# Adapting by Analogy: OOD Generalization of Visuomotor Policies via Functional Correspondence

**Pranay Gupta** **Henny Admoni** **Andrea Bajcsy**
The Robotics Institute,
Carnegie Mellon University, United States
Correspondence: pranaygu@andrew.cmu.edu

**Abstract:** End-to-end visuomotor policies trained using behavior cloning have shown a remarkable ability to generate complex, multi-modal low-level robot behaviors. However, at deployment time, these policies still struggle to act reliably when faced with out-of-distribution (OOD) visuals induced by objects, backgrounds, or environment changes. Prior works in interactive imitation learning solicit corrective expert demonstrations under the OOD conditions—but this can be costly and inefficient. We observe that task success under OOD conditions does not always warrant novel robot behaviors. In-distribution (ID) behaviors can directly be transferred to OOD conditions that share functional similarities with ID conditions. For example, behaviors trained to interact with in-distribution (ID) pens can apply to interacting with a visually-OOD pencil. The key challenge lies in disambiguating which ID observations functionally correspond to the OOD observation for the task at hand. We propose that an expert can provide this OOD-to-ID functional correspondence. Thus, instead of collecting new demonstrations and re-training at every OOD encounter, our method: (1) detects the need for feedback by checking if current observations are OOD and the most similar training observations show divergent behaviors, (2) solicits functional correspondence feedback to disambiguate between those behaviors, and (3) intervenes on the OOD observations with the functionally corresponding ID observations to perform deployment-time generalization. We validate our method across diverse real-world robotic manipulation tasks with a Franka Panda robotic manipulator. Our results show that test-time functional correspondences can improve the generalization of a vision-based diffusion policy to OOD objects and environment conditions with low feedback. Qualitative results can be found on our project page https://adapting-by-analogy.github.io/project-page/

**Keywords:** visuomotor policy, OOD generalization, test-time adaptation

## 1 Introduction

A central goal in robot learning is to enable robots to generalize: to successfully perform tasks in environments they have never seen before. Imagine a robot encountering a pencil for the first time. With just an RGB image, it should be able to reason about the scene and delicately place the object into a nearby cup, as shown in Figure 1. One popular approach towards this is imitation-based visuomotor policy learning. However, while internet-scale datasets have powered generalization breakthroughs in vision and language, robotics still lacks access to the same data scale [1, 2, 3, 4, 5], and collecting expert demonstration data remains expensive and time-consuming. This results in robots failing in unintuitive ways when faced with out-of-distribution (OOD) environments (lower left, Figure 1). Nevertheless, even with modest expert demonstration datasets, recent advances in policy architectures and training algorithms have enabled robots to learn complex visuomotor skills—such as grasping thin tools or folding clothes and operating articulated objects—that work

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

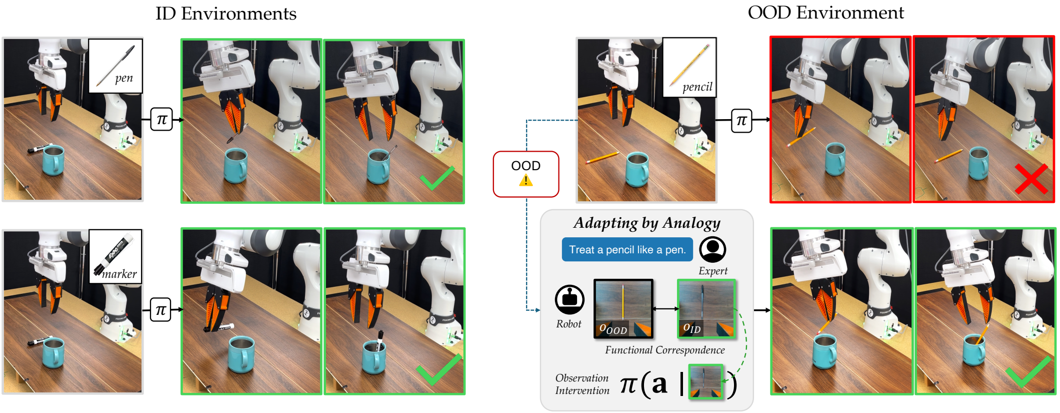


Figure 1: We present ***Adapting by Analogy***, a test-time method that uses functional correspondences between deployment and training conditions to improve a policy's performance in OOD conditions.

well in-distribution (ID) [6, 7, 8, 9, 10]. This raises the central question of our work: *How can we reuse robot behaviors learned in in-distribution settings to succeed in out-of-distribution scenarios?*

Our key insight is that behavior generalization may not always require more demonstration data: it may just need a better correspondence between the training and test conditions. For example, in Figure 1, even though the robot has never seen pencils before, it has seen similarly-thin pens and thicker markers (top row). Thus, if it understood that the pencil is functionally equivalent to the pen in this task, it could "imagine" that the pencil is a pen and reuse the pen pickup behavior to successfully complete the task. Based on this insight, we present ***Adapting by Analogy (ABA)***: a method which establishes *functional correspondences* between in-distribution and out-of-distribution scenes to steer a visuomotor policy through OOD conditions. A key aspect of our method is to leverage expert human knowledge—in the form of a textual description—to interactively learn high-level functional correspondences relevant to the task at hand. The textual description is decoded into a functional correspondence feature space that matches corresponding semantic segments of the scene to retrieve ID behaviors that are "relevant" for the current OOD scene. To measure whether the functional correspondence is well-specified, the robot estimates its uncertainty over the retrieved behavior modes and continues to ask for correspondence refinement until it is certain in the mapping.

We instantiate ***Adapting by Analogy*** on hardware with a Franka Research 3 manipulator acting with a diffusion-based visuomotor policy [6]. By controlling the training and test environments, we study i) how functional correspondences can improve the task success rate in increasingly OOD environments, ii) if our method seeks expert feedback efficiently, and iii) we verify how critical functional correspondences are for OOD generalization. We find that even a relatively small number of expert-guided functional correspondences can significantly improve the generalization capabilities of a visuomotor policy interacting with OOD objects from new semantic categories.

## 2 Related Works

**Test-time Policy Interventions.** Runtime policy interventions are a policy failure mitigation, where-in the policy's execution is intervened and new knowledge is supplied in-order to help mitigate the failure. For instance, a line of work directly proposes interventions on the policy's behavior space, steering the policy into desired modes either through human feedback [11], through Q functions optimized on large scale offline datasets [12], or through predictive modeling [13, 14]. Another line of work proposes intervention directly on the policy's observations, with synthesized observations to remedy known causes of failure [15]. Our work also proposes policy observations with functionally similar ID observations to generalize to novel out-of-distribution conditions.

**Functional Correspondence for Behavior Transfer.** The ability to transfer behaviors from one set of objects to an unseen set of objects hold the potential to unlock robot generalization in the wild.

