"Human" Vision for Bimanual Manipulation

https://ian-chuang.github.io/human-vision
https://github.com/ian-chuang/human-vision

Ian Chuang
University of California, Berkeley
ianc@berkeley.edu

Abstract

With recent advances in imitation learning, there is a growing interest in teaching robots complex behaviors directly from visual input. Recent works in this area mostly focus on improving action generation and heavily rely on input from raw camera images, which differ substantially from how humans use vision—particularly the role of gaze in directing attention and fixation. Given the superior dexterity of humans, it is reasonable to consider how robotic vision systems might benefit from mimicking human visual processing. We introduce a robot simulation platform for simultaneously collecting eye-tracking data and robot demonstrations from a human operator. We also present simple architectural modifications to a standard imitation learning pipeline to integrate gaze information, specifically by cropping images around the gaze point and incorporating a long history of gaze observations. These modifications improve overall task performance, enhance robustness to unseen distractors, and significantly reduce computational overhead. Videos and code can be found here: https://ian-chuang.github.io/human-vision

1. Introduction

Imitation learning has emerged as a powerful approach for enabling dexterous robot behaviors in complex systems, such as bimanual manipulation [15] and humanoid control [14]. These methods typically process camera images and robot proprioception to directly produce robot actions in an end-to-end manner [1]. However, their approach to visual input differs significantly from how humans utilize vision. Human vision relies on gaze to focus on critical information, reducing cognitive load. Moreover, gaze behavior provides important insights into human decision-making, with sequences of past fixations playing a crucial role in how humans perceive and act on their environment. This raises the question: can robotic vision systems become more human-like by leveraging human gaze patterns during learning?

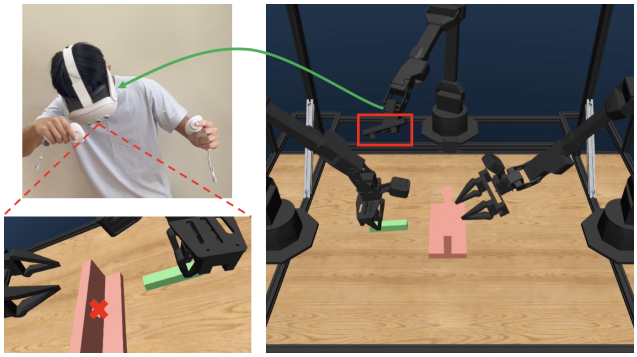

Figure 1. We introduce a method for simultaneously collecting robot demonstrations and recording eye-tracking data using a VR headset.

One promising avenue is the growing use of Virtual Reality (VR) headsets for collecting demonstrations in imitation learning. Modern VR headsets often feature built-in eye-tracking capabilities, enabling the simultaneous recording of gaze data and robot demonstrations. Capturing gaze data alongside motor actions offers valuable supervision, providing insights into where to focus attention. By learning not only from a demonstrator's actions but also from their visual attention to the scene, we can take steps toward developing robotic vision systems that more closely emulate humans.

How can we effectively leverage human gaze for imitation learning? In this work, we introduce a robot simulation that allows for collection of human demonstrations and eye-tracking data in an efficient and accessible manner using a VR headset. Building on prior work, AV-ALOHA [2], where we developed a robot platform to learn camera perspectives from human demonstrations, we extend this approach to also learn from human gaze behavior. We present modifications to a flow-matching imitation learning policy to address key limitations in robotic visual perception: (1)

minimizing unnecessary visual processing and (2) enabling the use of an extended gaze history. Preliminary experiments demonstrate various benefits, including improved overall performance, reduced memory usage, and shorter training times.

2. Related Work

2.1. Human Vision

Human vision is organized around the division between foveal and peripheral vision [10]. Foveal vision, centered at the gaze, provides high-resolution focus within a small area, making it ideal for understanding fine details. In contrast, peripheral vision encompasses the area outside the central gaze, offering low resolution but a broader field of view, which is well-suited for general awareness of surroundings. This division allows humans to reduce visual processing demands by focusing on important elements while filtering out distractions.

One important observation in the study of human vision is that human eyes are naturally directed toward areas where useful information is expected in the near future. This behavior has been demonstrated across various everyday tasks, such as making tea [6] and driving [13], showing how gaze precedes and guides human arm movements. This raises the question of what insights can be drawn from the fixation patterns of human gaze.

In this work, we aim to incorporate these distinct characteristics of human vision into robot learning.

