Eye, Robot: Learning Hand-Eye Coordination with Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

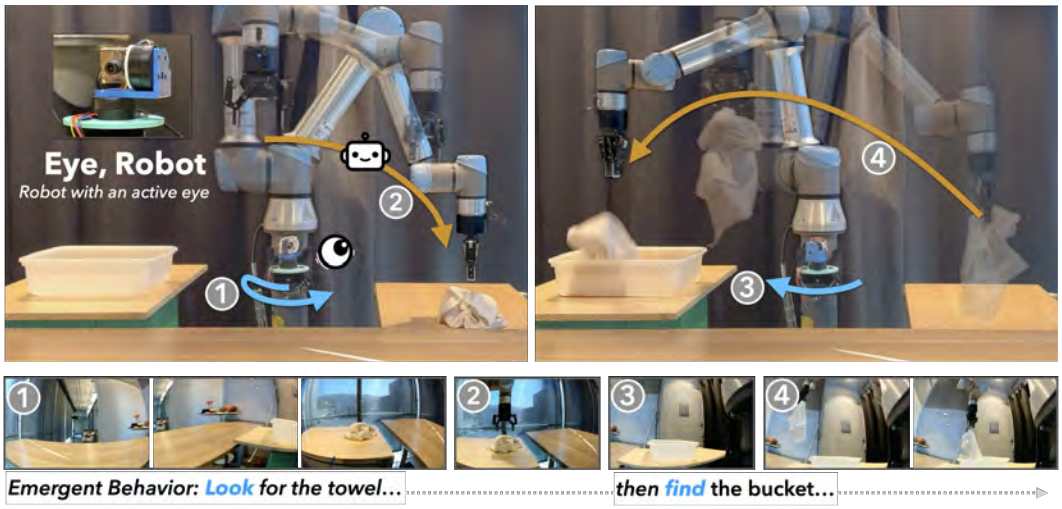

Figure 1: **EyeRobot.** We present a robotic system with an active eye, where the behavior of looking emerges from the need to act. A foveated mechanical eye, inspired by biological vision, is trained via reinforcement learning in a novel real-to-sim BC-RL loop. Shown here is a long-horizon pick-and-place task involving a towel and a bucket—neither visible in the initial view. The robot looks for the towel, grasps it, then searches for the bucket to complete the task. This hand-eye coordination emerges purely from the task reward without any gaze demonstration.

"We perceive in order to act and we act in order to perceive." — *JJ Gibson*

Abstract: Humans do not passively observe the visual world—we actively look in order to act. Motivated by this principle, we introduce EyeRobot, a robotic system with gaze behavior that emerges from the need to complete real-world tasks. We develop a mechanical eyeball that can freely rotate to observe its surroundings and train a gaze policy to control it using reinforcement learning. We accomplish this by introducing a BC-RL loop, which is trained using teleoperated demonstrations and eye gaze actions that can be simulated by sampling from 360° video. The hand (BC) agent is trained from rendered eye observations, and the eye (RL) agent is rewarded when the hand produces correct actions. In this way, hand-eye coordination emerges as the eye looks towards regions which allow the hand to complete the task. We evaluate EyeRobot on five large workspace manipulation tasks and compare performance to two common camera setups: wrist and external cameras. Our experiments suggest EyeRobot exhibits hand-eye coordination which effectively facilitates action such as visual search or target switching, which enable manipulation across large workspaces.

Keywords: Active Vision, Reinforcement Learning, Behavior Cloning, Manipulation

Submitted to the 9th Conference on Robot Learning (CoRL 2025). Do not distribute.

1 Introduction

Have you had water recently? Take a moment to reach for the nearest cup. As you do, your eyes move first—scanning the table, darting from region to region to locate the cup. Once it's in sight, your hand follows. This tight coupling between attention and eye movements is not incidental; it reflects a fundamental constraint of the human visual system. We are not built to perceive everything at once. As a result, we must look around—and in this sense, vision can be understood as a form of search. But what drives the search? The need to *act*—to accomplish something in the world. Whether gathering information about task-relevant properties or guiding the execution of a precise action, we look because we have something to do.

In this work, we present a robotic system where the same principle—looking to enable action—emerges naturally from a desired real world task without the need for gaze demonstrations, implemented on a mechanical eyeball that can freely rotate to observe its surroundings. The core challenge is how to train such a visual agent within the constraints of the physical world? To address this, we introduce a BC-RL loop enabled by a 360 camera-based real-to-sim environment. This does not require any gaze demonstrations; instead, hand-eye coordinate emerges solely from task supervision.

To specify what "action" we desire from the system, we build on recent advances in policy learning from behavior cloning [1]. We augment existing teleoperation systems to collect synchronized 360° video and robot trajectory data, creating an EyeGym environment that enables replay of demonstrations with renderings from simulated eyeball viewpoints. Equipped with this environment, we propose a BC-RL algorithm for training an RL eye policy with a task-based reward, which samples rollouts from the current eye agent to use as observations to supervise a behavior cloning agent. The accuracy of these action predictions are cyclically used as rewards for the eye agent (Fig. 2). As these agents co-train, the eye thus learns to look around to optimize the performance of the behavior cloning agent—search emerges from the need to act.

