IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Augmented Reality for RObots (ARRO): Pointing Visuomotor Policies Towards Visual Robustness

Reihaneh Mirjalili*[1], Tobias Jülg*[1], Florian Walter[1], Wolfram Burgard[1]

***Abstract*— Visuomotor policies trained on human expert demonstrations have recently shown strong performance across a wide range of robotic manipulation tasks. However, these policies remain highly sensitive to domain shifts stemming from background or robot embodiment changes, which limits their generalization capabilities. In this paper, we present ARRO, a novel visual representation that leverages zero-shot open-vocabulary segmentation and object detection models to efficiently mask out task-irrelevant regions of the scene in real time without requiring additional training, modeling of the setup, or camera calibration. By filtering visual distractors and overlaying virtual guides during both training and inference, ARRO improves robustness to scene variations and reduces the need for additional data collection. We extensively evaluate ARRO with Diffusion Policy on a range of tabletop manipulation tasks in both simulation and real-world environments, and further demonstrate its compatibility and effectiveness with generalist robot policies, such as Octo, OpenVLA and π_0. Across all settings in our evaluation, ARRO yields consistent performance gains, allows for selective masking to choose between different objects, and shows robustness even to challenging segmentation conditions. Videos showcasing our results are available at: augmented-reality-for-robots.github.io**

I. Introduction

Visuomotor policy learning in robotics has recently benefited significantly from advances in generative modeling [1], [2], [3]. State-of-the-art methods for imitation learning from human expert demonstrations show strong performance in both tabletop and mobile manipulation tasks [4], [5], [6], which has motivated the collection of a vast range of robotics datasets [7], [8], [9], [10], [11], [12] and concerted efforts to curate and share them [13]. This has enabled the development and training of large-scale visuomotor policies that are commonly referred to as robotics foundation models and cover a wide range of tasks, robots, and environments [13], [14], [15], [16], [17], [18].

Ideally, visuomotor policies should be robust to visual environment changes and agnostic to the robot's appearance. However, despite the generalization capabilities that some of these policies achieve in specific scenarios, generalization under domain shift remains a challenge. Even subtle visual changes—such as background variation, distractor objects, or differences in the appearance of the robot—that occur when a policy is deployed in conditions different from those seen during training can lead to notable performance degradation [19], [20]. This severely limits the applicability of such policies in real-world environments, where visual variability is inevitable. Although increasing the diversity of the training data can partially mitigate these issues, such strategies are costly and often fail to cover the full spectrum of real-world variation.

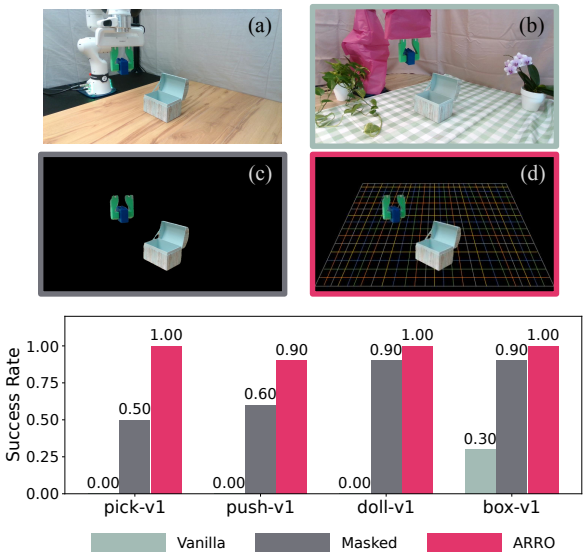

Fig. 1. Visualization of input formats (top) and performance comparison across four manipulation tasks (bottom). (a) shows the training scene, while (b) depicts an altered scene with visual domain shifts used during inference. (c) and (d) illustrate the corresponding inputs for the masked diffusion policy and ARRO, respectively.

To address these challenges, we propose a shift in perspective motivated by recent developments in foundation models for computer vision. Instead of training policies to handle every possible visual scenario, we aim to transform the input space to a canonical representation that cancels out scene variations not relevant for the task. Concretely, we ask:

Can we equip robots with a task-oriented augmented reality view that selectively filters irrelevant information and emphasizes only what is essential for task execution?

In this paper, we introduce **ARRO** (**A**ugmented **R**eality for **RO**bots), a method that functions like *AR glasses* for robots—minimizing visual distractions, highlighting task-relevant elements, and enhancing policy robustness without requiring retraining. ARRO is a calibration-free visual preprocessing pipeline for generating task-specific augmented visual observations. By leveraging open-vocabulary segmentation and object detection, it retains only the robot gripper and the target objects, and overlays them onto a structured

*Equal contribution [1]All authors are with the Department of Computer Science and Artificial Intelligence, University of Technology Nuremberg, Nuremberg, Germany.

virtual background. This process results in a consistent, simplified input space that supports visuomotor robustness across varied environments and embodiments.

In summary, we make the following contributions: (1) We propose ARRO, a calibration-free augmented reality pipeline that improves the robustness of visuomotor policies by selectively retaining task-relevant visual information—specifically, the fingers of the robot's gripper and manipulated objects—while masking out distractors. **(2)** We introduce ARRO's system design, which is based on vision-language models and computer vision foundation models for segmentation and object detection. As a result, it operates in an open-vocabulary and zero-shot manner. Neither camera calibration, nor modeling of the task or environment, nor training are required. In principle, ARRO can be combined with any visuomotor policy. **(3)** We create a virtual background with clear visual references that is included in the virtual scene. Our experimental results show that this improves performance compared to a uniform background without visual markers. **(4)** We evaluate ARRO in multiple tasks and settings on Diffusion Policy [4], Octo [16], OpenVLA [17] and π_0 [18]. Our results show improved robustness to background variations, distractors, and highlight how ARRO enables cross-embodiment transfer.