This problem has been studied through functional correspondences [16]. Prior work have leveraged functional correspondences to directly transfer behaviors across objects from a single demonstration to novel objects in a one shot manner, or in a zero shot manner by leveraging an affordance dataset [17, 18, 19]. Here, functional correspondences are typically established through keypoint based reasoning. In our work, we establish functional correspondences through a correspondence description provided by an expert. Furthermore, instead of directly adapting retrieved behavior, we intervene using the functionally similar training observation

## 3 Problem Formulation

**Environment, Observation, & Action Models.** We model the robot's environment $E \in \mathbb{E}$ as broadly consisting of factors external to the robot such as the objects in the scene, the background, camera configurations, etc. In a particular environment $E$, the robot senses its proprioceptive states $q \in \mathcal{Q}$ (e.g., end-effector pose, gripper state) and uses a sensor $\sigma : \mathcal{Q} \times E \to \mathcal{I}$ to obtain high-dimensional RGB image observations of the scene. At any time $t$, let the stacked image-proprioception observations be, $o_t \in \mathcal{O} := \mathcal{I} \times \mathcal{Q}$. Finally, let $a \in \mathcal{A}$ be the robot's action (e.g., end-effector positions and rotations and gripper action).

**Training Data** The training dataset contains observation-action tuples, $\mathcal{D}_{\text{ID}} := \{(o_t^i, a_t^i)\}_{i=1}^N$, drawn from a set of $M$ "in distribution" environments: $E_{\text{ID}} := \{E_{\text{ID}}^1, E_{\text{ID}}^2, \dots, E_{\text{ID}}^M\}$. For example, a training distribution of environments could consist of $M$ unique objects and their configurations.

**Visuomotor Policy.** Let the robot's policy be a multimodal imitative action generation model [6, 9] denoted by $\pi(\mathbf{a}_t \mid \mathbf{o}_t)$. Here $\mathbf{a}_t := a_{t:t+T}$ is a $T$-step action plan and $\mathbf{o}_t := o_{t:t-H}$ is an $H$-step history of observations. We assume that the policy network first encodes any observation into a corresponding latent state, $z_t = \mathcal{E}(o_t)$, via an encoder. Let $\mathbf{z}_t = \mathcal{E}(o_{t:t-H})$ be a sequence of latent state embeddings. The policy is pre-trained via an imitation learning loss on the in-distribution dataset of observation-action pairs from $\mathcal{D}_{\text{ID}}$.

**OOD-to-ID Generalization via Functional Correspondances.** Given a visuomotor policy $\pi(\mathbf{a}_t \mid \mathbf{o}_t)$ pre-trained on behaviors from in-distribution environments $E_{\text{ID}}$, we seek to generalize the robot's task performance to *out-of-distribution* (OOD) environments, $E_{\text{OOD}}$. Since the general problem of OOD generalization is an extremely challenging open problem, in this paper we assume that (1) $E_{\text{ID}}$ and $E_{\text{OOD}}$ differ only by the objects present in the scene and background color (but the environment geometry remains the same), (2) the training observations $\mathcal{O}_{\text{ID}}$ and deployment time observations $\mathcal{O}_{\text{OOD}}$ are obtained on the same robot embodiment, and (3) we have access to the training data, $\mathcal{D}_{\text{ID}}$.

Our key idea is to identify *functional correspondences* between the test-time OOD scene—in which the base policy would fail to act correctly—and training-time ID scenes, in which the policy can generate high-quality behaviors. Functional correspondences identify parts of the image observations with similar affordances for the task at hand. Intuitively, learned robot behaviors should be transferable across observations whose affordance maps are aligned, i.e., observations where regions that have similar affordances overlap. Thus, we aim to retrieve ID observations whose functional correspondences are aligned with the test-time OOD observation.

**Problem Formulation: Expert-Guided Functional Correspondences.** The core challenge lies in identifying the functional correspondences across the OOD image observations and the ID image observations. Humans possess the ability to infer object affordances, and generalize them to novel objects. Thus, we propose to leverage experts feedback in the form of natural language to acquire these functional correspondences between the OOD and the ID image observations.

Formally, let the **functional correspondance map** be denoted by $\Phi : \mathcal{I} \times \mathcal{I} \times \mathcal{L} \to \mathcal{P}(\Omega \times \Omega)$. Given two images $i, \hat{i} \in \mathcal{I}$ and a natural language description $l \in \mathcal{L}$ provided by the expert, this mapping returns all pairs of functionally corresponding image segments $(\omega, \hat{\omega})$ where $\omega \in \Omega$ are image segments from image $i$ and $\hat{\omega} \in \hat{\Omega}$ are image segments from image $\hat{i}$. Here, $\mathcal{P}(\Omega \times \Omega)$ is

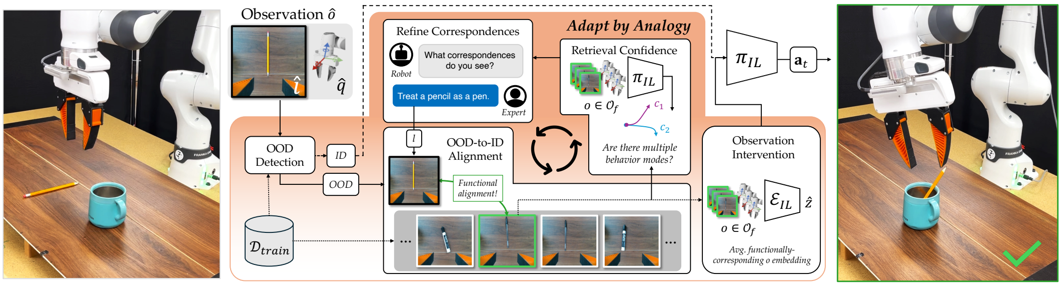


Figure 2: **Adapting by Analogy** consists of four key phases. (left) First, we run a fast OOD detector by checking the cosine similarity between the current observation $\hat{o}$ and the training observations. (center, top-left) Given a correspondence description $l$, we establish OOD-to-ID functional correspondences to retrieve corresponding ID observations (center, bottom). We refine the correspondances with the expert as long as there is ambiguity in the predicted behavior mode (center, top-right). Once finalized, we intervene on the observations and execute the planned actions (right).

the powerset of all paired image segments. Let $K$ be the number of corresponding image segments. Thus, the functional correspondence map is defined as:

$$\Phi(i, \hat{i}, l) := \{(\omega_j, \hat{\omega}_j) \mid j \in \{0, 1, \dots, K\}\} \tag{1}$$

We measure the **functional correspondence alignment** of any two images via $f : \mathcal{P}(\Omega \times \Omega) \to \mathbb{R}$. In this work, we model $f$ as the total Intersection over Union (IoU) between functionally corresponding regions of the images returned by the functional correspondance map, $\Phi$:

$$f(\Phi(i, \hat{i}, l)) = \sum_{i=0}^{K} \text{IoU}(\omega_j, \hat{\omega}_j) \tag{2}$$

Finally, given any OOD observation $\hat{o} = (\hat{q}, \hat{i})$ observed by the robot at deployment time and an expert language input $l$ describing the functional correspondences, we retrieve the ordered set of in-distribution observations $\mathcal{O}_f = (o_1, o_2, \dots, o_k) \subseteq \mathcal{O}_{\text{ID}}$ ranked by their functional alignment from Eq. (2). For intervention, we use the behaviors extracted from the top-$M$ observations in $\mathcal{O}_f$.