Inspired by nature's solution, we design a Foveal Robot Transformer (FoRT) architecture which processes visual input in a foveated manner: peripheral vision provides broad, low-resolution coverage of the visual field, while foveal vision offers high-resolution input over a restricted area. Our experiments show this multi-resolution architecture results in enhanced fixation during task execution, which can improve downstream performance. In addition, we find that pretraining the eye policy on visual search task slightly improves manipulation performance and convergence speed, though hand-eye coordination with search behavior still emerges even without it. Figure 1 illustrates an example of behavior which emerges in EyeRobot to accomplish a long-horizon pick-and-place task involving a towel and a bucket—neither of which is visible in the initial camera view. The robot begins by scanning the table to locate the towel. After identifying it, the robot picks it up, then shifts its gaze to find the bucket, and finally places the towel inside, all with a single monocular ego-view.

Our experiments evaluate EyeRobot on 5 large-workspace manipulation tasks, involving objects on a 180° panoramic workspace surrounding the arm. Across these tasks we observe a number of emergent behaviors not explicitly trained for—switching from one target to another depending on task stage, long-range search, and independently coordinated hand-eye movements. We find EyeRobot enables promising performance on a variety of large-workspace pick-and-place and servoing tasks with a single egocentric active camera, a physical setup more practically suited to mobile deployment than external cameras. Nevertheless, we compare to external mounted camera baseline, and find that EyeRobot outperforms them, likely due to the limited resolution of the zoomed-out perspective. We also compare with a wrist camera, which performs well when objects are in-view but struggles in a large workspace setup that requires search.

2 Related Work

Active Vision Active perception systems physically move sensors to not only see, but to *look* [2, 3, 4, 5]. This arises naturally from physical constraints faced by embodied agents, which

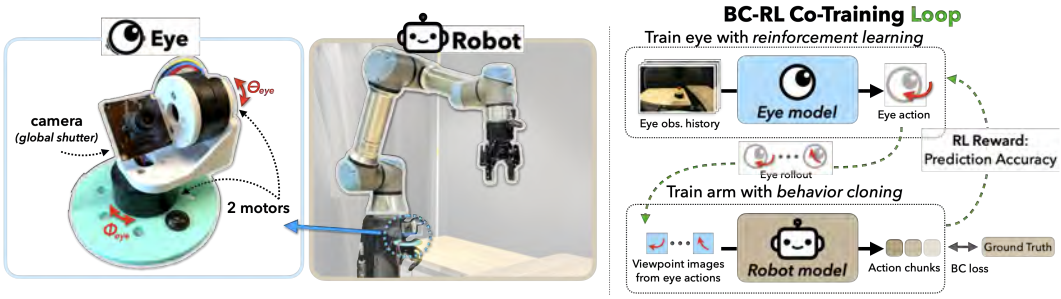

Figure 2: **EyeRobot framework.** Left: Physical hardware setup. We develop a mechanical eyeball with two degrees of freedom, mounted on a high-speed gimbal and equipped with a fisheye lens and global shutter. Right: The eye policy is trained via a BC-RL loop.

have only partial observations of the world at any point in time. The key insight of active vision is that actuation lets agents shape the utility of these observations to better achieve their goals. Existing systems primarily use active vision to maximize information gain for visual tasks: examples include tracking [6, 7], search [8, 9, 10], observation completion [11], and view selection for 3D reconstruction [12, 13, 14, 15] using voxel [16], surfel [17], or point cloud [18] representations. Other works couple active sensing setups with robot manipulators and evaluate systems in terms of downstream task performance: this includes improvements in semantic understanding [19], as well as computational efficiency [20] and occlusion robustness [21] for robot manipulation. Like these systems, EyeRobot also studies active vision for robot manipulation. Instead of relying on heuristics [19] or camera action demonstrations [20, 21], however, EyeRobot proposes to learn optimal eye gaze policies directly with reinforcement learning; we learn where to look in order to act.

Behavior Cloning Behavior cloning (BC) [22, 23, 24, 25] is the dominant paradigm for teaching robots manipulation skills. BC is advantageous because it does not require hand-designed behaviors, costs, and rewards—instead, BC policies are simply trained to imitate human demonstrations. Prior work has shown how this can enable new capabilities in mobile [26, 27], bimanual [1, 28, 29, 30], and language-conditioned [31, 32, 33, 34] robot manipulation. Policies of this form can also be implemented using a diverse range of architectures, including energy-based [35], diffusion [33], and autoregressive [36, 34]. In EyeRobot, we adopt a similar imitation-based system for manipulation in large workspaces. In contrast to prior systems that solely focus on learning from demonstration, however, EyeRobot's BC-RL training shows how behavior cloning objectives for robot actions can (i) be optimized jointly with an eye gaze reinforcement learning objective and (ii) used as a reward itself for reinforcement learning.