2.2. Gaze for Robot Learning

The use of gaze in robot learning remains a relatively underexplored area. One study improves representation learning in reinforcement learning by pretraining masked autoencoders to reconstruct saliency maps [7]. Closely related to our approach is a line of work that incorporates gaze data into imitation learning. The initial work used Mixture Density Models to predict gaze points, cropping the corresponding regions of interest, and feeding these cropped regions into a policy [4]. Subsequent works include transitioning from low-resolution full images to high-resolution selective crops for tasks demanding greater precision [5], as well as segmenting trajectories into subgoals based on gaze data [11].

In this work, we present a simpler and more accessible solution, introducing a robotic platform that integrates both camera perspective and gaze control, while avoiding complex architectural modifications. Beyond cropping, we explore how leveraging gaze history can help policy learning by compressing and extending our visual memory.

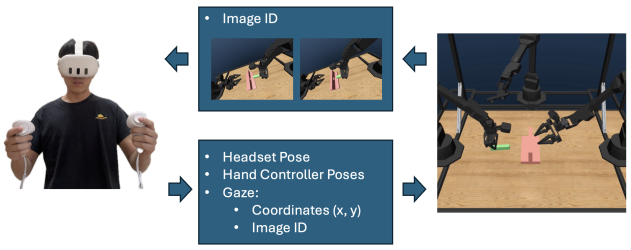

Figure 2. The robot transmits stereo camera images to the VR headset, appending an image ID to each frame as metadata. The VR headset sends back head and hand controller poses to control the robot, along with gaze coordinates and the corresponding image ID, synchronizing gaze data with the images.

3. Method

3.1. Data Collection with Eye Tracking

For collecting robot demonstration data with eye tracking we use a MuJoCo [12] simulation environment from our previous work, AV-ALOHA [2]. The setup includes two robotic arms for bimanual manipulation and a third arm equipped with a stereo camera, which can move freely to observe the scene. The entire system is controlled by a user wearing a VR headset, enabling simultaneous control of all three arms through head and hand movements.

Communication between the VR headset and the robot is facilitated via the WebRTC protocol. The VR headset streams head and hand pose data to the robot, which are then converted to joint commands using inverse kinematics. The VR headset also streams eye tracking data which are recorded by the robot. The robot streams images from its stereo camera to the headset, which are displayed to the user's left and right eye to provide a sense of depth.

One key consideration is the latency inherent in streaming data, particularly if the user is controlling the robot system remotely over long distances. This latency can cause misalignment between the eye-tracking data and the corresponding images. To address this, we annotate each image frame sent from the robot to the VR headset with a unique ID. When the VR headset streams head and hand pose data to the robot, it also sends eye-tracking data tagged with the corresponding image ID, ensuring proper synchronization. For images that are not labeled in time with the corresponding gaze, we interpolate between known eye-tracking labels to approximate the gaze data. Figure 2 illustrates the details of the information exchange process.

When collecting data, we record the robot's camera images, joint states, actions, and human gaze coordinates.

3.2. Incorporating Gaze

For the imitation learning policy, we design a custom approach based on flow-matching [8]. We choose flow-matching over a diffusion-based policy due to its faster in-