## 4 Method: Adapting by Analogy

### 4.1 Detecting Out-of-Distribution Observations

At each timestep, our method first detects if the robot's observations $\hat{o}$ are anomalous via a fast OOD detector. We measure the cosine similarity between the encoded observation $\hat{z} = \mathcal{E}(\hat{o})$ and embeddings of in-distribution observations $z \in \mathcal{E}(o_i)$, $o \in \mathcal{O}_{\text{ID}}$ via $\text{IDScore}(\hat{o}, \mathcal{O}_{\text{ID}}) := \min_{o \in \mathcal{O}_{\text{ID}}} \frac{\mathcal{E}(\hat{o}) \cdot \mathcal{E}(o)}{\|\mathcal{E}(\hat{o})\| \|\mathcal{E}(o)\|}$. If the IDScore is above a threshold $\lambda$, then we deem the observation to be nominal and directly execute the action $\hat{a} \sim \pi(\cdot \mid \hat{o})$. Otherwise, we deem the observation to be OOD, ask the expert for an initial language instruction $l$ describing relevant functional correspondences, and use those to intervene on the observation before action generation.

### 4.2 Establishing OOD-to-ID Functional Correspondences

Given an OOD observation $\hat{o} = (\hat{q}, \hat{i})$ identified via our fast anomaly detector and the expert's language description $l$, we want to intervene on the policy by reusing learned behaviors from functionally similar ID observations. This requires computing the functional alignment from Eq. 2 between $\hat{o}$ and every ID observation $o \in \mathcal{O}_{\text{ID}}$. However, implementing this matching is challenging in practice for two reasons: first, it is computationally expensive (requiring image segmentation and IoU computation over all corresponding segments), and second, matching correspondances between 2D image segments does not directly reveal correspondences in the high-dimensional robot state. Thus, we filter the demonstration dataset $\mathcal{D}_{\text{ID}}$ consisting of $N$ observation-action trajectories $\tau$ to retrieve one observation $o \in \tau$ per trajectory which contain similar proprioceptive states $q$ to the test-time robot state $\hat{q}$. Mathematically, for a distance threshold $\lambda_q \in \mathbb{R}^+$ and the current configuration $\hat{q}$, let the filtered observation dataset $\mathcal{O}_q \subset \mathcal{O}_{\text{ID}}$ be:

$$\mathcal{O}_q = \left\{ o \,\middle|\, o = \arg\min_{(o,a)\in\tau} \big(\|q - \hat{q}\|_2 - \lambda_q\big), \quad \forall \tau \in \mathcal{D}_{\text{ID}} \right\} \tag{3}$$

Using this filtered dataset, we can now compute our **functional correspondence map** from Eq. 1 via two internal models: one which converts the expert's language feedback $l \in \mathcal{L}$ into a functional feature set (denoted by $\phi_l$) and another which semantically segments each ID image $i \in \mathcal{O}_q$ and the current OOD image observations $\hat{i}$ to generate the set of image masks and semantic labels denoted by $\hat{\Omega}$ and $\Omega$ respectively. We use Grounded Segment Anything [20] for semantic segmentation.

Next, the expert's language input $l$ is decoded into a set of correspondence features $\phi_l$ that can be applied to the semantic segmentations $\Omega, \hat{\Omega}$ to return a set of $K$ functionally corresponding image segments $(\omega^j, \hat{\omega}^j), j \in \{0, \ldots, K\}$. For example, the $\phi_l$ that is decoded from $l =$*"Match pencils with pens"* lifts pixels corresponding to the segmentation label *'pencil'* in the OOD image, and pairs it with pixels corresponding to the label *'pen'* in the ID images. In this work, we use a templated $l$, but future work could explore the use of LLMs as an interface.

Ultimately, after $\Phi$ extracts the set of functionally corresponding image segments, we measure their alignment using Eq. 2. Each ID observation $o \in \mathcal{O}_q$ is ranked based on its functional alignment with the OOD observation $\hat{o}$ to obtain the ordered set of functionally corresponding ID observations $\mathcal{O}_f \subseteq \mathcal{O}_q$ used during intervention (Sec. 4.4).

### 4.3 Refining Functional Correspondences Until Confident

Thus far, we have assumed that the initial expert description $l$ of functional correspondences was sufficient for the entire task. However, correspondences may evolve during task execution. For example, consider the task of picking up trash and sorting it into organic and recycling. The functional correspondence between types of trash (organic and recycling) does not matter initially when the robot is planning a grasp, but becomes relevant once the item has been picked up and needs to be sorted. Thus, our method interactively refines the functional correspondence description until the robot is confident in the behavior it has retrieved.

Intuitively, a well-established functional correspondence will reduce the diversity in robot action plans, focusing on the "correct" behavior mode. To quantify the relevant behavior modes before functional alignment, we obtain a set of action plans $\mathcal{A}_q := \{\mathbf{a} \sim \pi(\cdot \mid o) \mid o \in \mathcal{Q}_q\}$ for all observations with the same proprioceptive state via a forward pass through the policy. Behavior mode labels are obtained by fitting $n_c$ clusters to $\mathcal{A}_q$ via K-means clustering. Since the current functionally-aligned observations are a subset $\mathcal{Q}_f \subseteq \mathcal{Q}_q$, we can obtain labels for all functionally-aligned *action plans* $\mathcal{A}_f \subseteq \mathcal{A}_q$ and measure the reduction in behavior modes via the entropy over the action plan labels. As long as the entropy in the retrieved actions is high, the robot keeps asking the expert to refine their functional correspondence description $l$ by showing them the current observations and their behaviors, then re-doing the OOD-to-ID matching from Sec. 4.2.

### 4.4 Intervening on Observations to Generate Functionally-Corresponding Behavior

Once the correspondence description $l$ is complete and the retrieved action mode uncertainty is sufficiently low, the robot intervenes on its observations to generate functionally "correct" behavior. Specifically, observations in the final refined $\mathcal{O}_f$ are ranked based on their functional alignment as measured by Eq. 2. To smooth out action prediction, we generate the final executed action plan by interpolating the embeddings of $M$-highest ranked corresponding observations: $\hat{z} := \frac{1}{M}\sum_{o\in\mathcal{O}_f} \mathcal{E}(o)$ before passing the average embedding to the policy network.

## 5 Hardware Experiments

We conduct a series of experiments in robot hardware to study: (1) How much does *Adapting by Analogy* improve the visuomotor policy's closed-loop performance on OOD environments induced by novel objects and backgrounds conditions?, (2) What kind of features (e.g., base policy's embedding, DINOv2 [21], or functional correspondences) maximally help observation interventions?, (3) How efficient is our method at seeking expert feedback for adaptation in OOD environments?, (4) When intervention schemes succeed, are they retrieving functionally-aligned observations?

**Real Robot Setup.** We use a Franka Research 3 robotic manipulator equipped with a 3D printed UMI gripper [22] for our real-world experiments. The RGB image observations $i \in \mathcal{I}$ come from a wrist mounted RealSense D435 camera and a third-person Zed mini 2i camera overlooking the workspace. The overall robot observation $o := (i, q)$ consists of the concatenated images and the robot proprioception. More details about our setup can be found in the supplementary.