Biology-Inspired Machine Perception Many computer vision systems draw inspiration from biology to improve efficiency, adaptability, and performance. For example, simple stereo cameras [37, 38, 39] mimic the binocular vision of animals to support depth perception, while event cameras [40, 41, 42] mimic retinal spikes to enable low-latency perception. Foveated systems emulate the resource-rational nonuniformity of the retina, using either specialized hardware [43, 44, 45] or learned neural mechanisms [46, 47]. Beyond sensing, gaze control has been modeled after oculomotor behaviors like smooth pursuit and saccades [48], and implemented in hybrid pipelines that combine peripheral detection with foveal tracking [49]. EyeRobot builds on the same principles as these prior works, drawing on the benefits of foveation and gaze control. However, its implementation differs significantly: rather than rely on specialized sensors or hardcoded behaviors, we mount a standard RGB camera on a high-speed gimbal, apply foveation using a multi-resolution transformer, and learn a gaze policy via reinforcement learning.

3 Approach

EyeRobot trains gaze policies for manipulation using reinforcement learning. We conduct experiments using a UR5e robot arm with a gimbal-mounted camera mounted rigidly to the base of the robot. To train gaze policies for hand-eye coordination, we propose (i) EyeGym, an environment

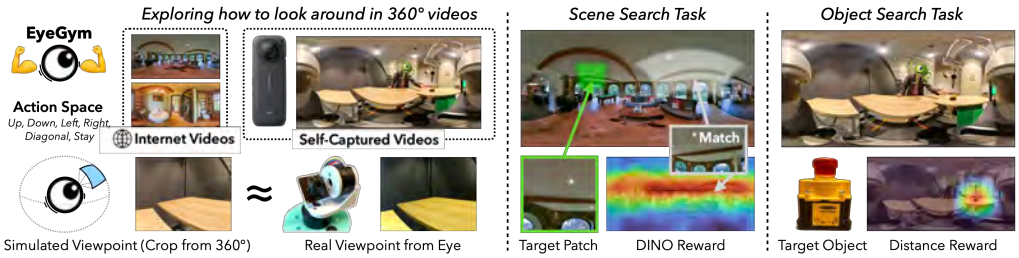

Figure 3: **Learning to Look with EyeGym.** EyeGym enables training policies on 360° internet images or robot data to perform semantic visual search tasks using DINO or Distance-based rewards.

for eye gaze simulation, and (ii) reinforcement learning methodology—visual search rewards and a joint BC-RL loop—for learning gaze policies.

3.1 Scalable Experience Collection with EyeGym

Reinforcement learning benefits from scalable experience collection. To facilitate this for eye gaze policies, we introduce EyeGym: an RL environment that simulates eye gaze by sampling from 360° image and video data. Unlike prior approaches that rely on synthetic environments and physics simulators to render simulated views [50, 51, 52, 53, 54, 55, 56], EyeGym renders directly by sampling from real equirectangular videos and images. This has several key advantages: (i) it reduces the sim2real gap by exposing policies to authentic textures, lighting, and noise; (ii) it enables training on native 360° datasets [57, 58, 59], making EyeGym practical and scalable for learning active visual perception policies that transfer to real-world camera systems (Fig. 3); and (iii) it incurs less overhead than traditional rendering. We will release EyeGym code to support further research.

We use EyeGym for robot manipulation by first replacing our robot's physical eyeball with an off-the-shelf Insta360 X4 360° camera. We can then use this camera to capture robot demonstrations teleoperated using a GELLO system [60], where the 5.7K, 30FPS equirectangular video sequences are recorded and synchronized with robot trajectories. Paired data can then be imported into EyeGym for simulating eye gaze on top of demonstrated robot motion. Advantages of this setup include minimal additional hardware, minimal additional data bandwidth, and compatibility with existing teleoperation systems [1, 60].

3.2 Learning Gaze Policies with Reinforcement Learning

The goal of EyeRobot is to learn gaze policies that improve downstream task performance. This is done without expert gaze demonstrations. We instead use the EyeGym environment to present reinforcement learning policies trained with two sets of rewards: pure visual search and BC-RL.

Visual Search Rewards Search is a critical step of vision for robot manipulation, especially for large workspaces. As an initial gaze policy study, we evaluate two search tasks with explicit visual rewards. *(a) Scene Search* policies attempt to locate image patches that are visually similar to a given target patch. For this, we use DINO feature similarity between current and target views as a reward signal. *(b) Object Search* policies are more fine-grained, and aims to locate specific objects within scenes. For Object Search, we primarily investigate a "truncated distance reward". This reward is zero if the target object is outside the camera's field of view (FOV) and increases linearly to 1 as the agent perfectly centers the target in its view.

