Hand-Eye Autonomous Delivery: Learning Humanoid Navigation, Locomotion and Reaching

Sirui Chen*
Stanford University
United States
ericcsr@stanford.edu

Yufei Ye*
Stanford University
United States
yufeiy2@stanford.edu

Zi-ang Cao*
Stanford University
United States
ziangcao@stanford.edu

Jennifer Lew
Stanford University
United States
jennlew@stanford.edu

Pei Xu
Stanford University
United States
peixu@stanford.edu

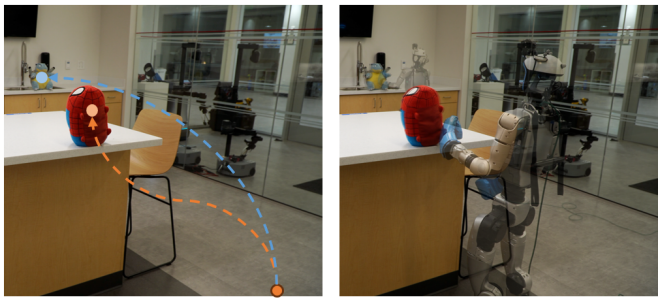
C. Karen Liu
Stanford University
United States
karenliu@cs.stanford.edu

Abstract: We propose Hand-Eye Autonomous Delivery (HEAD), a framework that learns navigation, locomotion, and reaching skills for humanoids, directly from human motion and vision perception data. We take a modular approach where the high-level planner commands the target position and orientation of the hands and eyes of the humanoid, delivered by the low-level policy that controls the whole-body movements. Specifically, the low-level whole-body controller learns to track the three points (eyes, left hand, and right hand) from existing large-scale human motion capture data while high-level policy learns from human data collected by Aria glasses. Our modular approach decouples the ego-centric vision perception from physical actions, promoting efficient learning and scalability to novel scenes. We evaluate our method both in simulation and in the real-world, demonstrating humanoid's capabilities to navigate and reach in complex environments designed for humans. Our code and more results are available at Website

Keywords: Humanoid, Hand-Eye Delivery, Learning from Human Data

1 Introduction

Any human manipulation task begins with moving close to the target object so we can perceive it and touch it. Consequently, a fundamental skill a humanoid must master is the ability to deliver its end-effectors and cameras to the place they are needed in a 3D environments designed by and for humans. One possible approach is to directly combine existing navigation methods for mobile robots and reaching methods for manipulators. However, this often leads to control strategies that either spatially isolate upper-body manipulation from lower-body locomotion, or a lack of coordination necessary for seamless transition between navigation and reaching behavior.



(a). Select goals in image (b). Robot reach the goals

Figure 1: Given a user selected goal from the humanoid's ego-centric view, the humanoid is able to navigate in the 3D world and reach the goal.

We propose a hand-eye autonomous delivery (HEAD) system for humanoids, designed to fully utilize their human-like morphology to achieve concurrent navigation, locomotion and reaching tasks in a coordinated manner. While learning from human demonstrations is a promising avenue, training all three skills end-to-end would require heterogeneous human data involving both egocentric vision and full-body motion. Instead, we adopt a modular approach that decouples egocentric perception from physical actions, enabling flexible training of whole body navigation, locomotion and reaching using different sources of human data and different algorithms. This design also mitigates the challenges of training a unified visuomotor policy. Our framework consists of a high-level policy that predicts target positions and orientations for the humanoid's eyes and hands, and a low-level controller that executes the corresponding full-body motions.

Given a command to reach for and touch an object, indicated by a point in the initial RGB image perceived by the humanoid, the high-level policy predicts head positions and orientations to guide the humanoid toward the target while keeping it in view and navigating around obstacles. Once the target is within arm's reach, the high-level policy also controls the hands to make contact with the object. Existing visual navigation methods [1, 2] often abstract the robot as a point mass, limiting actions to 2D movements on the ground plane. While suitable for wheeled robots, such assumptions are insufficient for humanoids, which must coordinate an articulated body to navigate complex 3D spaces, simultaneously reaching for and avoiding objects at varying heights. To enable this 3D navigation capability, our method leverages a mix of different datasets for different purposes—internet large-scale human exploration datasets for generalization to new scenes, mid-scale demonstrations in the target environment for mitigating domain shift due to perception, and a small amount of robot-specific experience for mitigating domain shift due to embodiment gap.

The low-level whole-body controller is trained to track three key points—the eyes, left hand, and right hand—using large-scale human motion capture data. We employ imitation-based reinforcement learning (RL) for training, leveraging the diversity of large datasets to handle a wide range of target configurations. Training such a whole-body policy with imitation-based RL presents three major challenges. First, unlike full-body tracking, our targets are spatially sparse, guiding only three points. Second, whole-body skills require the upper and lower body to perform different tasks simultaneously, necessitating a large number of demonstrations to cover the joint action space. Third, obtaining accurate root position and velocity information in the real world is difficult, requiring a more robust policy that functions without precise root data. We address the first challenge by formulating a GAN-based RL framework that imitates the distribution of human demonstrations, rather than relying on specific full-body trajectories as policy input. To tackle the second challenge, we design two separate discriminators to reward the upper and lower body independently, promoting composability and coordination between them. Finally, to address the third challenge, we train a policy that does not depend on root position or velocity in world coordinates; instead, global information is inferred from the navigation goal and estimated via the onboard camera.

We evaluate each component of our system separately to better understand their contributions. For the low-level policy, we assess its ability to accurately track diverse target points across a wide range of motions. For the high-level navigation module, we find that using both human and robot data is essential for achieving reliable performance, while large-scale human data significantly improves generalization to novel environments. Finally, we integrate the full system and deploy it on a humanoid (Unitree G1) in the real world, demonstrating robust navigation and reaching performance in a human indoor space.