**ID Environments & Tasks.** We train two visuomotor policies on two different real-world manipulation tasks. The first task is **sweep-trash**, wherein robot must sweep trash towards different goals, based on whether the trash is organic and recycling. The next task is **object-in-cup**, where-in a robot arm is tasked with picking up a object such as a marker or a pen and dropping it in a mug. Pens—which are grasped above their center-of-mass—need to be dropped into the mug from the bottom, and markers—which are grasped below their center-of-mass—need to be dropped from the front. We divide the task in 3 sub-goals (A) grasping the object, (B) picking the correct behavior mode based on the grasp, and (C) dropping object into the cup.

**Visuomotor Policy Training.** We use a diffusion policy [6] as the base visuomotor policy $\pi(\mathbf{a} \mid \mathbf{o})$. It takes as input $o$ and predicts a T-step action plan, where $T = 16$. For **sweep-trash**, the training dataset $|\mathcal{D}_{\text{ID}}| = 100$ consists of 50 demonstrations cleaning up crumpled paper (recycling trash) and 50 demonstrations cleaning up M&Ms (organic trash). For **object-in-cup**, the policy is trained on $|\mathcal{D}_{\text{ID}}| = 200$ demonstrations with 100 placing a marker (dropped from the back) and 100 placing a pen (dropped from the top).

**OOD Environments.** We test on two in-distribution environments and five OOD environments for each task. In addition to pens and markers, we e evaluate sweep trash with one background variation (workspace covered with black cloth), and three novel instances of trash (doritos, crumpled napkin, thumb tacks). For the object-in-cup, we test with in-distribution objects, one novel background (workspace covered with black cloth), and three instances of novel objects varying in shapes and sizes (pencil, battery, jenga block).

**Baselines.** We compare our method, **ABA**, with three baselines. **Vanilla** is the base visuomotor policy without any intervention mechanism. **PolicyEmbed** intervenes on the observations with a similar mechanism to ours, but it retrieves ID observations using cosine similarity in the base policy's learned embedding space, $\mathcal{E}(o) \in \mathcal{Z}$. It does not use any expert feedback. **DINOEmbed** also intervenes on the base policy, but it retrieves ID samples using cosine similarity in the DinoV2 [21] feature space of the OOD and ID observations. We use this to test if powerful pre-trained vision foundation models can implicitly capture functional correspondences beyond semantic object categories. We use both the class token features and the patch features. For **ABA**, we generate correspondence features $\phi$ via decoding $l$ into a pre-templated set of features. The choices of the features are (1) match *'ood object semantic label'* with *'id object semantic label'*, (2) overlap segments of *'ood object'* with *'id object'* (3) Align left/right edge of segments, (4) align top/base of segments, and (5) "Pass", which meant that the expert does not want to refine the set of correspondence features. All intervention methods perform matching in the refined set of ID observations based on the robot's current proprioception $\mathcal{O}_q$, as described in Sec. 4.2.

**Evaluation Procedure.** All methods are evaluated via the same procedure and in the same conditions. For each ID and OOD environmental condition, we perform 10 rollouts of each method,

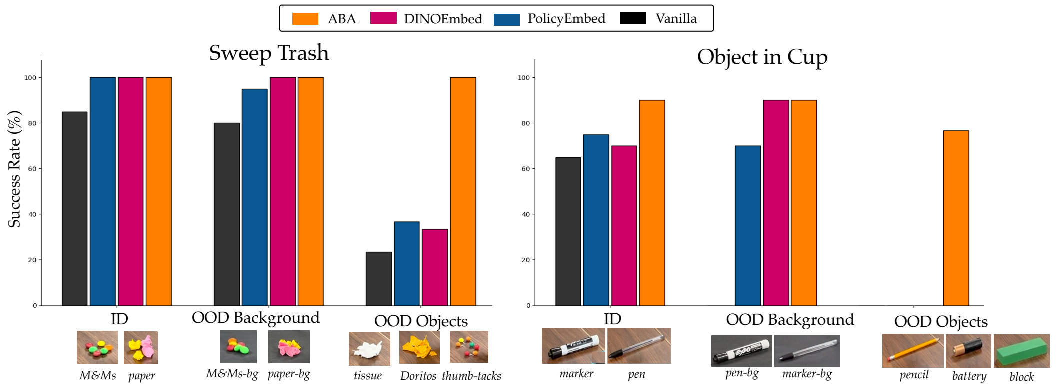


Figure 3: **Task Success in ID and OOD Environments.** We report the task success rate averaged across 10 rollouts (per each ID and OOD conditions) and averaged across ID, OOD background, or OOD object conditions. For both the sweep-trash and the object-in-cup tasks, we see that **ABA** consistently achieves the highest task success rate compared to baselines.

placing the object of interest uniformly at random within a 15 cm horizontal range on the table. With a total of 14 environmental conditions, we collect a total of 140 rollouts for evaluation.

### 5.1 How much does ABA improve the policy's closed-loop performance?

In this section, we compare the overall task success rate of **ABA** with **Vanilla**. As shown in Fig. 3, **ABA** improves the **Vanilla** policy even in in-distribution environments by 15% on the sweep-trash task and 25% on the object-in-cup task. While the vanilla policy was robust to the novel background for the sweep-trash task, it completely degraded when faced with the novel background in the more challenging object-in-cup task. By reasoning about functional correspondences, **ABA** improved over the vanilla policy by 20% on the sweep trash task and by 90% on the object in cup task, staying robust to task-irrelevant changes to the background. Finally, we observe strong OOD generalization with **ABA** when evaluated under OOD objects where it improves over the vanilla policy by 76% on both tasks, showcasing that learned behaviors can be transferred to OOD objects from different semantic categories by reasoning about functional correspondences.

### 5.2 What kind of features maximally help observation interventions?

In this section, we compare how the features used for retrieving ID observations affect policy performance. For in-distribution environments, on the sweep trash task both **PolicyEmbed** and **DINOEmbed** perform on par with **ABA**. However, **ABA** outperforms by 15% on the object in cup task. Similar to **ABA**, both the **DINOEmbed** and **PolicyEmbed** are also robust to the novel background. Interestingly, **DINOEmbed**'s performance *improves* under the novel background. We hypothesize that this is due to the exceptional capabilities of the dino features at dense correspondence matching across objects within the same semantic category [21]. When tested on OOD objects, both **DINOEmbed** and **PolicyEmbed** struggle, achieving only 36.67% and 33.34% success rate respectively on sweep trash. On the object in cup task both baselines failed to successfully complete the task. Taking a closer look at the performance with specific OOD objects revealed common failures at the grasping stage and at picking the correct behavior mode. More analysis in supplementary.

### 5.3 How efficient is ABA at seeking expert feedback in OOD environments?

Next we study how often does **ABA** request feedback from the expert at test-time. Fig. 4 shows the number of times **ABA** requested feedback on average across 10 rollouts (with standard error bars) for both the sweep trash and the object in cup task, in all the three experiment settings (ID, OOD-Bg, OOD-Object).