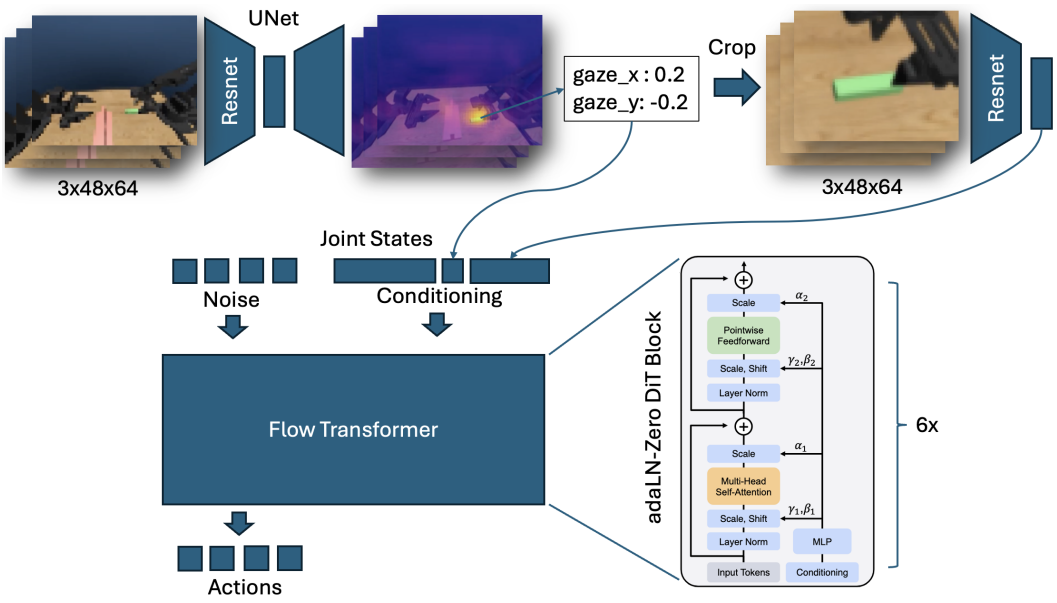

Figure 3. Model Architecture: a sequence of low-resolution images is passed through a UNet, which generates a heatmap. A spatial softmax is applied to the heatmap to extract gaze coordinates, and a loss is computed between the predicted gaze and the ground truth human gaze during training. Next, the image is cropped at the predicted gaze location, and the cropped region is encoded to extract image features. These features and predicted gaze are then used as conditioning inputs to a flow transformer, which predicts the robot's actions.

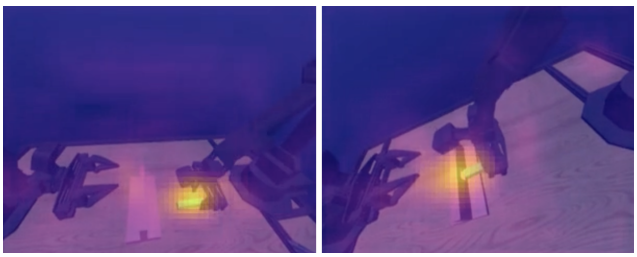

Figure 4. Example gaze heatmaps generated by the UNet after training, used to predict gaze coordinates via spatial softmax.

ference time, requiring fewer denoising steps—in our case, only 6 steps. Additionally, we use an AdaLN-Zero Diffusion Transformer [9] for the denoising network, which has shown to be more stable during training [3]. The full architecture is illustrated in Figure 3.

To integrate gaze into imitation learning, we propose adding two key features: the ability to focus and retaining a memory of previous gaze observations. Our approach emphasizes simple, practical solutions that can be applied across a variety of tasks.

To enable the system to focus, we adopted a method inspired by [4], where the image is cropped based on gaze predictions. We process a downscaled version of the robot's camera image through a UNet with a ResNet backbone to generate a heatmap, followed by a spatial softmax operation on the heatmap to extract a keypoint that represents the predicted gaze. During training, this keypoint is supervised using the ground truth human gaze coordinates with a mean squared error loss. We then apply a differentiable cropping mechanism to extract a region around the gaze, with the crop size set to 48×64 pixels in our implementation. The cropped image is then processed by a ResNet encoder to extract features. These features, along with the gaze coordinates and joint state positions, are concatenated and used to condition the denoising network. Examples of predicted heatmaps generated by the UNet are illustrated in Figure 4.

To account for temporal context beyond the current observation, we extended the model to process a sequence of images from previous timesteps. For each image in the sequence, the model predicts the gaze, crops the image around the predicted gaze, extracts features, and concatenates the sequence of features with the gaze predictions and joint states to create the condition. By integrating a history of gaze, the model can retain critical fixation points, particularly during transitions between different gaze targets. An example of an observation sequence is shown in Figure 5.

4. Experiments

4.1. Robot Task

We conduct preliminary experiments focusing on a single task from the AV-ALOHA paper [2], referred to as the *Thread Needle* task. This is a simulation task where the robot's right arm grasps a needle and threads it through a

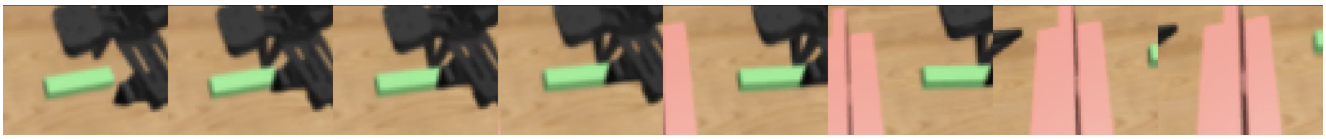

Figure 5. Example of input to the flow policy with an observation history size of 8, cropped at the gaze. The oldest observations are on the left. At this moment, the gaze transitions from focusing on picking up the green thread to focusing on the pink object.

Figure 6. Visualization of a human demonstration for the thread-needle task, including recorded human gaze. In this task, the robot picks up the green needle and threads it through a pink object.