II. Related Works

Domain Adaptation in Imitation Learning Domain adaptation for visuomotor robot policies has been explored in different contexts. In reinforcement learning, it is especially relevant when transferring a policy trained in simulation to the real world and can be addressed, for example, through domain randomization [21], [22]. In imitation learning, large-scale datasets have emerged as a key strategy for enabling generalization across varied tasks, embodiments, and environments. The Open X-Embodiment dataset [13] consolidates demonstrations from a wide range of robotic platforms and has enabled the training of generalist robot policies [13], [16], [17]. Given the cost of collecting large-scale robot data, several works leverage human videos [23], [24], [25]. Another line of research addresses embodiment differences by using wrist-mounted cameras [26], [27], which maintain consistent robot-centric viewpoints but cannot benefit from the large amount of available third-person data and are limited in tasks requiring broader scene understanding. In contrast, ARRO operates directly on third-person data without retraining or new data collection.

Environment Representations for Visuomotor Manipulation Policies Representations of the environment that are robust against domain shift are a long-standing research topic in robot learning. Object-centric representations [5], [28] aim to extract task-relevant objects from the scene while excluding everything else. In contrast, ARRO performs the same operation directly in image space and does not require any training. Transporter networks [29] extend the idea of object-centric representations by not only identifying relevant objects but also outputting corresponding actions that can be conditioned on language [30]. Another line of work leverages keypoints and constraints between them to represent objects and goal poses of manipulation tasks [31], [32]. Recently, foundation models were used to generate both keypoints and constraints automatically [33]. Vision-language models have also been shown to operate entirely without keypoints by reasoning over visual annotations in the camera image [34], [35]. A disadvantage of keypoint-based methods is that they make implicit assumptions about the types of tasks to be executed. More general approaches learn representations from large datasets, such as human videos [36], [37] and also support 3D data [38].

Visual Editing Most closely related to our approach are methods that apply direct visual editing to RGB camera images. Several works augment training data by synthesizing new backgrounds, tasks, or distractor objects [9], [39], [40], [41], [42]. *RoboSaGa* leverages saliency extraction to overlay out-of-distribution images adaptively while preserving task-relevant image regions [43]. Other approaches focus on adapting visual inputs for cross-embodiment transfer and scene variations. In a method termed *VR-Goggles for Robots*, the authors apply style transfer methods to transform real-world camera images to synthetic images that capture the style of the simulation environment that was used for training [44]. *Mirage* [45] enables zero-shot cross-embodiment transfer via cross-painting—masking the target robot, inpainting missing regions, and overlaying rendered source robot images. While effective for robot arm transfer, it requires precise camera calibration, URDF files, and does not handle background shifts. *Shadow* [46] overlays robot segmentation masks for embodiment transfer without new data collection, but still requires policy retraining, static backgrounds, and precise calibration. *RoVi-Aug* [47] augments robot datasets using fine-tuned diffusion models to generate new robot embodiments and camera viewpoints. However, it requires paired data, additional training, and does not address background variation. In contrast, our method filters out irrelevant visual information via open-vocabulary segmentation, achieving robustness to background and embodiment changes without needing calibration, re-training, or extra data.

III. Approach

We adopt the standard visuomotor control setting, where at each time step t, a robot observes a camera frame I_t and executes an action a_t based on a policy $\pi(a_t, \ldots, a_{t+T_a} \mid I_{t-T_o}, \ldots, I_t)$ with observation horizon T_o and action horizon T_a. Throughout this paper, I_t is an RGB camera image and a_t an absolute or relative task space action. π is trained on a source-domain dataset of expert demonstrations $\mathcal{D}_{\text{train}} = \{(I_t, a_t)\}$. When deployed in a target domain $\mathcal{D}_{\text{test}}$ with novel backgrounds, distractor objects, or changes in robot embodiment, such policies often fail due to sensitivity to task-irrelevant visual variations. Our objective is to compute a calibration-free visual transform $\Phi(I_t) = \tilde{I}_t$ that takes as input the unaltered image I_t and outputs the corresponding augmented image $\tilde{I}_t$, which only contains task-relevant elements. We design Φ based on state-

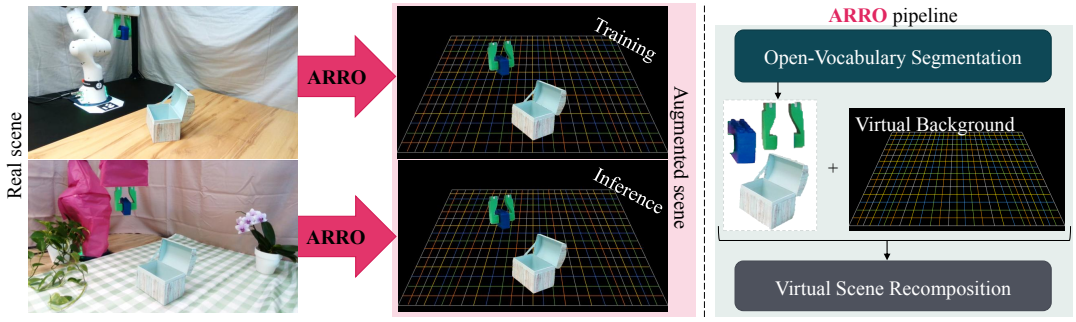

Fig. 2. ARRO in a nutshell: Our pipeline segments the robot gripper and task-related using open-vocabulary vision models and overlays them onto a virtual background. We consistently use this process across training and inference to enhance robustness against visual domain shifts.

of-the-art vision foundation models to enable zero-shot open-vocabulary object detection without training and apply a calibration-free initialization scheme that leverages a vision-language model. ARRO can thus be deployed without any setup-specific adjustments. Fig. 2 provides an overview of ARRO's processing pipeline.