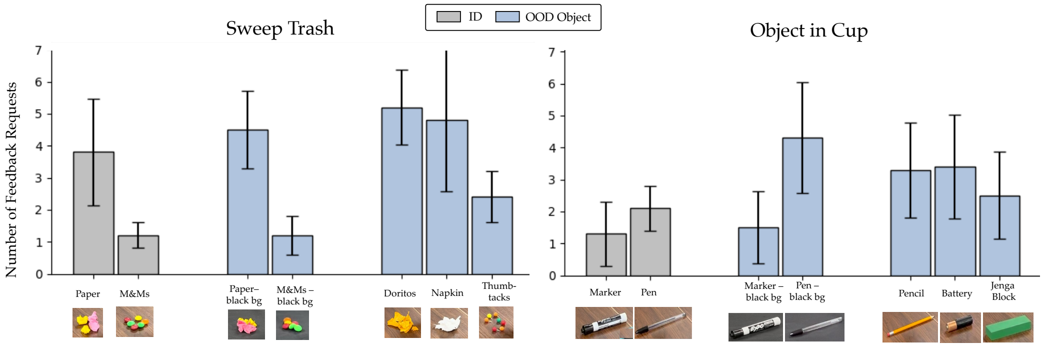


Figure 4: **Expert Feedback Requested by ABA.** We show mean and standard error for the number of feedback requests across 10 rollouts per each environment. We find that ABA infrequently queries the expert for correspondances, given that sweep-trash has 70 timesteps and object-in-cup has 120.

For the **sweep-trash** task, ABA asks for feedback $3.8 \pm 1.66$ times for ID crumpled paper. In OOD backgrounds,feedback requests increased, e.g., to $4.5 \pm 1.2$ times per rollout for crumpled paper. ABA requested feedback the most for OOD objects, with the highest number of requests for doritos with an average of $5.2 \pm 1.16$ times per rollout. Note that each rollout for sweep trash ran for 80 timesteps, so this corresponds to asking for feedback  6% of the rollout. For the **object-in-cup** task, feedback was requested $2.1 \pm 0.7$ times for the ID pen, and $1.3 \pm 1.0$ times for the ID marker. Similar to sweep-trash, the feedback requests increased with OOD backgrounds: e.g., feedback about the pend was requested $4.3 \pm 1.73$ times per rollout. Finally, amongst the OOD objects, feedback was requested the most for battery at $3.4 \pm 1.62$ times per rollout. Note that each rollout in the object in cup task ran for 120 timesteps.

### 5.4 When intervention methods succeed, do they retrieve functionally-aligned observations?

We looked at what observations the intervention-based baselines retrieved when they succeed. We measured the precision of the set of observations retrieved by **DINOEmbed** and **PolicyEmbed** against the observations retrieved by ABA.

Fig. 5 shows precision vs. cumulative success rate, where each dot on the plot is a rollout[1] from the **object-in-cup** task in ID and OOD background environment configurations. These had sufficient coverage over successes and failures. Overall, the trends indicate that when the baselines were successful, they also retrieved functionally corresponding observations with high precision. This shows that interventions using functionally corresponding observations help generalize under OOD conditions.

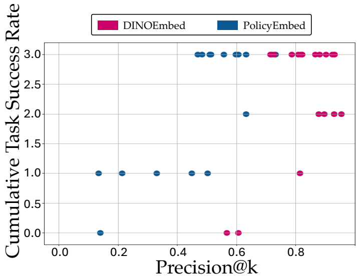


Figure 5: Retrieval overlap vs task success.

## 6 Conclusion

In this work, we present *Adapting by Analogy*, a method for enabling deployment-time generalization of visuomotor policies by leveraging functional correspondences between out-of-distribution (OOD) and in-distribution (ID) observations. Rather than requiring new expert demonstrations for each novel scenario, our approach uses expert-provided functional features—which are interactively refined to represent during task execution—to repurpose existing ID policy behaviors in OOD environments. Empirical results across two real-world manipulation tasks with ten OOD environments demonstrate that establishing functional correspondences can improve a diffusion policy's success rate by  76% to new objects and backgrounds with minimal human intervention.

[1] The final precision value is averaged over each timestep where retrieval happened during the rollout.

## 7 Limitations

Our method is not without its limitations.

**Assumes functional overlap:** Our method assumes that there exists a functional behavior overlap between the OOD scenarios and ID scenarios, enabling the learned behaviors from the ID scenarios to be reused. However, this may not hold in tasks where new objects or environment geometries require fundamentally new strategies that go beyond what was seen in training. Future work should rigorously quantify the robot's *confidence* in retrieving a behavior that is relevant and actively asking for expert demonstrations when no behaviors are relevant.

**Reliance on experts:** Our method does still rely on an expert to provide the functional correspondences at test time, which can be challenging for novice end-users and limit the autonomy of the robot. Future work should study the autonomous identification of correspondence features (e.g., via another foundation model). Relatedly, the decoding of the language description into the functional feature set can be ambiguous, prompting the need for future work on grounding natural language into embodied representations.

**Limited experiment scope:** We tested our method on two types of OOD conditions for two real world tasks. Future work, should incorporate more OOD scenarios, for ex. lighting perturbations, distractor objects, on more complex, fine-grained and long-horizon tasks.

**Uncalibrated OOD detection:** Finally, our system's performance requires a reliable OOD detection mechanism, which should be properly calibrated to balance how frequently the robot has to do interventions and ask for features from the expert.

### Acknowledgments

The authors thank members of the Intent Lab and Harp Lab at the Robotics Institute at CMU for their helpful discussions and feedback. This material is based on work that is partially funded by an unrestricted gift from Google. Andrea Bajcsy is supported in part by NSF CAREER Award No. 2441014 and a Google Research Scholar Award.

## References

[1] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[3] A. W. C. contributors. Agibot world colosseum. https://github.com/OpenDriveLab/AgiBot-World, 2024.

[4] H.-S. Fang, H. Fang, Z. Tang, J. Liu, J. Wang, H. Zhu, and C. Lu. Rh20t: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023.

[5] O. X.-E. Team. Open X-Embodiment: Robotic learning datasets and RT-X models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[6] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[7] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023. URL `https://arxiv.org/abs/2304.13705`.

[8] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn. Openvla: An open-source vision-language-action model. *Conference on robot learning (CoRL)*, 2024.

[9] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto. Behavior generation with latent actions. In *Forty-first International Conference on Machine Learning*, 2024.

[10] M. Reuss, Ö. E. Yağmurlu, F. Wenzel, and R. Lioutikov. Multimodal diffusion transformer: Learning versatile behavior from multimodal goals. In *Robotics: Science and Systems*, 2024.

[11] Y. Wang, L. Wang, Y. Du, B. Sundaralingam, X. Yang, Y.-W. Chao, C. Perez-D'Arpino, D. Fox, and J. Shah. Inference-time policy steering through human interactions. *arXiv preprint arXiv:2411.16627*, 2024.

[12] M. Nakamoto, O. Mees, A. Kumar, and S. Levine. Steering your generalists: Improving robotic foundation models via value guidance. *Conference on Robot Learning (CoRL)*, 2024.

[13] Y. Wu, R. Tian, G. Swamy, and A. Bajcsy. From foresight to forethought: Vlm-in-the-loop policy steering via latent alignment. *arXiv preprint arXiv:2502.01828*, 2025.