The BC-RL Loop The quality of gaze policies in EyeRobot can ultimately be measured by downstream task performance. Final gaze policies should therefore be optimized for task metrics, rather than hand-designed rewards like visual search. We propose a BC-RL loop to achieve this. Given a BC policy controlling a robot arm and an RL policy controlling gaze, the key idea of BC-RL is that observations flow from the RL policy to the BC policy, while task success metrics flow from the BC policy to the RL policy. At every BC-RL optimization step, the eye policy attempts to optimize the

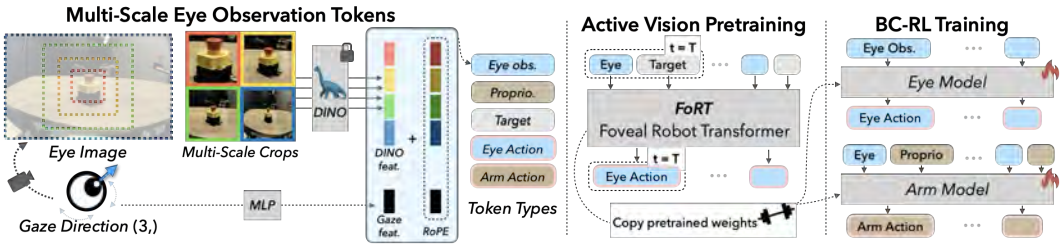

Figure 4: **Foveal Robot Transformer (FoRT)**. Eye observations are processed in a *foveal* manner, where each image is processed at multiple scales and concatenated together with the gaze direction.

BC agent's *current performance* at matching demonstrated actions, while the BC agent learns how to perform the task best given the current gaze behavior.

We operationalize BC-RL in EyeGym using the synchronized demonstration and 360 video pairs detailed in Section 3.1. At each step, the eye receives a reward comparing the predicted action chunk to the ground-truth action chunk. We use a reward based on the action chunk's forward-kinematics, which is better normalized than joint-space error. Specifically, we compute the end effector position over predicted and ground-truth trajectories, and assign reward to be the negative Fréchet distance between these splines (penalizing deviation). At the beginning of each demonstration, we pause time for 30 frames to allow the eye to visually search before advancing time in the video. The BC agent is trained with a standard L1 loss between predicted and ground truth chunks.

Active Visual Pretraining (AVP) Like other systems that incorporate behavior cloning, a key bottleneck of BC-RL is limited demonstration quantity. Pretraining offers one way to improve learning efficiency. Inspired by the importance of object search in large-workspace robot manipulation, we use visual object search reward as a pretext task for pretraining. We pretrain policies in EyeGym using static video frames sampled randomly from demonstrations. The eye is initialized at the start of each episode randomly in $\pm 90°$ and $\pm 15°$ from the neutral azimuth and elevation positions. During pretraining, we condition on image feature embeddings of search objects to give networks the ability to learn about multiple objects; this input is replaced with zero tokens after pretraining.

3.3 Foveal Robot Transformer

EyeRobot uses transformer architectures for its eye and robot policies (Fig. 4), which convert all observations to tokens and predicts all outputs as tokens. The eye and robot policies share projection matrices for shared inputs, but otherwise use separate transformer weights.

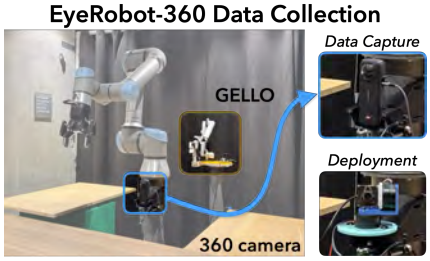

Figure 5: **Teleop Data Setup with EyeRobot.** We collect actions with GELLO [60], and the EyeGym environment with a time-synced 360 camera.

Observation Feature Extraction Though sophisticated mechanisms for multi-resolution foveation have been proposed in prior vision work [46, 61], we wish to leverage pretrained vision backbones and thus opt for a simpler architecture involving multi-cropping. We process input images into an image pyramid of crops with N scales centered at the center pixel, rescaling all images to the same 224 resolution. We embed all patches independently with a frozen DINOv2-ViT/S [62] encoder, and flatten them token-wise as inputs to the transformer. Each policy additionally takes as input the eye gaze direction as a 3D vector, the current joint proprioception, and an optional "target" token for visual search. Each are projected to the input token dimension with small MLPs. We apply 10% dropout on the proprioceptive tokens to avoid overfitting. Input tokens are positionally embedded with RoPE [63] to enable batch-parallel training with sliding window attention (see the Appendix for details).

Outputs Action outputs are decoded from the transformer with lightweight projection heads. Eye actions are parameterized as a categorical distribution over 8 azimuth-elevation directions and $\vec{0}$.

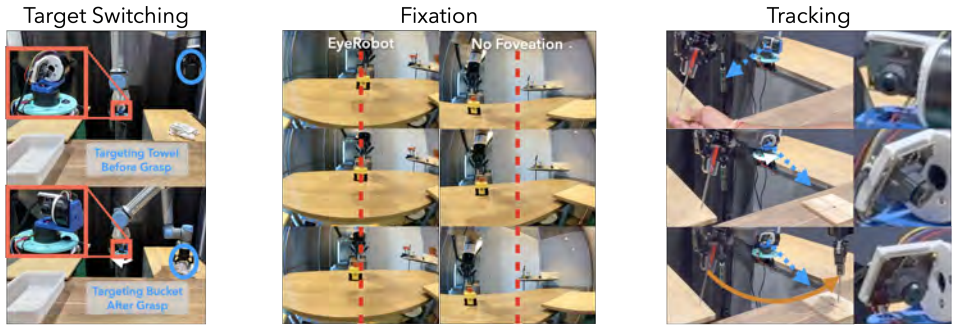

Figure 6: **Emergent Eye Behavior.** (Left) Gaze shifts from towel to bucket during grasping. (Middle) Gaze centers on E-stop with foveation. (Right) Gaze guides arm to align screwdriver to wood.