2 Related Works

Learning from Human Data. Following the growing popularity of humanoid robots, using human data to train humanoids has started to attract attention. Internet-scale human videos provide abundant training sources, making them suitable for pre-training implicit visual representation [3, 4, 5]. But their embodiment and observation gap make them less efficient and less relevant to learn a specific skill. Recent works show that high-quality task-relevant human data can benefit robot training, for

both tabletop manipulation [6, 7, 8], and indoor navigation [2]. Recent works propose creative ways to collect these data with various focuses. Portable devices, such as VR headsets [9], AR glasses [10], or SLAM cameras [7], can capture multiple modalities of human data, including head and hand poses, which can be easily transferred to the robot. While different tasks typically require different forms of human data for effective learning, we advocate for modular system interfaced with 3-point tracking for joint navigation, reaching, and locomotion.

Humanoid whole body control. Recent advances in humanoid hardware have made humanoids more accessible for academic research. To enable a humanoid to achieve meaningful tasks, the whole body controller (WBC) serves as a cornerstone in balancing the humanoid robot and coordinating whole-body movement. Traditionally, optimal control-based whole controller[11, 12, 13] has enabled a humanoid to walk, jump, and locomote through challenging terrain given a detailed kinematic trajectory to track. However, such high-quality kinematic trajectories are hard to obtain and highly specific to particular robot kinematics. Recently, reinforcement learning (RL) based whole body controller has shown impressive result to directly learn from human data[14, 15, 16, 17, 18]. Among them, most of WBCs are designed to track human joint positions[14, 15, 16, 17], which requires human whole body pose as input. To utilize sparse input that are easier to capture from virtual reality (VR) devices, [19, 20, 21] build their WBC to track human head and hand positions which can be accurately obtained from off-the-shelf VR headsets. Alternatively, [22] use VR headset and pedals to control upper-body and lower-body seperately. We use head and wrists tracking similar to [20, 19] as our WBC interface as it allows directly transfer human data in the task space. Compared to prior methods, our WBC also track head and wrists orientation which allow more versitile manipulation skills such as twisting the wrists.

Navigation. Extensive research in visual navigation has largely treated the robot as a point mass operating in a 2D plane. In long-term navigation, prior-work uses different exploration strategies ranging from local method [23, 24], global method [25, 26, 27] to end-to-end learning of goal-driven policies [28, 29, 30], planning over floor-plan waypoints or semantic landmarks to achieve robust performance across large spaces. Short-term navigation similarly relies on this 2D abstraction but focuses on socially compliant and reactive behaviors—dynamic obstacle avoidance and human–robot interaction [31, 32, 33, 33]. The closest work on humanoid platforms is NaVila [2], which applies long-horizon 2D waypoint navigation to a bipedal robot but decouples perception from locomotion and ignores full-body reaching. By comparison, our approach studies short-term 3D navigation directly through whole-body control, and to mitigate the embodiment gap between human and humanoid.

3 Method

Given a selected point on the initial RGB image observed by the robot, our system, HEAD, enables the humanoid to reach that point in the physical 3D world using its hand. HEAD is a modular system composed of a high-level policy for navigation and reaching, and a low-level policy for whole-body control (Figure 2). The core idea is that both navigation and reaching can be accomplished by commanding the same low-level whole-body policy to track the 6D poses of the head and hands.

3.1 Whole-body Controller

Given the target hand-eye positions and orientations provided by the high-level policy, the low-level, whole-body controller controls the humanoid through PD servos. To let the humanoid behave in a human-like manner while tracking arbitrary targets, we train the control policy through a GAN-like method [34] to perform motion imitation from unstructured motion data under the framework RL, combined with goal-directed control for target position and orientation tracking. Unlike two-stage distillation methods [35, 19], our method trains the low-level control policy in an end-to-end way for real-world deployment.

Curating Human Motion Dataset. We find that the quality of motion retargeting significantly impacts policy performance. We curated a 5-hour dataset by retargeting human motion capture data

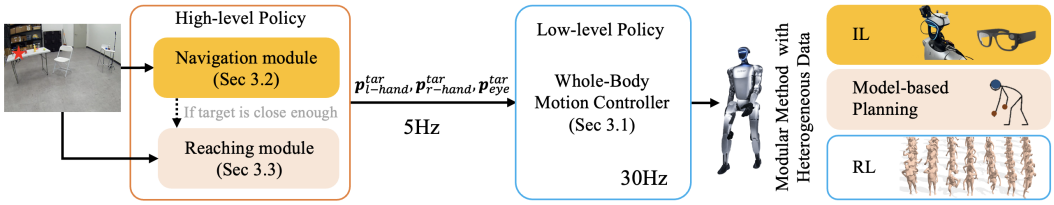


Figure 2: **System overview**: HEAD consists of a high-level policy with two modules, navigation and reaching, and a low-level policy that coordinates the whole-body motion. The high-level policy provides hand-eye tracking targets at a lower frequency while the whole-body controller tracks the hand-eye targets at a higher frequency. The learning-based navigation module learns from a mixed training dataset to map RGB ego-vision perception to camera target trajectories. The model-based reaching module generates hand-eye target poses. The low-level whole-body controller is trained using imitation-based RL on a set of human motion capture data.