A. Open-Vocabulary Segmentation

To isolate the robot's gripper fingers and task-related objects, we employ a two-phase segmentation process, consisting of an initial segmentation on the first incoming frame, followed by temporally consistent segmentation propagation over time. At the start of an episode, we apply an open-vocabulary object detection model (e.g., Grounding DINO [48]), prompted with object class labels $p_1^o, \ldots, p_n^o$, to the first frame I_0, yielding a set of bounding boxes $\mathcal{B}$ with:

$$B_i = \texttt{Detect}(I_0, p_i^o) \tag{1}$$

These bounding boxes can then be used to extract the object with a promptable segmentation model like SAM 2 [49]. Segmenting the gripper fingers requires a different approach. Standard object detection models are not well-suited for detecting robot grippers, as such objects are typically underrepresented in the training data. Therefore, to segment the fingers, we first apply standalone segmentation to I_0, producing a set $\mathcal{K}$ of region proposals without any prompts:

$$\{K_0, \ldots, K_l\} = \texttt{Segment}(I_0) \tag{2}$$

The original image I_0 is then annotated by placing numbered labels at the center of each segmented region. This annotated image, denoted as I_0^*, is passed to a vision-language model (e.g., GPT-4o [50]) along with a task prompt p^t to identify the regions corresponding to the gripper fingers. This approach also works for objects that have a simple shape and, therefore, can be segmented without a bounding box:

$$\{K_0^*, \ldots, K_m^*\} = \texttt{VLM}(I_0^*, p^t) \tag{3}$$

The output $\mathcal{K}^*$ of the model represents the detected keypoints $K_i^* = (x_{K_i^*}, y_{K_i^*})$ of the specified objects as well as the left and right gripper fingers. Next, using the object bounding boxes $\mathcal{B}$ and the keypoints $\mathcal{K}^*$ as prompts, we apply a memory-based segmentation model to extract the segmented regions from the input image:

$$S_0^{\text{obj}},\ S_0^{\text{gripper}} = \texttt{Segment}(I_0 \mid \mathcal{B}, \mathcal{K}^*) \tag{4}$$

In our experiments, we use SAM 2 [49]. Once the initial segmentation has been obtained, we track the object and gripper regions, S_t^{obj} and S_t^{gripper}, across all subsequent frames I_t for $t > 0$, by conditioning on the memory accumulated during earlier frames [49]. This enables temporally consistent segmentation over time without requiring re-identification by the vision-language model or additional supervision. The full initialization procedure is summarized in Algorithm 1.

B. Virtual Scene Recomposition

Once the relevant segmentations are obtained at each timestep t, we extract the task-relevant regions as described in Algorithm 2. After retrieving the segmentation masks for frame I_t at time step t with the initialized model, we compute their union $S_t = S_t^{\text{obj}} \cup S_t^{\text{gripper}}$, where S_t^{obj} and S_t^{gripper} denote the binary masks for the object and the robot's gripper fingers, respectively. We then overlay S_t on a simple black background or a hand-crafted colored grid that is reused across all sequences. While not photorealistic, this background provides visual cues and consistency across frames. The final background-augmented image $\tilde{I}_t$ is computed as

$$\tilde{I}_t = S_t \odot I_t + (1 - S_t) \odot I_B, \tag{5}$$

where I_B is the selected background image, and $\odot$ denotes element-wise multiplication. This augmentation process is applied to all frames in $\mathcal{D}_{\text{train}}$ and runs in real time during inference.

IV. Experimental Results

We evaluate ARRO both in real-world experiments and in simulation with Diffusion Policy [4], Octo [16], OpenVLA [17] and π_0 [18]. We collected datasets for training and fine-tuning with manual teleoperation and automated control scripts for which we used the Robot Control Stack [51] on a Franka Research 3 (FR3) robot with custom 3D-printed gripper fingers. Each step of an episode contains the complete robot state, including the end effector pose and a third-person view camera image of the scene, which were used for training. For π_0 we created a special dataset which

Algorithm 1 ARRO Initialization

Input: An RGB frame I; object prompts $p_1^o, \ldots, p_n^o$; task prompt p^t; open-vocabulary object detector `Detect` (e.g., `GroundingDINO`); uninitialized segmentation model `Segment` (e.g., `SAM2`); and a vision-language model `VLM` (e.g., `GPT-4o`).

Output: The initialized segmentation model $\texttt{Segment}_0$

// Get object bounding boxes for complex objects:
for p_i^o in $\{p_1^o, \ldots, p_n^o\}$ **do**
  $B_i \leftarrow \texttt{Detect}(I, p_i^o)$
end for

// Run unprompted segmentation on I to get region masks to
// segment simple objects and gripper fingers:
$\{K_0, \ldots, K_l\} \leftarrow \texttt{Segment}(I)$

// Annotate keypoints in I with numeric labels to retrieve I^* and
// identify task-relevant keypoints in I^* from $\{K_0, \ldots, K_l\}$:
$\{K_0^*, \ldots, K_m^*\} \leftarrow \texttt{VLM}(I^*, p^t)$

// Initialize and return the segmentation model:
return $\texttt{Initialize}(\texttt{Segment}, I, B_0, \ldots, B_n, K_0^*, \ldots, K_m^*)$

Algorithm 2 ARRO Masking

Input: RGB frame I_t; RGB background image I_B; initialized segmentation model $\texttt{Segment}_t$ (e.g., `SAM2`).