We use a learnable input token for each output type, which is shared across timesteps in the output action chunk.

Attention Masking and Memory EyeRobot masks self-attention to increase the throughput of inference and minimize overfitting. First, we mask all image-image attention since DINOv2 outputs have already undergone self-attention. Second, we utilize sliding window attention to train on long rollouts efficiently (up to 100 time-steps), with a window size of 10 for the eye to provide history of motion, and 1 for the hand to effectively make it single-frame. We use Flex Attention [64] to efficiently compute these custom masks.

4 Experiments

Our experiments aim to (1) evaluate the ability of using RL to visually search in scenes and find objects, (2) compare EyeRobot's performance on manipulation tasks to conventional camera placements, and (3) investigate emergent properties of the hand-eye coordination learned during BC-RL.

4.1 Evaluating EyeGym on Visual Search Tasks

Here we perform a series of RL-only experiments to evaluate the effectiveness of the EyeGym at training agents to look around, after which we can confidently move to using the EyeGym for training BC-RL policies.

Table 1: **Scene Search Results.**

Method	CLIP ↑	Exact Match ↑
Random Walk	0.629	28.1%
Distance Reward	0.704	64.8%
DINO Reward	0.715	60.2%
DINO+Distance	0.711	66.5%

Scene Search We train semantic visual search policies on 2,000 360° images from [57]. For each training episode, we select a randomly located target crop with a field of view between 10° and 65° and condition FoRT on pooled extracted DINO tokens. To reflect the physical limitations of our gimbal, we constrain both horizontal and vertical movements, preventing full wrap-around. We experiment with different rewards during training. Evaluation is performed on 500 unseen images, each with 24 equally space target crops, and results are reported in Table 1. We move the target in an S-shaped pattern, and for each new location, we allow the policy 20 steps to move before evaluating CLIP [65] similarity and "Exact Match" rate, which marks how often it can put the target object within its field of view at the end of the rollout. We find that both the DINO reward and Truncated Distance Reward lead to successful search behaviors, which we show videos of in the supplement. Distance reward increases the chances of finding an exact match of the target object.

Object Search We define a task where the goal is to locate target object(s) that the robot may interact with. We investigate this setting using self-captured videos from robot demonstrations, train with the "Truncated Distance" reward, and deploy the learned policy to our physical eyeball. In one evaluation, we train the robot to locate the towel and assess its ability to find it. The towel is deliberately placed outside the initial field of view, requiring the eyeball to actively explore the

scene. We achieve a success rate of 87% with an average search time of 1.8 seconds, with failures occurring only when the towel lies near the far edges of the workspace.

4.2 Robot Manipulation

We evaluate EyeRobot on 5 tasks (Fig. 7) which involve manipulation over a 210° workspace, to probe the limits of hand-eye coordination over a large region. All experiments are performed on a UR5e with a Robotiq gripper. Data was collected using GELLO [60], totaling 100-500 demos for each task. This workspace represents a significant challenge: some tasks require tolerance on the order of cm, while the workspace is on the order of $1000cm^2$. For all trials, the robot is programmatically reset to the same pose (shared across ablations and baselines). We evaluate on 5 large-workspace manipulation tasks: eraser-on-shelf, e-stop reaching, brush handoff, screwdriver servoing, and towel-in-bucket. See Appendix for details.

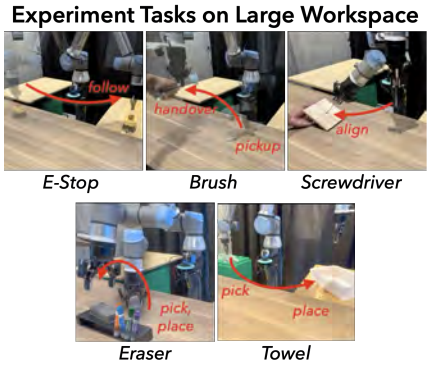

Figure 7: **Tasks.** We evaluate EyeRobot on 5 large-workspace tasks.

Results Results are reported in Tables 2, 3, 4 and results are best viewed in the included execution videos to better understand qualitative performance. EyeRobot can consistently perform manipulation tasks over a large workspace, For the **Towel** task, Most of EyeRobot's failures are in narrowly missing grasps on the towel, and we notice it tends to struggle more in switching gaze from bucket to towel, as evidenced by worse performance in trials where only the bucket is visible. Sometimes it also grasps only a towel corner, leading to a difficult bucket drop where half the towel dangles outside the bucket. In our **E-Stop** trials, EyeRobot never loses track of the object, and its main error is in z-distance towards the eyeball, owing to the difficulty of resolving depth with a monocular viewpoint. In the **Eraser** task, EyeRobot primarily fails by narrowly missing grasps on the eraser. In the place only trials where the eraser begins in the same location each time, EyeRobot can robustly follow the perturbed location of the shelf post-grasp. In **Screwdriver**, EyeRobot achieves a mean error of 4.0cm when the target is flat, and 5.2cm when the target is tilted 45°, compared to a total test area spanning 115cm left to right. The **Brush** task successfully grasps the brush at the correct orientation 15/20 trials, and successfully completed the human handover 14/15 times.