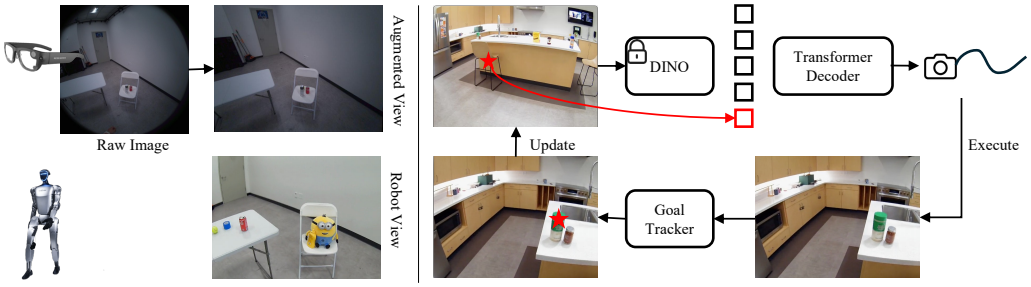


Figure 3: **Navigation Training Data** (left): we augment images (undistortion and homography transform) collected from Aria Glasses to make them resemble robot views. **Navigation Module Overview** (right): given an image and a goal as 2D point during inference, we extract DINO features, append the goal coordinate, and feed them to a transformer decoder to predict the future eye (camera) trajectory. The low-level whole-body controller executes the prediction and obtains a new observation. The goal is then tracked in the new image using an off-the-shelf point tracker.

to the G1 robot, from the AMASS [36] and OMOMO [37] datasets. The retargeting is achieved using keypoint matching similar to [17]. The collected motions ensemble representative behaviors across both manipulation and locomotion domains. Dataset will be open source upon acceptance.

Deployable Observation Space. To support real-world deployment, the observation space must be restricted to information accessible from the robot's onboard sensors. Our observation vector consists of robot link poses $\boldsymbol{p}_{\text{link}}$ in the robot's local coordinate frame in two consecutive time steps and joint velocity $\dot{\boldsymbol{q}}$ locally. It does *not* include any future information or rely on any privileged data in the world coordinate frame, such as root position and linear velocity, which are difficult to obtain outside of simulation. We found that removing the dependency on privileged information outperforms any alternative approaches that depend on reconstructed or predicted substitues.

Motion Imitation. We decouple the full-body motion into upper and lower body groups, and employ two discriminators simultaneously to perform imitation learning. By doing so, the policy can learn the combination of the poses from upper and lower body parts, instead of being limited by fixed full-body poses provided in the motion dataset. The GAN-like approach of RL allows the policy to imitate motions from arbitrary segments in the motion dataset without needing to generate or obtain a full trajectory of imitation beforehand, while fulfilling the tracking task.

Sparse Target Tracking. To avoid introducing target information defined in the global space, we represent the tracking target, as the input to the policy network, through relative transformations: $\boldsymbol{g} = [\boldsymbol{p}^{\text{tar}}_{\text{eye}} \ominus \boldsymbol{p}_{\text{eye}}; \boldsymbol{p}^{\text{tar}}_{\text{l-hand}} \ominus \boldsymbol{p}_{\text{l-hand}}; \boldsymbol{p}^{\text{tar}}_{\text{r-hand}} \ominus \boldsymbol{p}_{\text{r-hand}}]$ where $\ominus$ denotes the relative transformation operator, and "tar" refers to the target pose. To perform tracking, during training, we define the goal-directed reward based on $\boldsymbol{g}$ after the action is executed at each timestep.

Sim-to-real Considerations. Along with the task reward of target tracking, we additionally define a regularization term to aid sim-to-real transfer. To further improve robustness, we apply extensive domain randomization over dynamics parameters and sensor noise during training.

We employ the multi-objective learning framework from [38] to perform policy training, while optimizing the two imitation objectives using rewards provided by the discriminators and the goal-directed objective through the manually defined reward function at the same time. We refer to the supplementary materials for the implementation details.

3.2 Navigation Module

Given a low-level whole-body controller capable of tracking three points, our navigation module guides the robot to a designated goal specified as a 2D point in the initial RGB image observed by the robot. During inference, the navigation model takes the current RGB image from the navigation camera along with the tracked 2D goal—provided by a point tracker [39]—and predicts the future eye trajectory in both position and orientation (Fig. 3 right). Specifically, we extract DINO features of the input image I_t and add positional embedding to the goal g_t. We pass them to a transformer decoder to output a future camera trajectory $C_{t:t+T}$ as transformations relative to the previous frame.

Collecting Human Data. We propose an automatic method to use Aria Glasses for collecting goal-conditioned human training data as tuples of images, future camera trajectories, and 2D goals $(I_t, C_{t:t+T}, g_t)$. The glasses provide accurate camera poses, static point clouds and gaze estimation for all captured data. We approximate the current goal by finding the closest point in the static point cloud along the future gaze vector and projecting it onto the image plane via current camera pose.

Domain Shift. However, a navigation model trained on a limited set of human data struggles with two potential domain shifts that need to be addressed. First, to improve generalization to unseen scenes, we incorporate the large-scale egocentric dataset Aria Digital Twin (ADT) [40], which contains 400-minute of various indoor activities such as cleaning and cooking. Thanks to our automatic data curation pipeline, we can easily convert any Aria Glasses data into goal-conditioned navigation training data. Second, due to the embodiement gap between the robot and an average human adult, there is a significant disparity in visual perception. To align the Aria Glasses' wide fisheye view with the robot's narrower camera, we apply undistortion and homography transforms to produce virtual views of robot from human data (Fig. 3 left). (See appendix for details.) Beyond visual discrepancies, humans and robots also operate at different speeds. Empirically, we find that the robot moves approximately 7× slower than humans, so we subsample robot videos accordingly during training.

