soup and put it in tray
9: Pick up butter and put it in basket
10: Pick up alphabet soup and put it in basket white mug
11: Pick up tomato sauce and put it in basket

Fig. 9: Soup-Sauce LIBERO Task. (Left) Retrieval distribution across tasks. **(Right)** Retrieval distribution across timesteps.

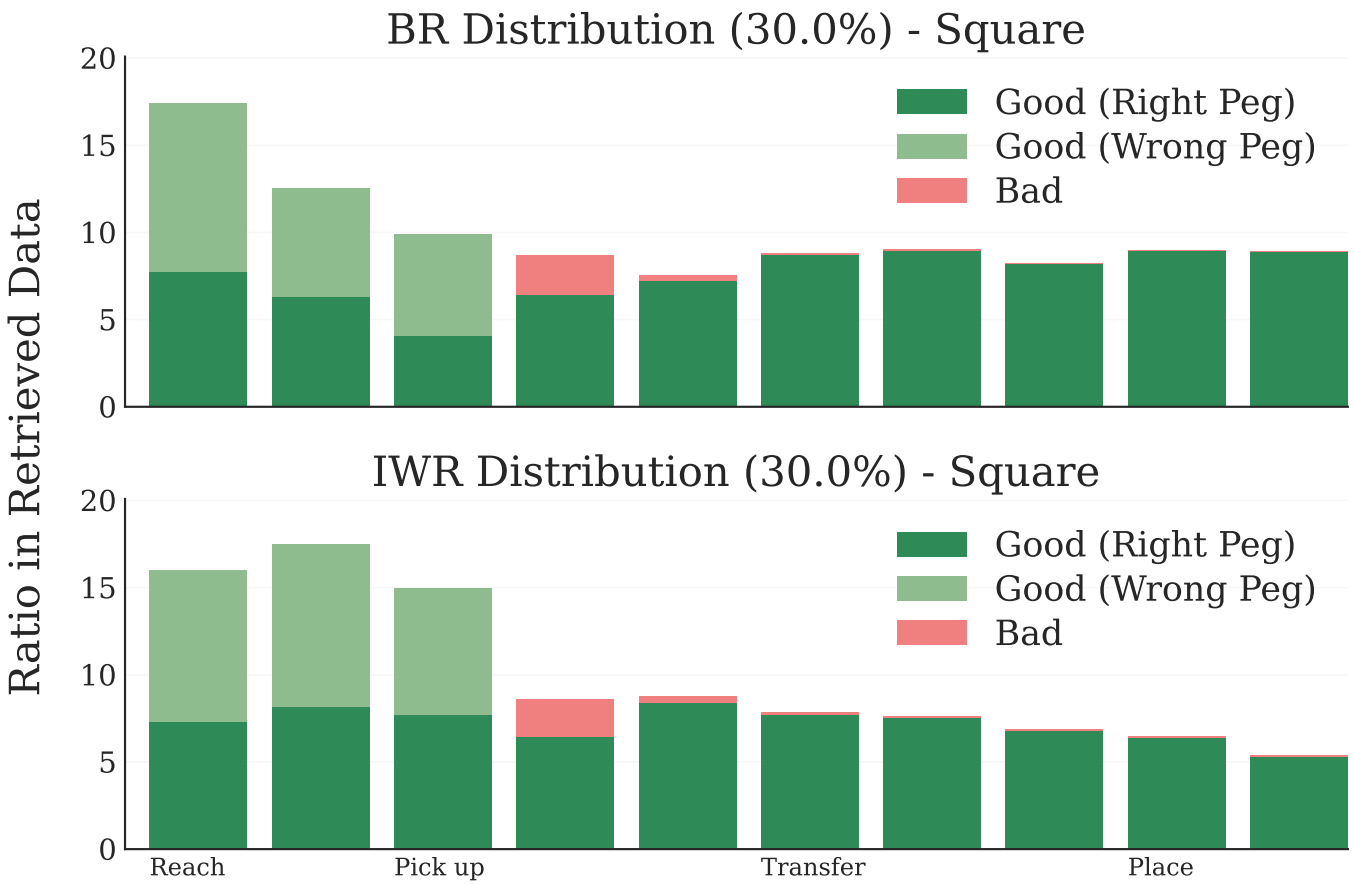


Fig. 10: Retrieval distribution across timesteps for **Robomimic Square Task.**