Behavior Cloning for Active Perception with Low-Resolution Egocentric Vision

Anthony Bilic[1], Chen Chen[1], and Ladislau Bölöni[1]

***Abstract*— We investigate whether behavior cloning is sufficient to produce active perception in a structured object-finding task. A low-cost robot arm equipped with a wrist-mounted egocentric RGB camera must reposition to center a partially visible plant before triggering a grasp signal, requiring actions that improve future observations. The model predicts joint commands directly from low-resolution RGB images under closed-loop control. We show that low-resolution egocentric vision is sufficient for reliable task completion and that predicting relative joint deltas substantially outperforms absolute joint position prediction in our setting. These results demonstrate that visually grounded active perception can emerge from behavior cloning in a reproducible setting.**

I. Introduction

This work investigates whether behavior cloning is sufficient to produce active perception in a structured object-finding task. Behavior cloning learns policies by imitating expert demonstrations [1], [2], [3], [4] and does not explicitly optimize for information-seeking actions. Active perception, by contrast, concerns systems in which actions are selected to deliberately influence future observations in order to enable task completion [5], [6].

To evaluate this question, a controlled object-finding experiment is constructed using a low-cost robot arm equipped with a wrist-mounted egocentric RGB camera. A target plant is initially only partially visible, requiring the robot to reposition the camera until the object is centered and fully observable before triggering a grasp signal. Successful completion therefore requires actions that improve subsequent visual input.

A neural network is trained to map low-resolution RGB observations to joint commands under closed-loop control. We consider two representation for the joint commands: absolute joint positions [4], [7] and joint deltas [3], [8], [9], where a joint delta denotes the change from the current joint configuration to the next. The impact of this action representation choice is evaluated empirically for our task.

We make the following contributions:

- We propose a simple and reproducible experimental setup using a low-cost robot arm equipped with a wrist-mounted egocentric RGB camera to evaluate active perception.
- We demonstrate that behavior cloning produces active perception behavior without explicit supervision for information gathering.

[1]Department of Computer Science, University of Central Florida, Orlando, FL, USA

Fig. 1: Left: Overhead view of experimental setup. Right: Initial observation of wrist-mounted view with the plant on the left.

- We show that low-resolution egocentric RGB input is sufficient for reliable active perception task completion under closed-loop control.
- We show that predicting relative joint deltas yields substantially better performance, smoothness, and generalization than absolute joint position prediction in this setting.

II. Method

A. Task Definition and Setup

Experiments are conducted using a table-mounted Lynxmotion AL5D robotic arm with 6 degrees of freedom and a wrist-mounted monocular RGB camera (Fig. 1). The system is built from inexpensive off-the-shelf components. Demonstrations are collected via teleoperation using an Xbox controller. The wrist camera captures 64×64 RGB images at 10 Hz, and joint positions are recorded synchronously at the same rate.

At the start of each demonstration, a plant is placed to the left or right such that only part of it is visible against an uninformative white background (Fig. 1). The task is to center the plant in view and close the gripper. Because the plant is not fully visible at initialization, the robot must execute movements that change the wrist camera viewpoint until the object becomes centered. Actions therefore influence subsequent observations, providing a controlled evaluation of active perception in an intentionally minimal and reproducible setting.

B. Approach

We train a visual encoder and temporal controller end-to-end using behavior cloning (Fig. 2). The visual encoder is a four-layer convolutional neural network (CNN) that processes individual RGB images to produce compact feature

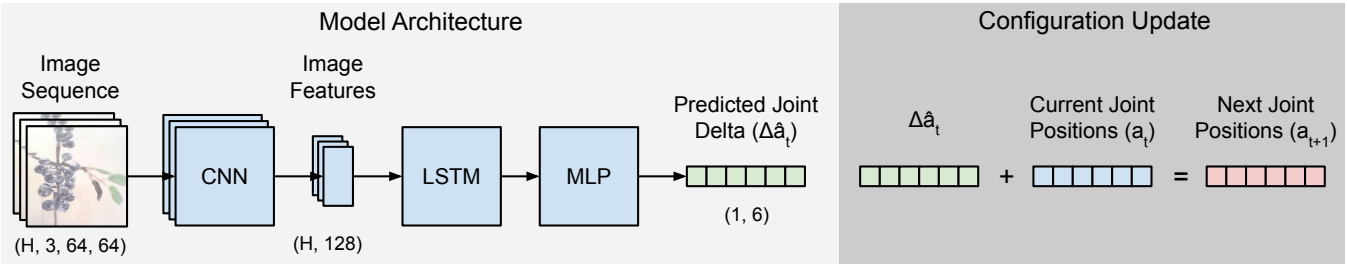

Fig. 2: Model architecture (left) and joint configuration update (right).

representations, leveraging spatial inductive bias for efficient perception. The controller is an LSTM that processes these features over time, integrating information across timesteps to capture dependencies that are not observable from a single frame.