Output: A masked RGB frame $\tilde{I}_t$ and the updated segmentation model $\texttt{Segment}_{t+1}$

// Track object and gripper masks:
$S_t^{\text{obj}},\ S_t^{\text{gripper}},\ \texttt{Segment}_{t+1}(I) \leftarrow \texttt{Segment}_t(I)$

// Combine masks:
$S_t \leftarrow S_t^{\text{obj}} \cup S_t^{\text{gripper}}$

// Overlay S on virtual background I_B retrieve $\tilde{I}$:
$\tilde{I}_t \leftarrow S_t \odot I_t + (1 - S_t) \odot I_B$

return $\tilde{I}_t,\ \texttt{Segment}_{t+1}$

also contains wrist-mounted camera images. All models were trained or fine-tuned using the code released along with their original publications. We selected representative tabletop manipulation tasks to evaluate the performance of our proposed image augmentation pipeline under complex visual conditions. They involve the manipulation of soft deformable objects (e.g., an octopus plush toy), visually rich and colorful textures (e.g., a doll), and dynamic geometric object variations (e.g., closing the lid of a box).

A. Real-World Experiments for Diffusion Policy

We conduct four real-world tabletop manipulation tasks that are illustrated in Fig. 3. They include: (a) picking up a single blue cuboid (pick-v1), (b) pushing a cube to a red cross location (push-v1), (c) placing a plush octopus next to a doll (doll-v1), and (d) dropping a cube into a box and closing the lid (box-v1). The tasks were selected to evaluate a range of manipulation behaviors under realistic visual conditions. For each of them, we collect 90 human demonstrations via teleoperation for training and fine-tuning the policies.

1) Performance Across Domain Shifts: In this section, we evaluate whether ARRO mitigates the degradation in

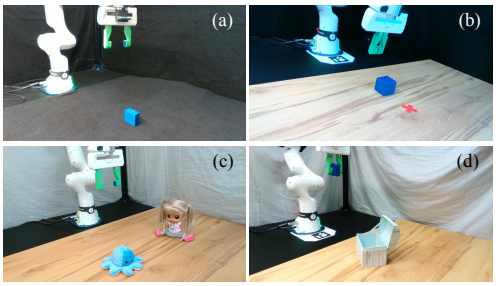

Fig. 3. Real-world experiment setups for the (a) pick-v1, (b) push-v1, (c) doll-v1 and (d) box-v1 tasks.

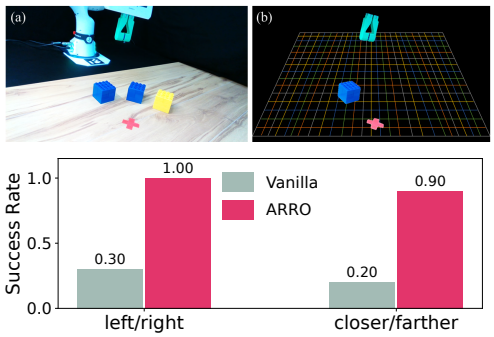

Fig. 4. Handling distractor objects. *Top:* example input image with distractors (a) and the ARRO-segmented version (b) where the task is to "Push the *blue cube that is farther from the yellow cube to* the red cross". *Bottom:* Success rates for the spatial reasoning tasks.

policy performance caused by visual domain shifts. To assess robustness, we introduce significant visual changes at evaluation time, including alterations to the background, modifications to table texture and robot appearance, and the addition of irrelevant objects to the scene. An example of these perturbations for the box-v1 task is shown in Fig. 2.

We compare three variants: the *Vanilla Diffusion Policy*, trained and evaluated on unmodified RGB images; the *Masked Diffusion Policy*, which segments the task-relevant objects and places them on a plain black background by masking out all other regions; and *ARRO*, which uses the same segmentation but overlays the task-relevant components onto a structured virtual grid background. Fig. 1 illustrates these representations for the box-v1 task. For each task and each model, we train the policy for 1000 epochs and evaluate on ten trials.

As shown in Fig. 1, ARRO consistently outperforms both baselines. The Vanilla Diffusion Policy suffers from substantial performance degradation across all scenarios due to its reliance on raw visual features, which no longer remain reliable under domain shifts. The Masked Diffusion Policy partially retains performance by removing distractors. However, it still performs noticeably worse than ARRO. This suggests that while masking suppresses irrelevant information, it may also discard useful visual cues that contribute to task execution. In contrast, ARRO not only filters out distractors but also reintroduces a structured and consistent background, improving robustness and the ability to generalize across varied visual conditions.

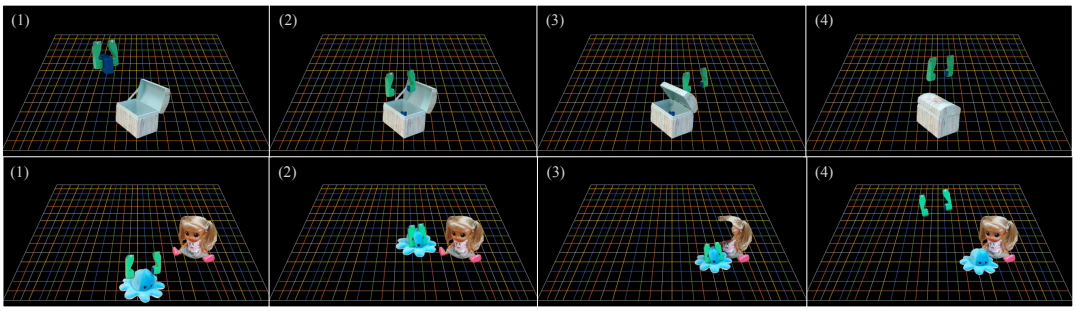

Fig. 5. Execution sequences using ARRO on the box-v1 and doll-v1 tasks. By overlaying the segmented task-relevant regions on a virtual background, ARRO neutralizes the effect of visual domain shifts. ARRO's segmentation remains robust to transient occlusions caused by the robot arm or other objects. The doll is temporarily occluded by the octopus plush toy and the gripper in frame (3), but its segmentation accurately reappears once the occlusion clears, without manual correction.