[14] H. Qi, H. Yin, Y. Du, and H. Yang. Strengthening generative robot policies through predictive world modeling. *arXiv preprint arXiv:2502.00622*, 2025.

[15] A. J. Hancock, A. Z. Ren, and A. Majumdar. Run-time observation interventions make vision-language-action models more visually robust. *arXiv preprint arXiv:2410.01971*, 2024.

[16] Z. Lai, S. Purushwalkam, and A. Gupta. The functional correspondence problem. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15772–15781, 2021.

[17] C. Tang, A. Xiao, Y. Deng, T. Hu, W. Dong, H. Zhang, D. Hsu, and H. Zhang. Functo: Function-centric one-shot imitation learning for tool manipulation. *arXiv preprint arXiv:2502.11744*, 2025.

[18] Y. Liu, J. Mao, J. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling. One-shot manipulation strategy learning by making contact analogies. *arXiv preprint arXiv:2411.09627*, 2024.

[19] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu. Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation. In *European Conference on Computer Vision*, pages 222–239. Springer, 2024.

[20] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[21] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. HAZIZA, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023.

[22] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

## Supplementary

This is the supplementary material to the paper, Adapting by Analogy: OOD Generalization of Visuomotor Policies via Functional Correspondence.

## A Hardware Experiment Setup

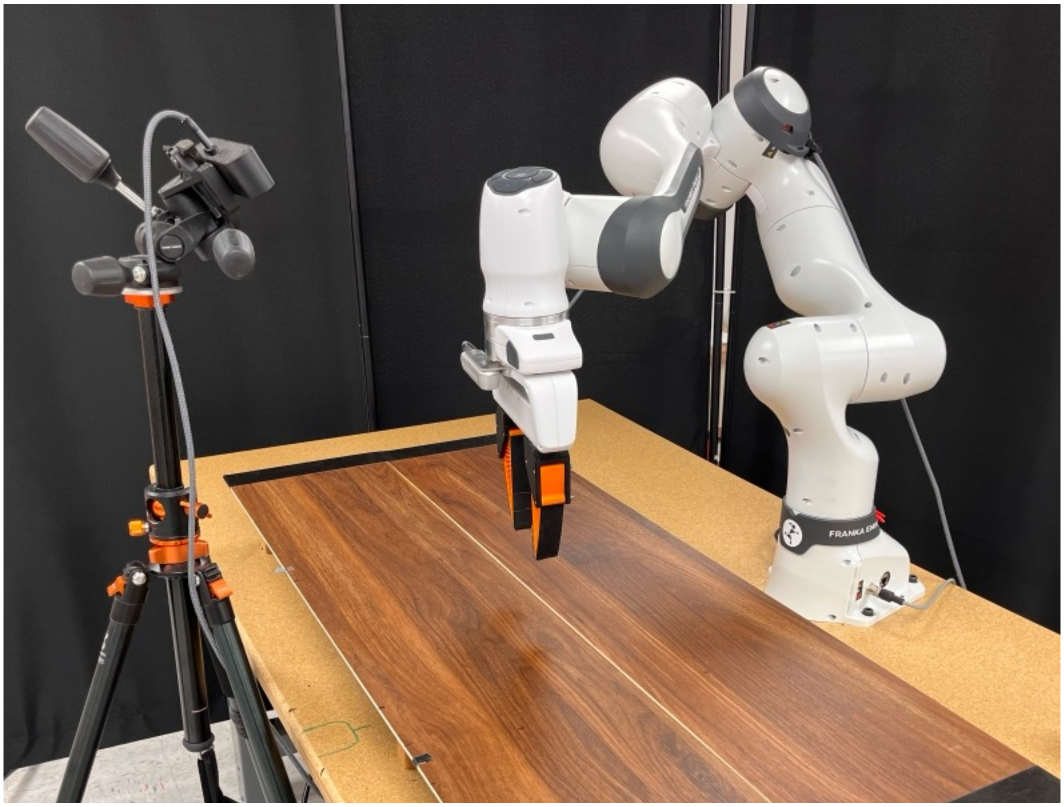
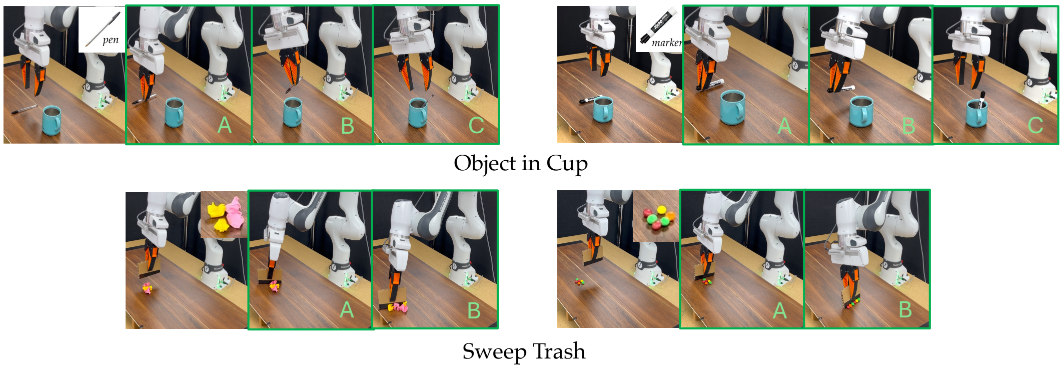


Figure 6: Our hardware experiment setup, we use a Franka Research 3 robot, with a UMI gripper. The RealSense D435 wrist camera, and Zed mini 2i third person camera are placed as shown.

## B Tasks



Object in Cup

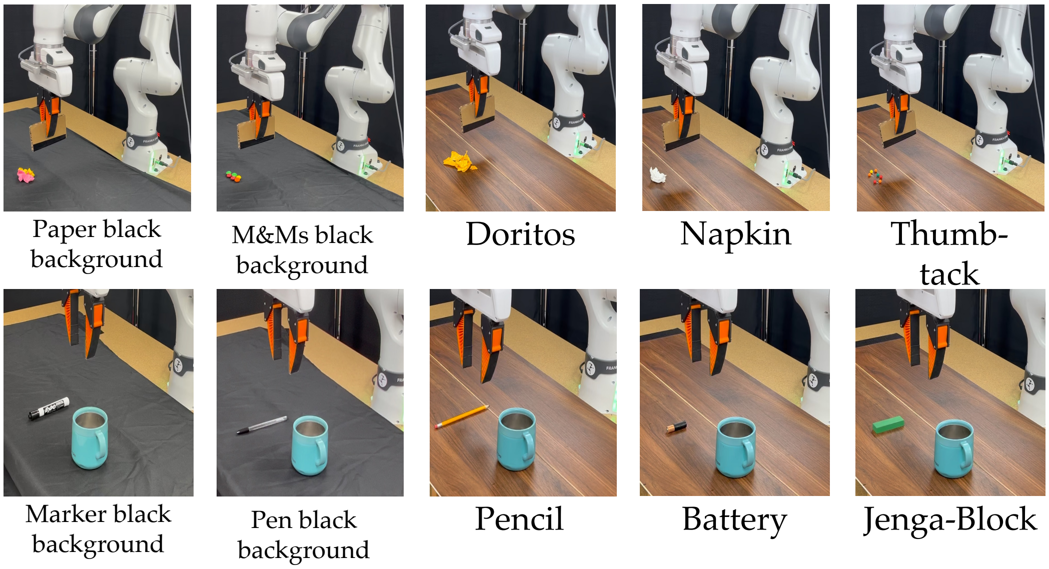
Sweep Trash

Figure 7: The training demonstrations for our two tasks, with their sub-goals(A, B, C). For the object in cup task, the pen is grasped below the center-of-mass, and is dropped into the mug from the front. The marker is grasped above the center-of-mass and is dropped into the mug from the bottom. For the sweep trash task, paper (i.e., recycling) is swept up, and M$Ms (i.e., organic) is swept down.