We also collect a small amount of robot data by commanding it to navigate while recording its head poses with the mocap system. We co-train the navigation module with both human and robot data.

3.3 Reaching Module

While the navigation module drives the robot toward the target object at room scale, the reaching module handles the final approach to touch the object. We use a second downward-facing RGB-D camera with a narrower, zoomed-in FoV for reaching. The high-level policy switches from navigation to reaching while the low-level policy continues running to ensure a smooth transition.

Navigation-Reaching Transition. The navigation policy hands control to the reaching module when the target object enters the downward-looking RGB-D camera's view and is within reaching range. The goal is transferred to the RGB-D frame via correspondence matching [41].

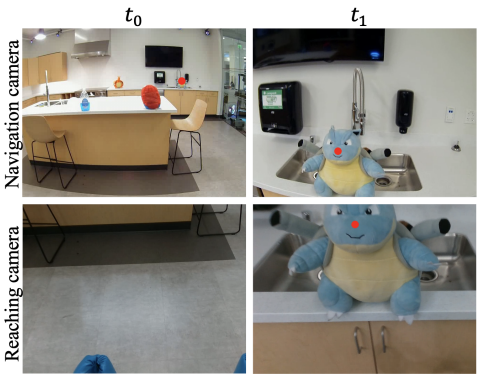


Figure 4: Robot is away from the goal at t_0, goal is only visible in navigation camera. At transition time t_1, the goal is visible by the reach camera and close enough to the robot.

(a). Unitree G1 (b). Sensors used by robot (c). Lab room and kitchen, each with 3 different arrangements.

Figure 5: Hardware setup and test environments.

Reaching the Target. The reaching module approximates the goal as a 3D hand position and computes the target hand orientation and head 6D pose for the low-level policy. Since the tracked goal only specifies the hand position, we solve Inverse Kinematics (IK) using Mink [42] to infer missing head poses and hand orientations. To ensure a smooth high-level transition and keep the robot posture looking natural, we initialize the IK optimization from the current robot state and add an objective term to encourage small change in the center of mass position and pelvis orientation.

4 Experiment

We first specify our hardware setup (4.1). We evaluate our overall hand-eye delivery system with novel objects in different environments(Sec. 4.2). Then, we analyze individual modules – the design choice of the whole-body controller (Sec. 4.3), and contribution of each data ingredient to train the navigation module (Sec. 4.4).

4.1 Hardware Setup

We build our proposed system around a Unitree G1 humanoid robot as shown in Fig.5. For navigation, we use a wide-angle USB webcam with 90° HFOV and 67.5° VFOV; for the reaching module, we use G1 built-in realsense D435 RGB-D camera. During deployment, all modules are running on a PC equipped with an RTX4090 GPU and an i9-14900K CPU, and the robot is controlled via an Ethernet connection.

We conduct our experiment in two rooms: a lab (training room) and a kitchen (deploy room). Robot-specific training data is only collected in the lab room. This is to emulate a deployment scenario where no hardware is available to record robot ground truth training data. In each room, we arrange several pieces of furniture (e.g., shelves, chairs, tables, stools) to create diverse layouts. All test layouts and objects are unseen.

4.2 Whole Body Reaching with Different Scenes

We evaluate our method across three different layouts in each room. For each layout, robot will be ask to reach 4 objects placed at different locations and of different heights, as shown in Fig. 5.c. Detailed experiment result is shown in Tab 1. Overall, our method achieves 71 % success rate across different environments. Success rate in lab room is 25% greater than in the kitchen, which has narrower corridors and more reflective surfaces, which pose challenges to the goal tracker, navigation transformer and transition module. For failure cases, motion blur from robot movement may cause the tracker to lose track of the object or track the wrong object, which may confuse the navigation

module. Also, humans tend to move faster and take a more aggressive path when collecting data, which may not be feasible for robots and result in a collision.

Room	Scene	Success rate	Number of misses	Number of collision
Lab room	Scene 1	3/4	1/4	0/4
	Scene 2	3/4	1/4	0/4
	Scene 3	4/4	0/4	0/4
Kitchen	Scene 1	2/4	1/4	1/4
	Scene 2	2/4	1/4	1/4
	Scene 3	3/4	0/4	1/4

Table 1: Number of successes and different kinds of failures across different evaluations

4.3 Performance of Whole Body controller

Setup. We train our whole-body controller in Isaac Gym and report its performance in two simulations: Isaac Gym [43] and MuJoCo [44]. As the policy was trained in Isaac Gym, the performance reported on the left side of Table 2 highlights the impacts of the design choice in the training procedure. Furthermore, on the right side of Table 2, the simulation-to-simulation evaluation in MuJoCo further quantifies the robustness of each policy when the simulated contact model is closer to the real world [45].

Single-stage RL Training Recipe. We observed that guiding reinforcement learning (RL) exploration with generative adversarial networks (GANs) greatly improves sample efficiency. Compared to a single discriminator that jointly criticizes whole-body motion, we find that disentangling upper-body and lower-body motion via separate reward from two discriminators helps. It is probably because separate discriminators prevent the policy from entangling irrelevant motions during training and achieve a lower tracking error. For instance, in most walking clips, people naturally swing their arms, while dual-arm manipulation typically occurs from a squatting posture. A policy trained with a *single* full-body discriminator tends to memorize the arm-swing pattern and then fails to handle a carried box while walking. In contrast, our setup decouples arm manipulation from balance control, enabling the whole-body controller to produce more diverse motions under 3-point tracking. In unseen tasks, our policy consistently outperforms the single whole-body discriminator variant.