To complement the quantitative results, Fig. 5 presents representative execution sequences using ARRO for the box-v1 and doll-v1 tasks. Notably, the segmentation module maintains reliable performance even under challenging conditions, such as partial occlusion (e.g., when the doll is partially blocked by the plush toy) or shape deformation (e.g., when the plush toy is grasped or the box begins to close). Despite the presence of visually complex and colorful objects, such as the multi-textured doll, the segmentation remains accurate throughout execution.

2) Handling Distractor Objects: To assess whether ARRO can handle distractor objects and enable spatial reasoning, we repeat the push experiment with additional distractor items in the scene. Specifically, we investigate whether the vision-language model can move beyond basic object recognition (e.g., selecting "the blue cube") to spatial grounding (e.g., selecting "the blue cube farther from the yellow cube").

We design two spatial reasoning tasks to evaluate whether the system can correctly identify and act upon objects based on spatial relationships. In the first task, the robot must "push the *blue cube on the left/right* to the red cross." In the second, it must "push the *blue cube that is closer to/farther from the yellow cube* to the red cross." Once the vision-language model identifies the correct task-relevant object, ARRO masks out all other image regions, including distractor objects, ensuring that the policy operates solely on the relevant visual input. The resulting task seen by the policy is therefore identical to the push-v1 task, which means that no additional data collection or training is required.

As shown in Fig. 4, ARRO demonstrates strong performance in both tasks and reliably selects the appropriate object based on the spatial relation described in the instruction. In contrast, the vanilla diffusion policy exhibits a substantial drop in performance. It frequently fails to identify the correct object, sometimes selecting a distractor, switching targets mid-execution, or moving ambiguously between multiple candidates. Even when the correct object is selected, the resulting motion is often imprecise. These results highlight ARRO's robustness to distractors and its capacity to support spatially grounded reasoning in visually complex environments.

3) Handling Occlusions in ARRO: Partial occlusions of objects may arise during certain tasks, either due to the robot's own movements or the presence of other objects within the scene, as illustrated in Fig. 5. In frame (3), the doll is partially occluded—either by the robot arm or by the octopus plush toy temporarily blocking the view. Despite these transient occlusions, ARRO's segmentation remains stable and accurate. As shown in the figure, once the occlusion subsides, the segmentation of the doll reliably reappears without requiring manual intervention or additional adjustment. This demonstrates that our segmentation pipeline is robust to temporary occlusions and consistently preserves the identification of task-relevant objects over time.

B. Real-World Experiments for Generalist Policies

Because ARRO operates directly on raw camera images, it is compatible with any type of visuomotor policy. We therefore explore whether it can also enhance the performance of the language-conditioned generalist policies Octo [16], OpenVLA [17], and π_0 [18]. While the former two are fine-tuned on pick-v1, the latter is fine-tuned on pick-wrist-v1, which contains episodes for the same task with a green cuboid, but also includes images from a wrist camera mounted close to the robot's gripper. We found that the performance of π_0 decreases substantially when it is used with a fixed side camera only.

Using a wrist camera requires additional steps to apply our method, as the background changes when the robot moves, and task-relevant objects may not always be in the camera's field of view. To address this, we always use a black background and apply the ARRO pattern only to the side camera. Moreover, we initialize the segmentation with a pre-recorded image of the scene that shows all relevant objects. We found this method to be robust even if the scene is perturbed. As the gripper fingers always appear at the same initial position in the wrist camera view, we select them directly instead of using the VLM. The wrist camera view can be seen in Fig. 7.

1) Performance Across Domain Shifts: We evaluated all three models on the pick task under different conditions.

The results are presented in Fig. 6. As shown in the figure, both the masked and ARRO variants consistently outperform their vanilla counterparts in settings where the visual domain is altered at evaluation time. For Octo and OpenVLA, the degradation in performance of the vanilla variant under domain shifts in some cases results in abrupt and unpredictable behavior. We found π_0 to be more robust against visual changes, which is why we added more clutter to the scene, as depicted in Fig. 7. As shown in Fig. 6, this causes the vanilla success rate to drop to 0%, which can be recovered with both the black masked background and ARRO.

2) Language Guidance: While π_0 is more robust against scene changes compared to Octo and OpenVLA, we found that adding an additional red cuboid with the same shape can degrade its performance, as the model ignores the cuboid color in its task instruction. As a result, the vanilla policy often grasps the wrong cuboid leading to a success rate of 30%. The masked and ARRO variants, which exclude the task-irrelevant red cube from the scene, increase the success rates to 70% and 90%, respectively.

C. ARRO in Simulation

In addition to the real-world experiments, we also evaluate ARRO in simulation on the tasks pick-v2 and sim-pick-v1. Unlike in pick-v1, the cuboid is red, and we collected the demonstrations automatically via a script in both the real world and the simulation. For π_0 we choose pick-wrist-v1, which includes the wrist camera. We investigate whether ARRO can help mitigate the real-to-sim gap [20] and facilitate cross-embodiment policy transfer. Fig. 8 illustrates our experimental setup: the physical setup in our lab using the FR3 robot, its replication in a MuJoCo [52] simulation environment, and a cross-embodiment variant where the FR3 is replaced by a UR5e robot in the same environment.