We use a fixed-length history of images as input to the model. Training data is constructed by sliding a window of H frames across teleoperated demonstrations, where each window defines a history ending at timestep t. The supervision target for each history is the corresponding demonstrated joint delta $\Delta a_t = a_{t+1} - a_t$, representing the change in joint positions from the final observation in the history to the next timestep. The model is trained to predict joint deltas by minimizing mean squared error (MSE) between predicted and demonstrated joint deltas, $\mathrm{MSE}(\hat{\Delta a}_t, \Delta a_t)$.

During inference, the history is initialized by repeating the first observed image H times. Once initialized, the model predicts a joint delta from the current history. This delta is added to the robot's current joint positions to obtain $a_{t+1} = a_t + \Delta a_t$, which the robot executes to update its configuration. A new image is then captured from the updated state and appended to the history while the oldest is removed, maintaining a fixed-length input of H images, and the process repeats. To mirror this process during training, each demonstration is padded at the start by repeating the initial frame $H - 1$ times when constructing input histories.

III. Experiments

Table I reports the mean squared error (MSE) of joint predictions, computed in joint position space, for both joint delta and absolute position prediction across different numbers of demonstrations, together with task success rates. In the absolute position setting, the model is trained identically but with targets defined as a_{t+1} instead of $a_{t+1} - a_t$, as is commonly done in prior work. Demonstrations are divided into an 80–10–10 train/validation/test split, where training set sizes are sampled from the training portion while the validation and test sets remain fixed across experiments. Demonstrations consist of trials where the plant is placed on the left or right with equal probability, and splits are balanced accordingly. At test time, the model runs autonomously in closed loop for 10 seconds with $H = 20$. A trial is considered successful if the robot stabilizes on the plant and closes the gripper exactly once. Early or repeated closures are counted as failures.

With eight demonstrations, both models complete the task. With four demonstrations, only the delta model succeeds.

TABLE I: Mean squared error (MSE) of joint predictions (in joint position space) for joint delta and absolute position prediction across different training set sizes (Demo Count). MSE values are scaled by 10^{-3}.

	Delta			Position		
Demo Count	Train	Test	Success	Train	Test	Success
2	5.16	6.22	0/5	17.13	157.48	0/5
4	5.23	6.15	5/5	13.23	189.08	0/5
8	5.69	6.08	5/5	9.73	93.28	5/5
16	5.65	5.91	5/5	12.84	30.52	5/5
32	5.08	5.60	5/5	9.20	16.57	5/5
64	5.36	5.54	5/5	8.71	12.80	5/5

Training with joint deltas yields significantly lower test loss, which is reflected during execution. The delta model makes smaller and more consistent movements toward the target, whereas the absolute position models make larger movements and frequently overshoot before correcting. Both model's behavior becomes smoother as the number of demonstrations increases.

We also test plant placements between the fixed left and right training positions. The delta model adapts to these unseen placements when the plant is partially visible at initialization. The absolute position model instead moves toward one of the demonstrated left or right configurations once the plant comes into view. This difference stems from the action representation. The position model learns to move to a specific joint configuration given an image, whereas the delta model learns to produce a relative adjustment from its current state. This explains why delta prediction better supports adaptation to intermediate placements.

IV. Conclusion

We present a simple and reproducible setup for evaluating whether behavior cloning is sufficient for active perception. In this object-finding task, the learned model produces closed-loop behavior that repositions the camera to improve future observations. Low-resolution egocentric RGB input is sufficient for reliable task completion, and predicting relative joint deltas yields more stable and generalizable behavior than absolute joint position prediction. These results show that visually grounded active perception can emerge from behavior cloning in this setting.

Acknowledgment: This work was partly supported by the intramural research program of the U.S. Department of Agriculture, National Institute of Food and Agriculture via grant number 2024-67022-41788

REFERENCES

[1] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," in *Proc. of Int. Conf on Machine Learning (ICML-2019)*, 2019, pp. 2555–2565.

[2] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *Foundations and Trends in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[3] 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 preprint arXiv:2402.10329*, 2024.

[4] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *Proc. of Int. Conf. on Robotics and Automation (ICRA-2018)*, 2018, pp. 3758–3765.

[5] D. H. Ballard, "Animate vision," *Artificial intelligence*, vol. 48, no. 1, pp. 57–86, 1991.

[6] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.

[7] 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.

[8] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[9] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Object-centric imitation learning for vision-based robot manipulation," in *Conference on Robot Learning (CoRL-2022)*, 2022.