4.2.1 Eye Behavior

We observe three emergent behaviors that the eye learns while being rewarded for mimicking demonstrations: switching, search, and independent tracking. In multi-step tasks, the eye learns to automatically switch its gaze towards the next relevant object, depending on the state of the robot (Fig 1). For long-horizon tasks this requires that the eye search for the subsequent target objects when it is not in view, or even just make smaller fixation adjustments for objects that are only partially visible (Fig 6). Its search strategy tends to oscillate back and forth in a sweeping motion. When the target location is moved mid placement (e.g., 'perturb' condition in Table 2) the eye tracks the new target location, independently of the robot arm. Finally, in more dynamic tasks, we note that the eye learns to attend to predictive cues in the environment (e.g., humans placing a target object). Taken together, these qualitative results suggest the potential of an active vision agent trained with RL to naturally complement a behavior cloning agent in achieving tasks.

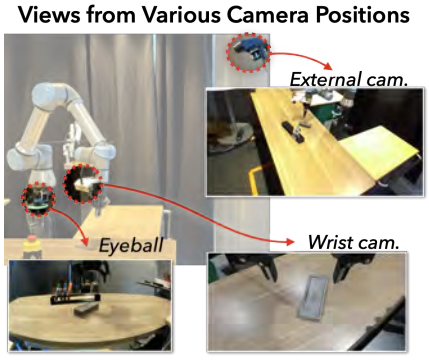

Figure 8: **Effect of Camera Placement.** Placing the camera on a gimbal allows it to observe across the whole workspace at a higher resolution.

Table 2: **Camera Comparisons**. We report success rate (Eraser & E-Stop) and distance to target (E-Stop).

Task	EyeRobot	Exo	Wrist	Wrist+Exo	Task	EyeRobot	Exo	Wrist	Wrist+Exo
Eraser (Pick & Place)	60%	0%	**100%**	60%	Eraser (Perturb Place)	**100%**	-	10%	40%
E-Stop (Slow)	**100%**	**100%**	80%	**100%**	E-Stop (Fast)	**100%**	**100%**	60%	**100%**
E-Stop (Slow)	**4.0cm**	7.8cm	5.3cm	7.1cm	E-Stop (Fast)	**4.7cm**	9.9cm	**4.7cm**	5.5cm

4.2.2 Wrist, Exo Comparisons

An important question is how EyeRobot compares to more conventional camera placements, namely exo- and wrist-mounted cameras (Fig. 7). We perform comparisons on the E-Stop and Eraser tasks, by training on *the exact same data* with the same architecture. For exo-only and wrist-only baselines, we input images at 640 resolution, while for wrist+exo we use 360. Both comparisons have *more* input image tokens than FoRT, and the same number of model parameters.

Results are reported for the eraser and e-stop tasks in Table 2. When the wrist camera can view the eraser, the wrist camera achieves a very high success rate, although this performance greatly suffers in the perturbation trials where the wrist cannot search. In contrast, EyeRobot can adjust its viewpoint to maintain the shelf in view at all times. The exo camera baseline struggles to achieve precise enough grasps given its low resolution; while often touching the eraser, it never successfuly grasps it. The exo+wrist baseline also performs worse at servoing compared to wrist-only, likely due to the complexity of merging image tokens from multiple cameras–though it can occasionally succeed in locating the perturbed shelf.

4.3 Ablations

No Foveation: To understand the contribution of foveated inputs on EyeRobot's behavior, we train and evaluate a BC-RL gaze model that operates over a uniform image resolution. To control for the number of image tokens we use inputs of 1x448x448 instead of 4x224x224. We find this model does not exhibit the qualitative behaviors that are evident in foveated models; it only loosely maintains the target object within its general field of view (Fig. 6). Quantitatively, this uniform image resolution leads to degraded performance on all metrics evaluated for E-Stop, while settling significantly slower due to its un-stable viewpoint (Table 3). These data suggest potential performance benefits in adopting a foveal architecture for manipulation owing to the arisal of fixation.

Table 3: E-Stop Ablations (Error ↓ / Speed ↓)

Task	EyeRobot	No Foveal
E-Stop (Slow)	**4.0cm / 4.2s**	5.9cm / 5.3s
E-Stop (Fast)	**4.7cm / 5.8s**	6.9cm / 7.1s
Average	**4.4cm / 5.0s**	6.4cm / 6.2s

Table 4: Towel Ablations (Success Rate)

Visible At Start	EyeRobot	No AVP
Both Visible	**80%**	73%
Towel Visible	**95%**	40%
Bucket Visible	50%	**70%**
Neither Visible	**60%**	**60%**
Average	**72.2%**(±13.1%)	**62.1%**(±14.2%)

No Active Visual Pretraining: We ablate AVP on the towel task, and surprisingly find that long-range visual search can emerge *purely* from task driven BC-RL. We note, however, that grasping performance suffers compared to the model with AVP, which we attribute partially to poorly initialized network weights and partially to poor training data distribution early in BC-RL training, as the eye fails to fixate correctly early in training.