1) Real-to-Sim: Since most datasets for robot imitation learning are collected on real-world setups, we examine a real-to-sim rather than the more typical sim-to-real paradigm to assess ARRO's performance in domain transfer. We trained Diffusion Policy, Octo and OpenVLA on the pick-v2 dataset, and π_0 on the pick-wrist-v1 dataset, and evaluated the task success rates for two settings: The unmodified real-world scene and the replicated scene in the simulation. We evaluated the trained policies in-distribution on the same unchanged real-world setup on 10 episodes (real-to-real success rate) and out-of-distribution on the replicated MuJoCo setup on 100 episodes (real-to-sim success rate). All models achieve decent real-to-real success probabilities in the vanilla case: 90%, 50%, 40%, and 100% for Diffusion Policy, Octo, OpenVLA, and π_0, respectively. But as shown in Table I, the performance drops to 0% of their real-world performance when evaluated in the real-to-sim setting for all models except for π_0. With ARRO, OpenVLA retains 55% of its real-world performance. Octo does not benefit as much and only achieves 5% of its original success rate. Diffusion Policy does not exhibit any immediate transfer in terms of success rate, which can be attributed, at least in part, to the absence of a pre-training stage and differences in the vision backbone architecture. Moreover, even after applying ARRO, the visual appearance remains different from that in the real-world experiment, where we found that subtle visual changes can cause the policy to fail, too. π_0 already has a high baseline performance and can retain 94% of its success rate when masking is applied. The low relative performance of ARRO on π_0 can be explained by a high real-to-real success probability of 90%, while the real-to-sim success rate is comparable to that of the masked variant. In addition to the success rates in the real-to-sim experiments, we computed reward values using a simple distance metric inspired by ManiSkill [53]. The results are shown in Fig. 9. For Diffusion Policy, Octo, and OpenVLA, both ARRO and the simple black background yield higher rewards than the vanilla baselines.

2) Cross-Embodiment: To test the cross-embodiment capabilities of ARRO, we train our policies on a simulation dataset, referred to as sim-pick-v1, using the FR3 and test it on a UR5e embodiment. We evaluate both in-distribution on the training environment for 100 episodes and on the new UR5e embodiment for 100 episodes. The relative per-

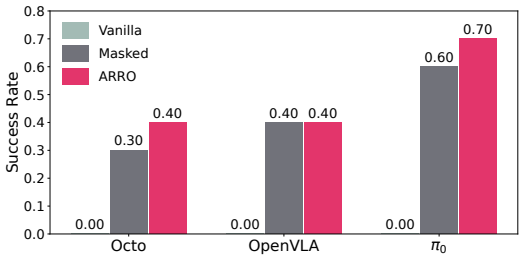

Fig. 6. Performance comparison across three generalist models, Octo and OpenVLA and π_0, evaluated under altered visual conditions.

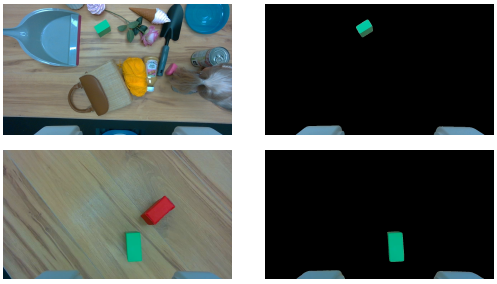

Fig. 7. Wrist camera view of ARRO for the π_0 experiment "pick up the green cube". *Top*: Example input image of a cluttered scene and the masked version. *Bottom*: Scene with a red cube that is masked by ARRO.

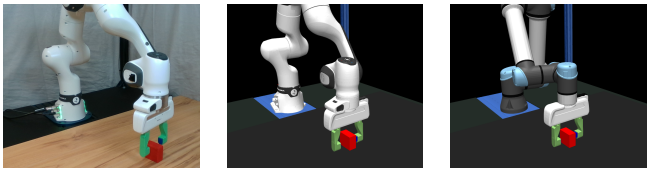

Fig. 8. Experimental setup for the pick task in two environments and one cross-embodiment scenario. *Left to right*: The real-world FR3 setup, the model of the setup in MuJoCo used for collecting sim-pick-v1, and the UR5e setup for cross-embodiment experiments in MuJoCo.

TABLE I: RELATIVE POLICY PERFORMANCE OF REAL-TO-SIM AND CROSS-EMBODIMENT EXPERIMENTS AFTER DOMAIN TRANSFER.

			Real-to-Sim				Cross-Embodiment			
Policy	**Camera**	**Control**	**Dataset**	**Vanilla**	**Masked**	**ARRO**	**Dataset**	**Vanilla**	**Masked**	**ARRO**
Diffusion Policy	Side	Cartesian	pick-v2	0%	0%	0%	sim-pick-v1	0%	80%	**99%**
Octo	Side	Cartesian	pick-v2	0%	3%	**5%**	sim-pick-v1	71%	**87%**	**87%**
OpenVLA	Side	Cartesian	pick-v2	0%	53%	**55%**	sim-pick-v1	29%	72%	**80%**
π_0	Side + Wrist	Joints	pick-wrist-v1	79%	**94%**	77%	—	—	—	—