We conduct our experiments on two real-world tasks. The first task is **sweep-trash**, wherein robot must sweep trash towards different goals, based on whether the trash is organic and recycling. For evaluation, we divide the task in two sub-goals (A) properly aligning the wiper with the trash, (B) sweeping to the correct location. The next task is **object-in-cup**, where-in a robot arm is tasked with picking up a object such as a marker or a pen and dropping it in a mug. Markers—which are grasped above their center-of-mass—need to be dropped into the mug from the bottom, and pens—which are grasped below their center-of-mass—need to be dropped from the front. We divide the task in 3 sub-goals (A) grasping the object, (B) picking the correct behavior mode based on the grasp, and (C) dropping object into the cup. Fig. 7 demonstrates the various modes and the sub-goals for the two tasks.

We evaluate both tasks on the in-distribution conditions and OOD conditions induced by background and novel objects. The OOD environments are shown in Fig. 8 For **sweep-trash** our ID environments are $E_{\text{ID-trash}} := \{E_{\text{ID}}^{paper}, E_{\text{ID}}^{M\&Ms}\}$. The OOD environments are $E_{\text{OOD-Trash}} := \{E_{\text{OOD}}^{\text{doritos}}, E_{\text{OOD}}^{\text{napkin}}, E_{\text{OOD}}^{\text{thumb-tack}}, E_{\text{OOD}}^{\text{paper-bg}}, E_{\text{OOD}}^{\text{M\&M-bg}}\}$.

For **object-in-cup** our ID environments are $E_{\text{ID-object}} := \{E_{\text{ID}}^{marker}, E_{\text{ID}}^{pen}\}$. The OOD environments are $E_{\text{OOD-object}} := \{E_{\text{OOD}}^{\text{pencil}}, E_{\text{OOD}}^{\text{battery}}, E_{\text{OOD}}^{\text{block}}, E_{\text{OOD}}^{\text{marker-bg}}, E_{\text{OOD}}^{\text{pen-bg}}\}$.



Paper black background

M&Ms black background

Doritos

Napkin

Thumb-tack

Marker black background

Pen black background

Pencil

Battery

Jenga-Block

Figure 8: Our OOD environments for both the sweep-trash and object-in-cup task

## C Additional Results

### C.1 How much does ABA improve the policy's sub-goal level closed-loop performance?

Fig. 9 shows that on the sweep-trash task, both **ABA** and **Vanilla** are able to successfully accomplish both subgoals on $E_{\text{ID}}^{M\&Ms}$. However, in $E_{\text{ID}}^{paper}$, **Vanilla** fails at aligning the wiper with the paper trash (subgoal A) $10\%$ of the times, and sweeps paper incorrectly (subgoal B) $30\%$ of the times. **ABA** maintains $100\%$ performance on $E_{\text{ID}}^{paper}$.

Showing a similar trend, both **Vanilla** and **ABA** show $100\%$ success rate on the $E_{\text{ID}}^{marker}$ for the object-in-cup task. On the $E_{\text{ID}}^{pen}$, while both **Vanilla** and **ABA** are able to grasp the pen (subgoal A) $100\%$ of the times, **ABA** improves over **Vanilla** by $40\%$ at picking the right mode (subgoal B), showing a $100\%$ success rate. Finally, since dropping the pen into the cup (subgoal C) is the most fine-grained aspect of the task, both **ABA** and **Vanilla** struggle but **ABA** still improves over **Vanilla** by $50\%$.

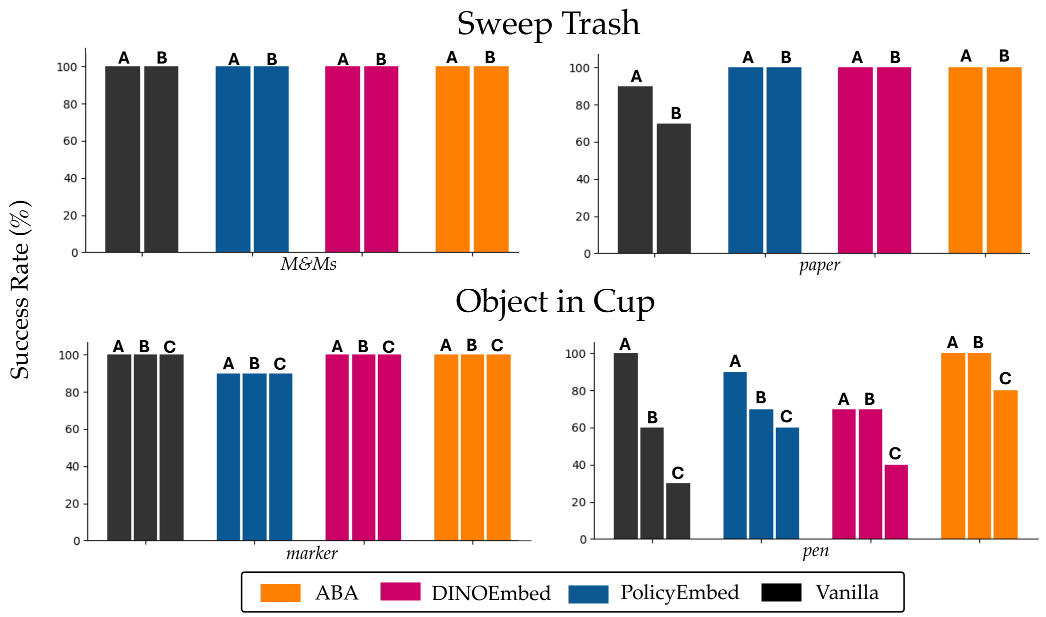


Figure 9: Subgoal Success in each ID Environment. We report the subgoal level task success rate averaged across 10 rollouts. For both the sweep-trash and the object-in-cup tasks, we see that **ABA** consistently achieves the highest task success rate compared to baselines.