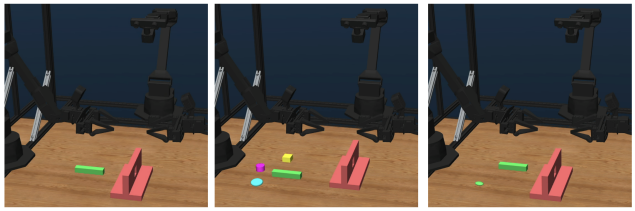

Figure 7. (*Left*) Thread-needle task without distractors (*in-distribution*), (*middle*) task with distractors of different colors (*multiple-distractors*), and (*right*) task with a distractor of the same color as the needle (*adverse-distractor*).

hole in an object, allowing the left arm to grab and pull it out from the opposite side. The object is positioned such that the hole has limited visibility from the static cameras in the setup. A visualization of the task is shown in Figure 6.

We evaluate the task across three different variants of the environment: an environment identical to the one used during data collection (*in-distribution*), a variation featuring three differently colored objects placed near the needle (*multiple distractors*), and another variation with a single object that shares the same color as the needle but has a different shape (*adverse distractor*). These variants are illustrated in Figure 7.

4.2. Results and Discussion

We collected 100 human demonstrations of this task, including eye-tracking data. The flow policy was trained for 100,000 steps using a learning rate of 1e-4, cosine scheduling, and a batch size of 64. The image input size was 3×240×320, with a cropped region of 3×48×64 centered at the gaze coordinates.

For evaluation, we tested the model at different checkpoints, performing evaluations every 10,000 training steps. For each evaluation, we run 50 rollouts across the *in-distribution*, *multiple-distractors*, and *adverse-distractor* environments and record overall success rate.

Method	Training Time	GPU Memory Usage
crop+8obs	8h 5m	5762 MiB
crop+1obs	4h 25m	2226 MiB
8obs	12h 15m	30720 MiB
1obs	4h 20m	5194 MiB

Table 1. Comparison of training time and GPU memory usage for 100,000 training steps and batch size of 64.

We performed a series of ablation studies to evaluate the impact of cropping (*crop*) and using different observation history lengths, either 1 (*1obs*) or 8 (*8obs*). The tested combinations included *1obs*, *8obs*, *crop+1obs*, and *crop+8obs*. The results are presented in Figure 8, with additional details on training time and GPU memory usage provided in Table 1.

We achieve better overall performance with *crop+8obs* compared to both *8obs* and *1obs*. For in-distribution scenarios, *crop+8obs* performs slightly better or is comparable to the no-cropping approach. However, its advantage becomes particularly evident in unseen environments with distractors, where *crop+8obs* significantly outperforms the others. This suggests that cropping based on predicted gaze helps the policy focus on critical information while disregarding distractors. Additionally, at the beginning of training, the success rate for both *crop+8obs* and *crop+1obs* is significantly higher than *8obs* and *1obs* across all task variants. This suggests that cropping at the predicted gaze is more sample-efficient.

One key observation is that a longer observation history significantly improves performance, particularly when cropping is applied. Across all task variants, *crop+8obs* performed substantially better than *crop+1obs*. While no cropping with *8obs* also showed performance gains compared to *1obs*, the improvement was less pronounced compared to the cropping methods. This highlights the importance of observation history, especially when cropping at the predicted gaze.

Although *8obs* occasionally matched the performance of *crop+8obs* (despite having lower overall performance), it is important to note that *8obs* consumes nearly 5x more GPU

Figure 8. Comparison of success rates between using cropping at gaze (*crop*) and no cropping at gaze, with variations including using only the current observation (*1obs*) or an observation history of size 8 (*8obs*), evaluated at different checkpoints during training.