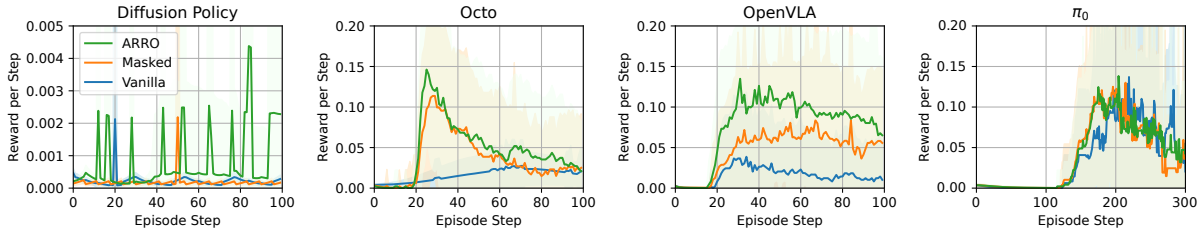

Fig. 9. Normalized mean per step reward over 100 episodes on the sim-pick-v1 task in simulation on the FR3 (same embodiment). All models were trained only on real-world datasets. Shaded areas indicate standard deviations.

formance is shown in Table I under cross-embodiment. Note that π_0 is not included, as it operates in joint space and therefore cannot be directly applied to the UR5e robot due to its different kinematics.

The diffusion policy's performance drops to 0% when evaluated on a novel embodiment, which is expected given that it was trained exclusively on the FR3 robot. In contrast, both the masking with a black background and ARRO exhibit only a minor reduction in success rates, retaining most of their performance. This is likely because the embodiment change has minimal impact on the diffusion policy: the visual input remains largely unaffected due to the masking of the robot body, and the control commands are issued in absolute Cartesian space, which remains invariant to the embodiment.

Octo and OpenVLA also exhibit reduced drops in success rates when combined with ARRO and masking. Unlike the Diffusion Policy, their performance does not drop to 0% in the vanilla setting for the new embodiment. This can be attributed to their larger and more diverse pre-training datasets, which are explicitly designed to support cross-embodiment generalization. As a result, both models encountered the UR5e robot during training, facilitating better transfer even in simulation settings.

V. LIMITATIONS

While our method shows clear advantages across all evaluated models and tasks, its performance depends heavily on the underlying segmentation model, especially for wrist cameras, where objects can appear or disappear abruptly. In our experiments, task-relevant objects were occasionally not tracked reliably, leading to degraded performance, which could be quantified using standard image segmentation metrics. This issue could be mitigated by periodically re-initializing the segmentation model or by running a lightweight object detector in parallel to trigger tracker resets for the segmentation model as objects enter or leave the scene. Notably, our approach can still function under minor segmentation errors, depending on the visuomotor policy's tolerance to such perturbations. Finally, our method cannot compensate for reflections and changes in lighting. These types of domain shifts could be mitigated with image inpainting methods or by integrating depth images. However, the real-to-sim results indicate that more recent models, such as π_0, are less sensitive to such changes.

VI. CONCLUSION

We introduce ARRO, a calibration-free visual augmentation pipeline that enhances the robustness of visuomotor policies by segmenting task-relevant regions and compositing them onto a structured virtual background. Our method improves generalization without requiring retraining or camera calibration, and experiments in both real and simulated settings show consistent gains for task-specific as well as generalist policies. A promising direction for future work is to investigate how the background pattern affects performance. This also includes replacing the static background overlay with dynamic, view-dependent frames for the wrist-mounted camera.

ACKNOWLEDGMENT

The authors acknowledge the HPC resources provided by the Erlangen National HPC Center (NHR@FAU) under the BayernKI project number v106be.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.
[2] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, 2020.
[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2021.
[4] C. Chi, Z. Xu, S. Feng, *et al.*, "Diffusion policy: Visuomotor policy learning via action diffusion," *Int. Journal of Robotics Research (IJRR)*, 2023.
[5] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2023.

[6] M. Dalal, A. Mandlekar, C. R. Garrett, A. Handa, R. Salakhutdinov, and D. Fox, "Imitating task and motion planning with visuomotor transformers," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2023.

[7] H. R. Walke, K. Black, T. Z. Zhao, *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2023.

[8] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile ALOHA: Learning bimanual mobile manipulation using low-cost whole-body teleoperation," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2025.

[9] H. Bharadhwaj, J. Vakil, M. Sharma, A. Gupta, S. Tulsiani, and V. Kumar, "RoboAgent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2024.

[10] H.-S. Fang, H. Fang, Z. Tang, *et al.*, "Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2024.

[11] A. Khazatsky, K. Pertsch, S. Nair, *et al.*, "DROID: A large-scale in-the-wild robot manipulation dataset," in *Proc. of Robotics: Science and Systems (RSS)*, 2024.

[12] K. Wu, C. Hou, J. Liu, *et al.*, "RoboMIND: Benchmark on multi-embodiment intelligence normative data for robot manipulation," in *Proc. of Robotics: Science and Systems (RSS)*, 2025.

[13] A. O'Neill, A. Rehman, A. Maddukuri, *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2024.

[14] A. Brohan, N. Brown, J. Carbajal, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," in *Proc. of Robotics: Science and Systems (RSS)*, 2023.

[15] B. Zitkovich, T. Yu, S. Xu, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2023.

[16] D. Ghosh, H. R. Walke, K. Pertsch, *et al.*, "Octo: An open-source generalist robot policy," in *Proc. of Robotics: Science and Systems (RSS)*, 2024.

[17] M. J. Kim, K. Pertsch, S. Karamcheti, *et al.*, "Openvla: An open-source vision-language-action model," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2025.

[18] K. Black, N. Brown, D. Driess, *et al.*, "π_0: A vision-language-action flow model for general robot control," https://arxiv.org/abs/2410.24164, 2024.