5 Conclusion

EyeRobot presents a method for training a mechanical eyeball to look around to achieve physical robot manipulation via a real-to-sim-to-real pipeline utilizing 360° videos. This simulation environment allows the training of active visual policies with RL on top of teleop demonstrations, with which we train a visual agent to look around to maximize task-based BC performance. We find this agent learns to look to facilitate action, resulting in emergent eye behaviors such as search and fixation, and that the eye enables manipulation across a large workspace.

6 Limitations

The primary limitation of EyeRobot is the inability of 360 video to simulate motion parallax, i.e, what a neck would provide. In addition, adding depth cues like stereo would require physically changing the setup and incorporating another eye, or using monocular depth estimators during EyeGym simulation. Alternatively, depth could be added as a separate fine-tuning stage after active vision. Another drawback of EyeRobot is its eagerness to learn strategies which match simulation very well, but fail in real due to narrow data distributions. Concretely, one behavior we observe in the towel task is "blind grasping", where the robot will sometimes look left, and upon observing no towel, grasps an average location to the right. This arises from the demonstration data distribution, owing to the fact there are fewer demos at the edges of the workspace. Finally, training BC-RL converges significantly slower than vanilla BC. This is because of the co-training of two models which are mutually used in the others' train loop. We are currently limited to a stationary workstation, but an exciting future direction would be to mount the eyeball on a mobile robot, further increasing the need for active vision.

References

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

[2] R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):966–1005, 1988.

[3] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42:177–196, 2018.

[4] K. Y. Goldberg and R. Bajcsy. Active touch and robot perception. *Cognition and Brain Theory*, 7(2):199–214, 1984.

[5] J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *International journal of computer vision*, 1:333–356, 1988.

[6] N. P. Papanikolopoulos, P. K. Khosla, and T. Kanade. Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE transactions on robotics and automation*, 9(1):14–35, 1993.

[7] N. Papanikolopoulos, P. K. Khosla, and T. Kanade. Vision and control techniques for robotic visual tracking. In *ICRA*, pages 857–864, 1991.

[8] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017.

[9] X. Ye, Z. Lin, H. Li, S. Zheng, and Y. Yang. Active object perceiver: Recognition-guided policy learning for object searching on mobile robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6857–6863. IEEE, 2018.

[10] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

[11] D. Jayaraman and K. Grauman. Learning to look around: Intelligently exploring unseen environments for unknown tasks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1238–1247, 2018.

[12] R. Zeng, Y. Wen, W. Zhao, and Y.-J. Liu. View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media*, 6(3):225–245, Sep 2020. ISSN 2096-0662. doi:10.1007/s41095-020-0179-3. URL https://doi.org/10.1007/s41095-020-0179-3.

[13] J. E. Banta, L. Wong, C. Dumont, and M. A. Abidi. A next-best-view system for autonomous 3-d object reconstruction. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(5):589–598, 2000.

[14] M. Mendoza, J. I. Vasquez-Gomez, H. Taud, L. E. Sucar, and C. Reta. Supervised learning of the next-best-view for 3d object reconstruction. *Pattern Recognition Letters*, 133:224–231, 2020.

[15] E. Smith, D. Meger, L. Pineda, R. Calandra, J. Malik, A. Romero Soriano, and M. Drozdzal. Active 3d shape reconstruction from vision and touch. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 16064–16078. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/8635b5fd6bc675033fb72e8a3ccc10a0-Paper.pdf.

[16] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484, 2016. doi:10.1109/ICRA.2016.7487527.

[17] R. Monica and J. Aleotti. Surfel-based next best view planning. *IEEE Robotics and Automation Letters*, 3(4):3324–3331, Oct 2018. ISSN 2377-3766. doi:10.1109/LRA.2018.2852778.

[18] R. Zeng, W. Zhao, and Y.-J. Liu. Pc-nbv: A point cloud based deep network for efficient next best view planning. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7050–7057, 2020.

[19] V. Sripada, S. Carter, F. Guerin, and A. Ghalamzan. Ap-vlm: Active perception enabled by vision-language models. *arXiv preprint arXiv:2409.17641*, 2024.

[20] X. Cheng, J. Li, S. Yang, G. Yang, and X. Wang. Open-television: Teleoperation with immersive active visual feedback. *arXiv preprint arXiv:2407.01512*, 2024.

[21] I. Chuang, A. Lee, D. Gao, M. Naddaf-Sh, I. Soltani, et al. Active vision might be all you need: Exploring active vision in bimanual robotic manipulation. *arXiv preprint arXiv:2409.17435*, 2024.

[22] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Survey: Robot programming by demonstration. *Springer handbook of robotics*, 2008.

[23] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys*, 2017.

[24] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 2020.

[25] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[26] Y. Du, D. Ho, A. Alemi, E. Jang, and M. Khansari. Bayesian imitation learning for end-to-end mobile manipulation. In *International Conference on Machine Learning*, pages 5531–5546. PMLR, 2022.

[27] Z. Fu, T. Z. Zhao, and C. Finn. Mobile ALOHA: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv:2401.02117*, 2024.