4.4 Performance of Navigation Module

Setup. We use Aria Glasses to collect 200 human clips (H-Lab/H-Kit) per room. For robot data (R), we use mocap system to collect 38 clips of robot trajecotries in the lab and 20 in the kitchen. Each clip lasts around 4 seconds (human) or 30 seconds (robot). Of the lab robot data, 24 clips are used for training and 14 for testing; all kitchen robot clips are used for testing. This is to mimick a deployment scenario where no device is available to record robot ground truth training data. Human goals are

Design Choice	Isaac Gym - Unseen Motions			MuJoCo - Sim2Sim Transfer		
	Pos Error [m] $\downarrow$	Quat Error [rad] $\downarrow$	Failure [%] $\downarrow$	Pos Error [m] $\downarrow$	Quat Error [rad] $\downarrow$	Failure [%] $\downarrow$
Ours	**0.075**	**0.120**	**0**	**0.153**	**0.326**	**3**
Single discriminator	0.149	0.169	0	0.525	1.015	13
No discriminator	0.540	1.138	97	1.127	2.044	99

Table 2: Tracking accuracy is reported as positional error (m) and orientation error (rad). A timestep is counted as a failure when the head height deviates from the target by ≥ 0.4 m. Each configuration was trained for 50 k epochs, evaluated on 1-minute unseen motion clips, and repeated five times.

approximated via eye gaze (Sec.3.2), while robot goals are manually annotated and tracked. We also include the out-of-distribution ADT dataset (O)[40], with 400 minutes of Aria Glasses footage of users doing tasks like cleaning and cooking.

Arch	Lab Room			Kitchen (Deploy Room)		
	Training Data	SR	Error	Training Data	SR	Error
shared	H-Lab	0.14	0.704	–	–	–
shared	R-Lab	0.71	0.427	zero-shot	0.35	0.827
shared	R-Lab + O	0.79	0.399	zero-shot	0.35	0.726
shared	R-Lab + H-Lab	0.79	0.374	+ H-Kit	0.45	0.664
2-branch	R-Lab + H-Lab + O	0.79	**0.356**	+ H-Kit	0.20	0.812
shared	R-Lab + H-Lab + O (Ours)	**0.86**	0.380	+ H-Kit	**0.60**	**0.608**

Table 3: **Navigation Evaluation:** we report success rate (SR) and mean position error (Error) in open-loop prediction.

Metrics: We report open-loop prediction performance in both the lab room and the kitchen. For each test video, we predict a 10-step trajectory at every time step and compute the mean error against the ground-truth trajectory after rolling out these prediction. A prediction is considered successful if the final error is within 0.6 meters, which corresponds roughly to the humanoid's effective manipulation range.

Recipe of Combining Human and Robot Data for Navigation. As reported in Table 3 Lab Room column, training with in-domain human data (H-Lab) only leads to poor success rates comparing to training with in-domain robot data (R-Lab), likely because the embodiment differences between humans and robots are not learned. The model that trains with in-domain robot data (R-Lab) together with human data, either from out-of-distribution (O) or from in-domain human data (H-Lab), further increases the performance. Human data in the same environment (H-Lab) helps moderately more than out-of-distribution data (O). By combining all existing data, we achieve the best performance.

Deploying into New Scenes. As reported in Table 3 Kitchen column, while training with only robot data shows reasonable performance in the lab room, it fails to generalize to the new (deploy) room. Collecting additional in-domain human data (+H-Kit) alone helps the robot to better in the new scenes. Note that although incorporating ADT (O) does not significantly improve performance within the lab room, it substantially boosts success rates in the deploy room, where no robot training data is available. This highlights the importance of leveraging diverse, unlabeled human data to enhance cross-scene generalization.

Shared Decoder Branch Improves Navigation Generalization. In contrast to common practice in manipulation tasks[6, 35], we find that sharing a single decoding branch for human and robot data improves scene generalization in navigation tasks. We hypothesize that this is because the embodiment differences between humans and robots in short-term navigation are smaller than those in manipulation, making a shared representation more effective.

5 Conclusion

We presented HEAD, an autonomous hand-eye delivery system for humanoid navigation and reaching. Our method achieves a 71% success rate in reaching different objects placed in two different environments with obstacles. A future extension could be building a general grasp framework that can grasp different objects placed at various locations. Learning more fine-grained whole body navigation that can be aware and avoid collision with varying parts of the body will also be an interesting direction for a humanoid robot to be useful in real-life environments.

6 Limitations

Although our method can work in environments with obstacles, it only collects human head poses and uses it as a robot control interface without considering other parts of the body. For reaching the heavily occluded target, human will utilize their whole body coordination to avoid collision, such as going sideways when facing a narrow gap or stepping over lower obstacles. Using our method will cause the robot to take a more conservative approach when facing a complex environment without fully utilizing its agility. Moreover, using head and wrist poses as a humanoid control interface has its intrinsic ambiguity for controlling lower body and may cause the humanoid robot to hesitate. For example, when the three-point is moving forward, it is hard for a humanoid to know whether the high-level intention is to bend forward or walk forward. More information that could be estimated from egocentric devices such as foot pose may help to reduce hesitation when taking actions.

Acknowledgments

We thank NSF #2153854, Stanford Institute for Human-Centered AI. Stanford Robotics Center for facilities for supporting our research, we also thank Yanjie Ze, Haochen Shi, Weizhuo Wang for providing useful suggestions.

References

[1] S. Levine and D. Shah. Learning robotic navigation from experience: principles, methods and recent results. *Philosophical Transactions of the Royal Society B*, 2023.

[2] A.-C. Cheng, Y. Ji, Z. Yang, Z. Gongye, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang. Navila: Legged robot vision-language-action model for navigation. *RSS*, 2025.

[3] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022.