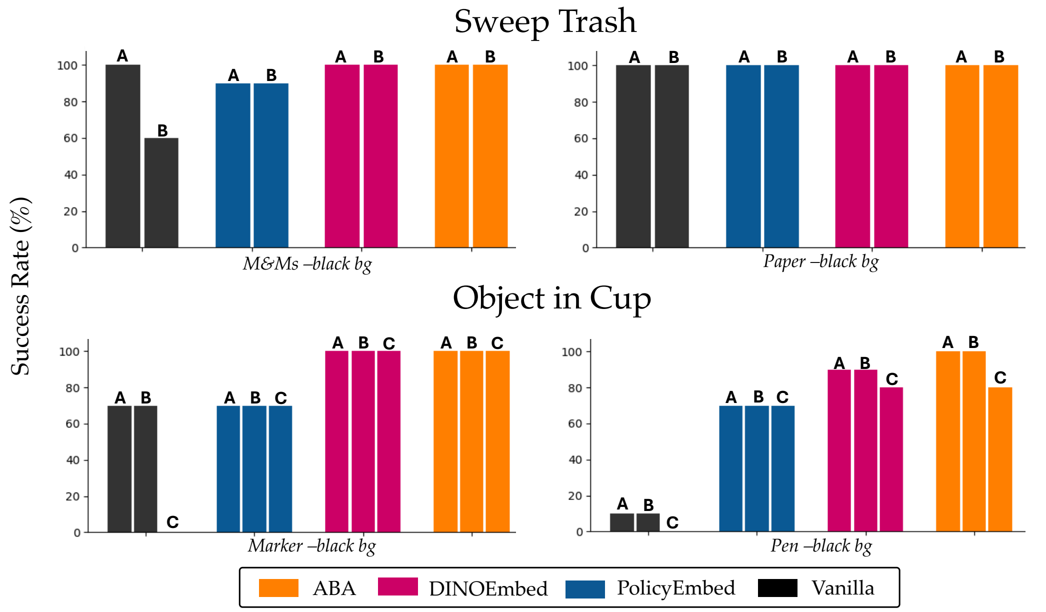


Figure 10: Subgoal Success in each OOD Environment, induced by changing the background. The success rate is averaged across 10 rollouts. **ABA** again consistently achieves the highest task success rate compared to baselines.

Next, we compare **ABA** and **Vanilla** across OOD environments, induced using a novel background ($E_{\text{OOD}}^{\text{paper-bg}}$, $E_{\text{OOD}}^{\text{M\&M-bg}}$, $E_{\text{OOD}}^{\text{pen-bg}}$, $E_{\text{OOD}}^{\text{marker-bg}}$).

Fig. 10 shows that on the sweep-trash task the performance trends are similar to the ID environments, although interestingly instead of $E_{\text{OOD}}^{paper-bg}$, **Vanilla** now shows poorer performance on subgoal B of $E_{\text{OOD}}^{\text{M\&Ms-bg}}$. **ABA** shows 100% success rate on both goals of both $E_{\text{OOD}}^{\text{M\&Ms-bg}}$ and $E_{\text{OOD}}^{\text{paper-bg}}$. On the object-in-cup task, **Vanilla** struggles on all subgoals of both $E_{\text{OOD}}^{\text{pen-bg}}$, $E_{\text{OOD}}^{\text{marker-bg}}$ environments. **ABA**

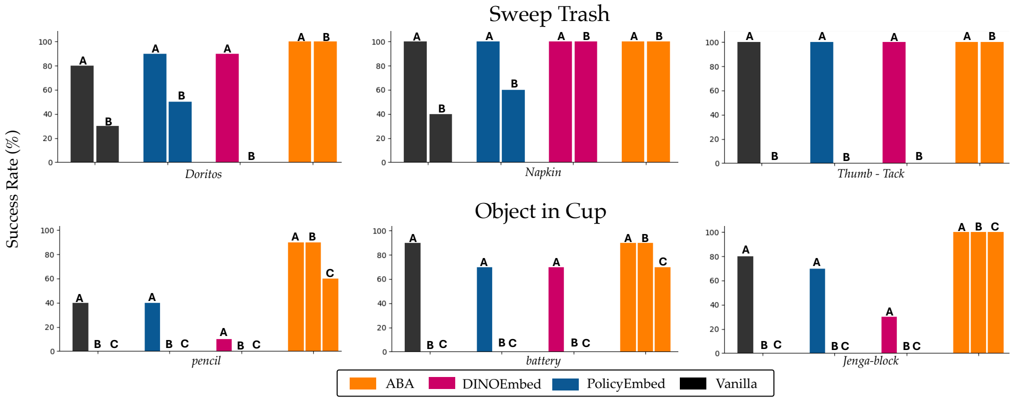


Figure 11: Subgoal Success in each OOD Environment with 3 novel objects for both sweep-trash and object-in-cup task. The success rate is averaged across 10 rollouts. **ABA** again consistently achieves the highest task success rate compared to baselines.

improves over **Vanilla** on both environments showing a 100% performance on all subgoals of both environments, except subgoal C of the $E_{\text{OOD}}^{\text{pen-bg}}$, where it shows an 80% success rate. This shows that a with novel background, **Vanilla** fails to even grasp the objects, however interventions with ID observations ignores the OOD conditions induced by the novel background, allowing **ABA** to uphold closed loop performance under the OOD environments.

Finally, on OOD environments induced by novel object categories for the sweep-trash task we observe from Fig. 11 that while **Vanilla** is able to align the wiper with the trash, it fails to pick the correct direction for sweeping the trash (subgoal B), as visual features are not enough to decide whether the trash is organic or recycling. **ABA** is able to successfully accomplish both subgoals for all novel objects as the relevant features for deciding the trash type are supplied by the expert as functional correspondences.

Since the object-in-cup task is more challenging, **Vanilla** is only performant at grasping (subgoal A). It is able to grasp the pencil with 40%, the battery with 90%, and the jenga-block with 80% success-rate. However, the sizes of the objects are such that they can only be dropped into the mug from the top (subgoal B), however **Vanilla** is not able to infer these features solely from the training data and hence fails at subgoal B and C. With **ABA**, the expert language feedback helps establish the correct functional correspondences, leading to an improvement in the performance across all subgoals.

### C.2 What kind of features maximally improve the sub-goal level performance for observation interventions based methods?

As shown in Fig. 9, all intervention based method demonstrate a 100% task success on all subgoals of the sweep-trash task, in the ID environments. On the object-in-cup task intervention based methods again perform comparably on the $E_{\text{ID}}^{marker}$, however on the $E_{\text{ID}}^{pen}$ both **PolicyEmbed** and **DINOEmbed** perform worse as compared to **ABA** on all subgoals.

As shown in Fig. 10, under a novel background, intervention based methods perform comparably on the sweep-trash task. On the object-in-cup task, **PolicyEmbed** performs worse compared to both **DINOEmbed** and **ABA**, whereas **DINOEmbed** performs comparably with **ABA**. This can be attributed to the ability of dino features to perform dense correspondence matching, specially across objects in the same semantic class.

Fig. 11 shows that under novel objects both **PolicyEmbed** and **DINOEmbed** struggle. Since **PolicyEmbed** relies on the policy embeddings, under 'doritos' and 'napkin' it sweeps them in either direction. **DINOEmbed** matches the visual features and since napkin closely resembles paper, it is able to correctly sweep napkin as recycling, and fails on other objects.

For the object-in-cup task, because policy embeddings and visual features alone are not enough to match the objects with the ID sample that lead to the desired behavior mode, both **PolicyEmbed** and **DINOEmbed** perform worse as compared to **ABA**.