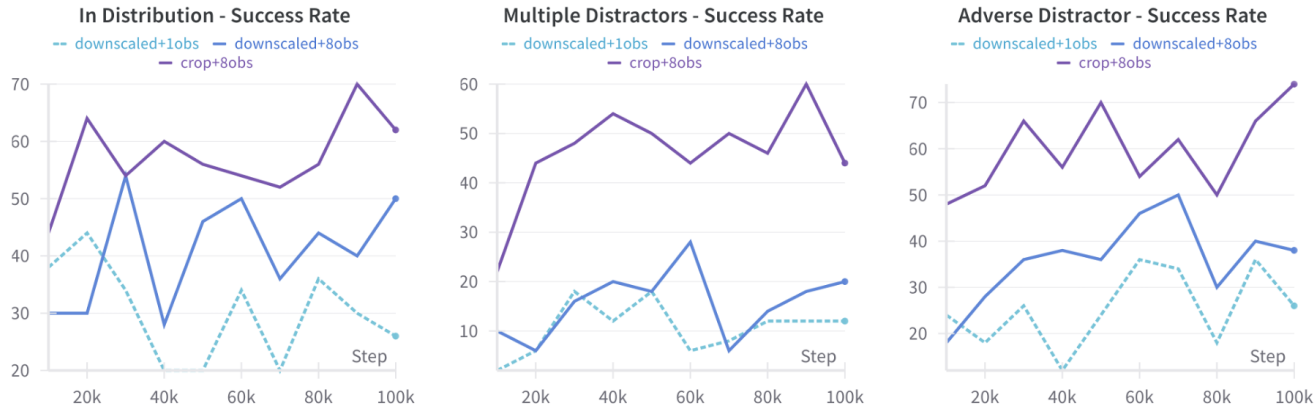

Figure 9. Comparison of success rates between our method, which involves cropping at gaze and using an observation history of size 8, and an alternative approach that forgoes cropping but downsizes the image to match the crop's dimensions.

memory and takes 1.5x longer to train. This means that even when performance was comparable for in-distribution tasks, *crop+8obs* achieved similar results with significantly fewer computational resources.

To address the issue of long training times and high GPU memory usage with *8obs*, we conducted experiments using uncropped images resized to 48x64, significantly lowering resolution and visual detail. These experiments, termed *downscaled+1obs* and *downscaled+8obs*, aim to determine whether downscaled images could reduce computational overhead while maintaining performance without the need for gaze information. Results in Figure 9 show that while both gaze-cropped and downscaled images reduce image size, downscaling performs significantly worse. This highlights the importance of cropping to focus on key image regions and demonstrates that maintaining higher image quality/resolution is beneficial.

4.3. Conclusion

This work explores how to use gaze for bimanual manipulation, drawing inspiration from human vision. We introduce an open-source robot simulation for collecting robot and eye-tracking data and propose a method that mimics human gaze by focusing on relevant areas through cropping and maintaining a memory of gaze observations. Our approach achieves strong performance while significantly reducing GPU memory usage and training time. Future work will involve additional experiments and testing on a diverse set of tasks.

References

[1] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023. 1

[2] Ian Chuang, Andrew Lee, Dechen Gao, and Iman Soltani. Active vision might be all you need: Exploring active vision in bimanual robotic manipulation. *arXiv preprint arXiv:2409.17435*, 2024. 1, 2, 3

[3] Sudeep Dasari, Oier Mees, Sebastian Zhao, Mohan Kumar Srirama, and Sergey Levine. The ingredients for robotic diffusion transformers. *arXiv preprint arXiv:2410.10088*, 2024. 3

[4] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Using human gaze to improve robustness against irrelevant objects in robot manipulation tasks. *IEEE Robotics and Automation Letters*, 5(3):4415–4422, 2020. 2, 3

[5] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Gaze-based dual resolution deep imitation learning for high-precision dexterous robot manipulation. *IEEE Robotics and Automation Letters*, 6(2):1630–1637, 2021. 2

[6] Michael Land, Neil Mennie, and Jennifer Rusted. The roles of vision and eye movements in the control of activities of daily living. *Perception*, 28(11):1311–1328, 1999. PMID: 10755142. 2

[7] Anthony Liang, Jesse Thomason, and Erdem Bıyık. Visarl: Visual reinforcement learning guided by human saliency. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2907–2912. IEEE, 2024. 2

[8] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 2

[9] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 3

[10] Emma E. M. Stewart, Matteo Valsecchi, and Alexander C. Schütz. A review of interactions between peripheral and foveal vision. *Journal of Vision*, 20(12):2–2, 2020. 2

[11] Ryo Takizawa, Izumi Karino, Koki Nakagawa, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Enhancing reusability of learned skills for robot manipulation via gaze and bottleneck. *arXiv preprint arXiv:2502.18121*, 2025. 2

[12] Emanuel Todorov, Tom Erez, and Yuval 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. 2

[13] Richard Wilkie and John Wann. Controlling steering and judging heading: Retinal flow, visual direction, and extraretinal information. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):363–378, 2003. 2

[14] Yanjie Ze, Zixuan Chen, Wenhao Wang, Tianyi Chen, Xialin He, Ying Yuan, Xue Bin Peng, and Jiajun Wu. Generalizable humanoid manipulation with improved 3d diffusion policies. *arXiv preprint arXiv:2410.10803*, 2024. 1

[15] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 1