[4] S. Dasari, M. K. Srirama, U. Jain, and A. Gupta. An unbiased look at datasets for visuo-motor pre-training. In *CoRL*, 2023.

[5] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *ICRA*, 2024.

[6] S. Kareer, D. Patel, R. Punamiya, P. Mathur, S. Cheng, C. Wang, J. Hoffman, and D. Xu. Egomimic: Scaling imitation learning via egocentric video. *arXiv preprint arXiv:2410.24221*, 2024.

[7] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *RSS*, 2024.

[8] 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.

[9] S. Chen, C. Wang, K. Nguyen, L. Fei-Fei, and C. K. Liu. Arcap: Collecting high-quality human demonstrations for robot learning with augmented reality feedback. *ICRA*, 2025.

[10] J. Engel, K. Somasundaram, M. Goesele, A. Sun, A. Gamino, A. Turner, A. Talattof, A. Yuan, B. Souti, B. Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023.

[11] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard. Crocoddyl: An efficient and versatile framework for multi-contact optimal control. In *ICRA*, 2020.

[12] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Autonomous robots*, 2016.

[13] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *IJRR*, 2011.

[14] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *TOG*, 2018.

[15] Z. Luo, J. Cao, K. Kitani, W. Xu, et al. Perpetual humanoid control for real-time simulated avatars. In *ICCV*, 2023.

[16] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[17] T. He, Z. Luo, W. Xiao, C. Zhang, K. Kitani, C. Liu, and G. Shi. Learning human-to-humanoid real-time whole-body teleoperation. In *IROS*, 2024.

[18] M. Ji, X. Peng, F. Liu, J. Li, G. Yang, X. Cheng, and X. Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

[19] T. He, Z. Luo, X. He, W. Xiao, C. Zhang, W. Zhang, K. Kitani, C. Liu, and G. Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[20] T. He, W. Xiao, T. Lin, Z. Luo, Z. Xu, Z. Jiang, J. Kautz, C. Liu, G. Shi, X. Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024.

[21] A. Winkler, J. Won, and Y. Ye. Questsim: Human motion tracking from sparse sensors with simulated avatars. In *SIGGRAPH Asia*, 2022.

[22] C. Lu, X. Cheng, J. Li, S. Yang, M. Ji, C. Yuan, G. Yang, S. Yi, and X. Wang. Mobile-television: Predictive motion priors for humanoid whole-body control. *arXiv preprint arXiv:2412.07773*, 2024.

[23] B. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Robotics and autonomous systems*, 1991.

[24] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *IROS*, 2002.

[25] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, et al. Minerva: A second-generation museum tour-guide robot. In *ICRA*, 1999.

[26] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'*, 1997.

[27] D. Holz, N. Basilico, F. Amigoni, S. Behnke, et al. A comparative evaluation of exploration strategies and heuristics to improve them. In *ECMR*, 2011.

[28] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov. Gated-attention architectures for task-oriented language grounding. In *AAAI*, 2018.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 2015.

[30] A. Sridhar, D. Shah, C. Glossop, and S. Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *ICRA*, 2024.

[31] N. Hirose, D. Shah, A. Sridhar, and S. Levine. Sacson: Scalable autonomous control for social navigation. *RAL*, 2023.

[32] J. Mumm and B. Mutlu. Human-robot proxemics: physical and psychological distancing in human-robot interaction. In *Proceedings of the 6th international conference on Human-robot interaction*, 2011.

[33] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh. Core challenges of social robot navigation: A survey. *ACM Transactions on Human-Robot Interaction*, 2023.

[34] P. Xu and I. Karamouzas. A gan-like approach for physics-based imitation learning and interactive character control. *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, 2021.

[35] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, L. Paulsen, G. Yang, S. Yi, et al. Humanoid policy˜ human policy. *arXiv preprint arXiv:2503.13441*, 2025.

[36] N. Mahmood, N. Ghorbani, N. F. Troje, G. Pons-Moll, and M. J. Black. Amass: Archive of motion capture as surface shapes. In *ICCV*, 2019.

[37] J. Li, J. Wu, and C. K. Liu. Object motion guided human motion synthesis. *TOG*, 2023.

[38] P. Xu, X. Shang, V. Zordan, and I. Karamouzas. Composite motion learning with task control. *TOG*, 2023.

[39] N. Karaev, I. Makarov, J. Wang, N. Neverova, A. Vedaldi, and C. Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. *arXiv preprint arXiv:2410.11831*, 2024.

[40] X. Pan, N. Charron, Y. Yang, S. Peters, T. Whelan, C. Kong, O. Parkhi, R. Newcombe, and Y. C. Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *ICCV*, 2023.

[41] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021.

[42] K. Zakka. Mink: Python inverse kinematics based on MuJoCo, July 2024. URL https://github.com/kevinzakka/mink.

[43] 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.

[44] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *IROS*, 2012.

[45] J. Z. Zhang, T. A. Howell, Z. Yi, C. Pan, G. Shi, G. Qu, T. Erez, Y. Tassa, and Z. Manchester. Whole-body model-predictive control of legged robots with mujoco, 2025. URL https://arxiv.org/abs/2503.04613.

[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[47] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[48] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[49] J. H. Lim and J. C. Ye. Geometric GAN. *arXiv preprint arXiv:1705.02894*, 2017.

[50] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved training of wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017.