[28] T. Lin, Y. Zhang, Q. Li, H. Qi, B. Yi, S. Levine, and J. Malik. Learning visuotactile skills with two multifingered hands. *arXiv preprint arXiv:2404.16823*, 2024.

[29] J. Grannen, Y. Wu, B. Vu, and D. Sadigh. Stabilize to Act: Learning to coordinate for bimanual manipulation. In *CoRL*, 2023.

[30] 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. *arXiv:2402.10329*, 2024.

[31] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[32] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

[33] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. $\pi 0$: A vision-language-action flow model for general robot control, 2024. *URL https://arxiv. org/abs/2410.24164*.

[34] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

[35] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson. Implicit behavioral cloning. In *Conference on robot learning*, pages 158–168. PMLR, 2022.

[36] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[37] D. Marr, T. Poggio, E. C. Hildreth, and W. E. L. Grimson. *A computational theory of human stereo vision*. Springer, 1991.

[38] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka. A stereo machine for video-rate dense depth mapping and its new applications. In *Proceedings CVPR IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 196–202. IEEE, 1996.

[39] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 1, pages I–I. IEEE, 2003.

[40] H. Rebecq, R. Ranftl, V. Koltun, and D. Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1964–1980, 2019.

[41] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J. Conradt, K. Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020.

[42] H. Kim, S. Leutenegger, and A. J. Davison. Real-time 3d reconstruction and 6-dof tracking with an event camera. In *European conference on computer vision*, pages 349–364. Springer, 2016.

[43] C. Bandera and P. D. Scott. Foveal machine vision systems. In *Conference Proceedings., IEEE International Conference on Systems, Man and Cybernetics*, pages 596–599. IEEE, 1989.

[44] S. Minut, S. Mahadevan, J. M. Henderson, and F. C. Dyer. Face recognition using foveal vision. In *Biologically Motivated Computer Vision: First IEEE International Workshop, BMCV 2000 Seoul, Korea, May 15–17, 2000 Proceedings 1*, pages 424–433. Springer, 2000.

[45] G. Killick, P. Henderson, P. Siebert, and G. Aragon-Camarasa. Foveation in the era of deep learning. *arXiv preprint arXiv:2312.01450*, 2023.

[46] A. Jonnalagadda, W. Y. Wang, B. Manjunath, and M. P. Eckstein. Foveater: Foveated transformer for image classification. *arXiv preprint arXiv:2105.14173*, 2021.

[47] B. Cheung, E. Weiss, and B. Olshausen. Emergence of foveal image sampling from learning to attend in visual scenes. *arXiv preprint arXiv:1611.09430*, 2016.

[48] E. Rivlin and H. Rotstein. Control of a camera for active vision: Foveal vision, smooth tracking and saccade. *International Journal of Computer Vision*, 39:81–96, 2000.

[49] S. Gould, J. Arfvidsson, A. Kaehler, B. Sapp, M. Messner, G. R. Bradski, P. Baumstarck, S. Chung, A. Y. Ng, et al. Peripheral-foveal vision for real-time object recognition and tracking in video. In *Ijcai*, volume 7, pages 2115–2121. Citeseer, 2007.

[50] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2023.

[51] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[52] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[53] K. Zakka, B. Tabanpour, Q. Liao, M. Haiderbhai, S. Holt, J. Y. Luo, A. Allshire, E. Frey, K. Sreenath, L. A. Kahrs, et al. Mujoco playground. *arXiv preprint arXiv:2502.08844*, 2025.

[54] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9339–9347, 2019.

[55] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021.

[56] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. W. Clegg, M. Hlavac, S. Y. Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

[57] S.-H. Chou, C. Sun, W.-Y. Chang, W.-T. Hsu, M. Sun, and J. Fu. 360-indoor: Towards learning real-world objects in 360deg indoor equirectangular images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 845–853, 2020.

[58] M. Wallingford, A. Bhattad, A. Kusupati, V. Ramanujan, M. Deitke, A. Kembhavi, R. Mottaghi, W.-C. Ma, and A. Farhadi. From an image to a scene: Learning to imagine the world from a million 360° videos. *Advances in Neural Information Processing Systems*, 37:17743–17760, 2024.

[59] H.-N. Hu, Y.-C. Lin, M.-Y. Liu, H.-T. Cheng, Y.-J. Chang, and M. Sun. Deep 360 pilot: Learning a deep agent for piloting through 360 sports videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1396–1405. IEEE, 2017.

[60] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12156–12163. IEEE, 2024.

[61] A. Deza and T. Konkle. Emergent properties of foveated perceptual systems. *arXiv preprint arXiv:2006.07991*, 2020.

[62] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[63] J. Su, Y. Lu, S. Pan, A. Murtadha, B. Wen, and Y. Liu. Roformer: enhanced transformer with rotary position embedding. arxiv. *arXiv preprint arXiv:2104.09864*, 2021.

[64] J. Dong, B. Feng, D. Guessous, Y. Liang, and H. He. Flex attention: A programming model for generating optimized attention kernels, 2024. URL https://arxiv.org/abs/2412.05496.

[65] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.