[19] A. Xie, L. Lee, T. Xiao, and C. Finn, "Decomposing the generalization gap in imitation learning for visual robotic manipulation," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2024.

[20] X. Li, K. Hsu, J. Gu, *et al.*, "Evaluating real-world robot manipulation policies in simulation," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2025.

[21] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[22] Y. Chebotar, A. Handa, V. Makoviychuk, *et al.*, "Closing the sim-to-real loop: Adapting simulation randomization with real world experience," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2019.

[23] A. S. Chen, S. Nair, and C. Finn, "Learning generalizable robotic reward functions from "in-the-wild" human videos," in *Proc. of Robotics: Science and Systems (RSS)*, 2021.

[24] H. Xiong, Q. Li, Y.-C. Chen, H. Bharadhwaj, S. Sinha, and A. Garg, "Learning by watching: Physical imitation of manipulation skills from human videos," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[25] M. Lepert, J. Fang, and J. Bohg, "Phantom: Training robots without robots using only human videos," https://arxiv.org/abs/2503.00779, 2025.

[26] J. H. Yang, D. Sadigh, and C. Finn, "Polybot: Training one policy across robots while embracing variability," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2023.

[27] C. Chi, Z. Xu, C. Pan, *et al.*, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," in *Proc. of Robotics: Science and Systems (RSS)*, 2024.

[28] C. Devin, P. Abbeel, T. Darrell, and S. Levine, "Deep object-centric representations for generalizable robot learning," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, 2018.

[29] A. Zeng, P. Florence, J. Tompson, *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2021.

[30] M. Shridhar, L. Manuelli, and D. Fox, "CLIPort: What and where pathways for robotic manipulation," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2022.

[31] M. Sieb, Z. Xian, A. Huang, O. Kroemer, and K. Fragkiadaki, "Graph-structured visual imitation," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2020.

[32] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "KPAM: Keypoint affordances for category-level robotic manipulation," in *Proc. of the Int. Symp. of Robotics Research (ISRR)*, 2019.

[33] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2024.

[34] S. Nasiriany, F. Xia, W. Yu, *et al.*, "PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs," in *Proc. of the Int. Conf. on Machine Learning (ICML)*, 2024.

[35] K. Fang, F. Liu, P. Abbeel, and S. Levine, "MOKA: Open-world robotic manipulation through mark-based visual prompting," *Proc. of Robotics: Science and Systems (RSS)*, 2024.

[36] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3M: A universal visual representation for robot manipulation," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2022.

[37] S. Karamcheti, S. Nair, A. S. Chen, *et al.*, "Language-driven representation learning for robotics," in *Proc. of Robotics: Science and Systems (RSS)*, 2023.

[38] S. Chen, R. Garcia, I. Laptev, and C. Schmid, "SUGAR: Pre-training 3d visual representations for robotics," in *Proc. of the Int. Conf. on Computer Vision (ICCV)*, 2024.

[39] Z. Mandi, H. Bharadhwaj, V. Moens, S. Song, A. Rajeswaran, and V. Kumar, "CACTI: A framework for scalable multi-task multi-scene visual imitation learning," in *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022.

[40] T. Yu, T. Xiao, A. Stone, *et al.*, "Scaling robot learning with semantically imagined experience," https://arxiv.org/abs/2302.11550, 2023.

[41] Z. Chen, S. Kiami, A. Gupta, and V. Kumar, "Genaug: Retargeting behaviors to unseen situations via generative augmentation," https://arxiv.org/abs/2302.06671, 2023.

[42] M. Shridhar, Y. L. Lo, and S. James, "Generative image as action models," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2025.

[43] Z. Zhuang, R. WANG, N. Ingelhag, V. Kyrki, and D. Kragic, "Enhancing visual domain robustness in behaviour cloning via saliency-guided augmentation," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2025.

[44] J. Zhang, L. Tai, P. Yun, *et al.*, "Vr-goggles for robots: Real-to-sim domain adaptation for visual control," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, 2019.

[45] L. Y. Chen, K. Hari, K. Dharmarajan, C. Xu, Q. Vuong, and K. Goldberg, "Mirage: Cross-embodiment zero-shot policy transfer with cross-painting," in *Proc. of Robotics: Science and Systems (RSS)*, 2024.

[46] M. Lepert, R. Doshi, and J. Bohg, "Shadow: Leveraging segmentation masks for cross-embodiment policy transfer," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2025.

[47] L. Y. Chen, C. Xu, K. Dharmarajan, *et al.*, "Rovi-aug: Robot and viewpoint augmentation for cross-embodiment robot learning," in *Proc. of the Conf. on Robot Learning (CoRL)*, 2025.

[48] S. Liu, Z. Zeng, T. Ren, *et al.*, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," in *Proc. of Europ. Conf. on Computer Vision (ECCV)*, 2025.

[49] N. Ravi, V. Gabeur, Y.-T. Hu, *et al.*, "SAM 2: Segment anything in images and videos," in *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2025.

[50] OpenAI, J. Achiam, S. Adler, *et al.*, "Gpt-4 technical report," https://arxiv.org/abs/2303.08774, 2024.

[51] T. Jülg, P. Krack, S. Bien, *et al.*, "Robot Control Stack: A lean ecosystem for robot learning at scale," https://arxiv.org/abs/2509.14932, 2025.

[52] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[53] J. Gu, F. Xiang, X. Li, *et al.*, "ManiSkill2: A unified benchmark for generalizable manipulation skills," in *Proc. of the Int. Conf. on Learning Representations (ICLR)*, 2023.