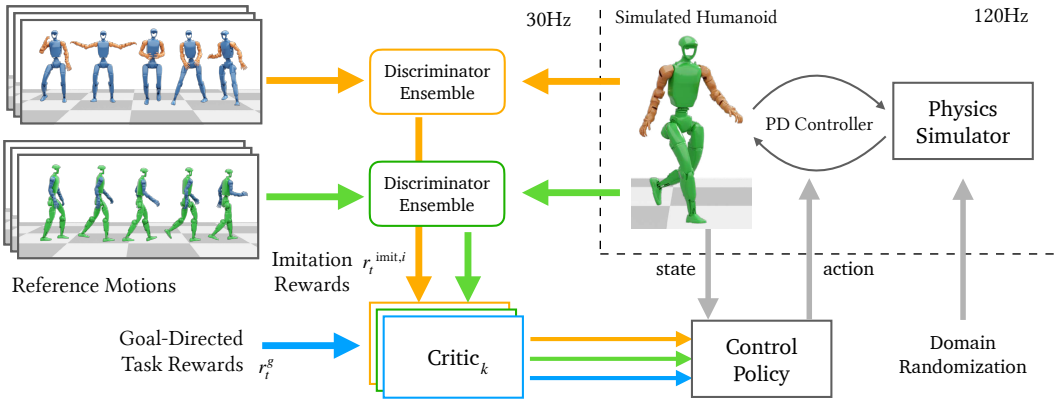


Figure S1: Systemic overview of the training scheme of our whole-body controller. We employ a multi-objective learning framework plus a GAN-like architecture for imitation learning. Our scheme allows imitating arm poses (orange) and those of the rest body parts (green) from different sources of reference motions simultaneously, and uses a goal-directed reward to fulfill the sparse tracking task for the head and hand poses.

Table S1: Hyperparameters

Parameter	**Value**
policy network learning rate	5×10^{-6}
critic network learning rate	1×10^{-4}
discriminator learning rate	1×10^{-5}
reward discount factor (γ)	0.95
GAE discount factor (λ)	0.95
surrogate clip range (ϵ)	0.2
gradient penalty coefficient (λ^{GP})	10
number of PPO workers (simulation instances)	1024
PPO replay buffer size	1024×8
PPO batch size	256
PPO optimization epochs	5
discriminator replay buffer size	$1024 \times 8 \times 2$
discriminator batch size	512

A Whole Body Controller

Figure S1 shows the overview of the systemic architecture of the whole body controller. We use IsaacGym [43] as the physics engine for policy training. The policy runs at 30Hz and controls the humanoid through a PD servo.

Instead of directly imitating full-body motions, based on the motion decoupling scheme from previous literature [38], we split the full-body motions into arm and torso groups, and employ two discriminators at the same time to evaluate the imitation performance for partial motions. Besides the current state of the humanoid, the control policy takes only the tracking target for the head and two hands as the goal input. Without needing the whole trajectory of full-body tracking, the discriminators evaluate the imitation performance of partial motions and allow the arm motions to be combined with the torso and lower-body motions from different motion clips in a free way.

Given an additional goal-directed reward for target tracking plus regularizations for sim-to-real consideration, we leverage the multi-objective learning framework [38] to balance the learning of multiple imitation and goal-directed objectives. The final optimization objective for policy training can be written as

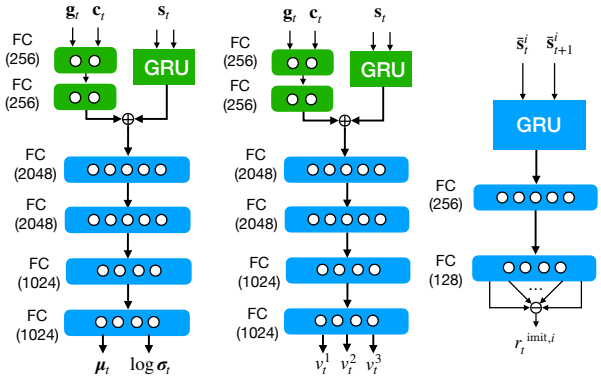
$$\max \mathbb{E}_t \left[\sum\nolimits_{\kappa} w_k \bar{A}_{t,k} \log \pi(\mathbf{a}_t | \mathbf{o}_t) \right] \tag{S1}$$

where $\bar{A}_{t,k}$ is the standardized advantage that is estimated according to the achieved reward of each objective k, w_k is an associated weight, and $\mathbf{o}_t$ is the observation including the humanoid's state $\mathbf{s}_t$ and the goal state $\mathbf{g}_t$. We choose $w_{\text{imit},i} = 0.2$ for each of the two imitation objectives and $w_g = 0.6$ for the goal-directed objective.

We use PPO [46] as the backbone reinforcement learning algorithm and take the Adam optimizer [47] to perform network optimization for policy training. The hyperparameters used for policy training are listed in Table S1 and the network structures are shown in Figure S2. We manually pick 1363 clips of locomotion and loc-manipulation motions from AMASS [36] and OMOMO [37]. The whole data set of motions is around 5 hours long.

A.1 Observation Space

The G1 humanoid has 33 links and 27 controllable joints, where the wrist and neck joints are fixed and the waist part only has 1 degree of freedom around the yaw axis. We take two historical frames as the input and process the state vector via a GRU [48]. This leads to a state space of $\mathbf{s}_t \in \mathbb{R}^{33\times7\times2}$ including the position and orientation (in quaternion) of each link in the local frame of the humanoid's root link, and an action space of $\mathbf{a}_t \in \mathbb{R}^{27}$.



(a) Policy Network (b) Value Network (c) Discriminator

Figure S2: Network structures. We use $\oplus$ denoting the add operator and $\ominus$ denoting the average operator. For multi-objective learning, we employ a value network with 3-dimensional output for the two imitation objectives and one goal-directed objective.

For control purposes, we take the root angular velocity and joint velocity locally as the control state $\mathbf{c}_t \in \mathbb{R}^{30}$. We ignore the linear velocity of the root link, since the linear velocity is hard to access from the humanoid when deployed in the real world.

The goal state vector $\mathbf{g} \in \mathbb{R}^{3\times3\times7}$ includes the target positions and orientations (in quaternion) of the three links (left hand, right hand and head) in the next three frames.

The final observation space $\mathbf{o}_t$ is composed of the three components $\mathbf{s}_t$, $\mathbf{c}_t$, and $\mathbf{g}_t$.

A.2 Reward Terms

Instead of using a single discriminator for each motion group, we take the GAN-like architecture from ICCGAN [34], and employ an ensemble of 32 discriminators for each motion group. The imitation-related reward of the discriminator ensemble D_i is computed via

$$r_t^{\text{imit},i}(\bar{\mathbf{s}}_t^i, \bar{\mathbf{s}}_{t+1}^i) = \frac{1}{N}\sum_{n=1}^{N} \textsc{Clip}\left(D_n^i(\bar{\mathbf{s}}_t^i, \bar{\mathbf{s}}_{t+1}^i), -1, 1\right), \tag{S2}$$

where the subscript i indicates different imitation objectives (upper or lower body groups), $\bar{\mathbf{s}}_t^i$ is the partially observable character state for the imitation objective i, and the discriminator ensembles N discriminators each which is trained using hinge loss [49] with gradient penalty [50]. We choose $N = 32$ in our implementation.

The goal-directed reward mainly measures tracking errors and also consists of three terms to stabilize the motions for sim-to-real consideration:

$$r_t^g = r_{\text{tracking}} + 0.8 r_{\text{in-air}} + 0.5 r_{\text{sliding}} + 0.005 r_{\text{energy}}. \tag{S3}$$

For simplicity, here we omit the subscript t for each of the reward terms.

Table S2: Domain Randomization Settings

Parameters	**Value**
Base Mass (kg)	$[-3, 3]$
Body Link Friction Coefficient	$[0.5, 1.25]$
Scale on PD Servo Gain (Kp)	$[0.7, 1.3]$
Scale on PD Servo Damping (Kd)	$[0.7, 1.3]$
Action Delay Ratio (ρ_{delay})	0.5

r_{tracking} is the reward measuring the tracking error:

$$r_{\text{tracking}} = 0.5 \exp\left(-\frac{5}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}|\varepsilon_{\text{pos},i}|\right) + 0.5\exp\left(-\frac{2}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}|\varepsilon_{\text{orient},i}|\right) \tag{S4}$$

where $\mathcal{I} = \{\text{left hand}, \text{right hand}, \text{head}\}$ is the set of links under tracking, $\varepsilon_{\text{pos},i}$ is the position error between the current position of the link i and its target position measured in Euclidean distance, and $\varepsilon_{\text{orient},i}$ is the orientation error measured by the angle between the orientation of the link i and the target.

$r_{\text{in-air}} = \min\{0, t_{\text{in-air}} - 0.5\}$ encourages the policy to keep the foot in the air for at least 0.5s during stepping. It is computed for the swing foot when it contacts the ground, and $t_{\text{in-air}}$ is the hanging time of that foot before the contact.

$r_{\text{sliding}} = -\sum_f c_f ||\mathbf{v}_f||_2$ penalizes the linear velocity of the foot link f if it contacts the ground, where $c_f = 1$ or 0 indicating the contact state of the foot f.

$r_{\text{energy}} = -\sum_j \left(0.1|\tau_j v_j| + 0.005\tau_j^2\right)$ penalizes the energy cost for each joint j, where τ_j is the torque applied on joint j and v_j is the joint's rotation velocity.

A.3 Domain Randomization

Parameters for domain randomization are listed in Table S2.

When testing the sim2real transfer, we found adding simulated delay during training time essential to prevent the robot from jittering. Compared to using a random delay across all environments, which is challenging to train, we found using a constant 1-step delay on a fixed portion ρ_{delay} of environments improves training speed, and can prevent the robot from jittering caused by the uncertainty delay during policy execution in the real world. We choose $\rho_{\text{delay}} = 0.5$, which means that the action delay is applied on half of the training environments.

B Image Augmentation of Navigation Data

We augment human data collected by Aria Glasses to resemble the robot's view. First, we address image discrepancies caused by differences in capture devices. Aria Glasses use an RGB fisheye camera with a $110°$ horizontal and vertical field-of-view (HFOV, VFOV), while the robot camera has a $90°$ HFOV and $67.5°$ VFOV. We undistort the fisheye images to obtain I_u, which approximates a pinhole camera view with intrinsics K_u. Second, we address discrepancies in camera pose (extrinsics) due to morphological differences. Humans tend to tilt their heads—e.g., looking downward when reaching for an object—whereas the robot's navigation camera, fixed at the top of the head, always looks forward. As a result, the same object may appear in different regions of the image despite similar camera positions. To align viewing angles, we apply a homography to the undistorted image I_u to match the pitch angle. Given the robot camera's intrinsics K_r and Aria's effective intrinsics K_u, we apply $H = K_r R_{\text{pitch}} R_{[\theta]}^T K_u^{-1}$, where $R_{[\theta]}$ is the pitch component of the current human camera pose and R_{pitch} is the desired robot pitch.

C Usage of motion capture system

Although our whole-body controller and navigation model don't rely on world coordinates, in our real-world experiment, we found it hard to keep the robot standing at the exact location without drifting. Therefore, we attach mocap markers only to the robot's head and transform the camera trajectory into the world frame; we found that having closed-loop tracking in the world frame can significantly improve the accuracy of robot reaching.