EgoGuide: Egocentric Guidance for Efficient Robot-Free Demonstration Collection and Learning

Yue Xu[1], Mingtao Nie[1], Tianle Li[1], Hong Li[1], Yibo Luo[1], Siyuan Huang[3], Yong-Lu Li[1,2]*
[1]Shanghai Jiao Tong University, [2]Shanghai Innovation Institute
[3]Beijing Institute for General Artificial Intelligence (BIGAI)
{silicxuyue, yonglu_li}@sjtu.edu.cn

Abstract: Robot learning from real-world demonstrations is currently constrained by data scaling. Universal Manipulation Interface (UMI) provides an efficient robot-free data collection interface, yet current UMI-style pipelines often collect redundant demonstrations and lack global scene context. To improve data efficiency, we present EgoGuide, a collection interface that records synchronized wrist and head/egocentric observations and couples them with online visual-geometric data quality guidance. We also introduce a Gated Egocentric Residual Policy for robust learning from a viewpoint-varying egocentric camera, allowing head/egocentric context to correct ambiguous local observations while preserving stable wrist-view control. Real-world experiments show that EgoGuide reduces the required number of data episodes and improves data efficiency. The residual policy further improves robustness under visual occlusion.
Project Page: https://silicx.github.io/EgoGuide/
Keywords: Hardware design and optimization, Universal Manipulation Interface

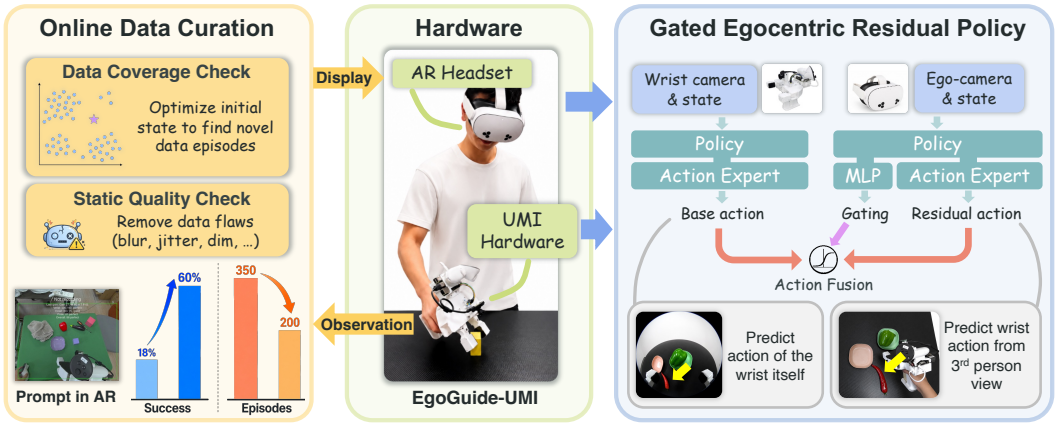


Figure 1: Overview of EgoGuide. We extend UMI-style data collection with synchronized wrist and head/egocentric sensing and online visual-geometric data quality guidance through AR feedback. A gated residual policy uses the head/egocentric view to complement a stable wrist-view policy.

1 Introduction

Currently, the scaling of robot learning remains fundamentally constrained by real-world demonstration data. Universal Manipulation Interface (UMI) [1] offers an appealing alternative to costly teleoperation systems and shows a promising direction for scalable manipulation data collection. However, scaling robot learning with robot-free interfaces such as UMI is not simply a matter of

*Corresponding author.

recording more trajectories. We define *data efficiency* as the human collection effort required to train a policy to a target success rate. From this perspective, UMI can be less data-efficient than teleoperation systems, sometimes requiring more than $5\times$ episodes for comparable success.

We identify two critical bottlenecks. First, UMI data are collected with a gripper surrogate and transferred to a robot with different kinematics, coordinates, and execution constraints, so policies require broad coverage of data variation rather than redundant successful trajectories; yet demonstrators receive little feedback about which states are already covered or underrepresented. Second, because most UMI systems rely on a single wrist camera, observations can be too local to capture the full task state under occlusion, object disappearance, or long-horizon progress. These bottlenecks suggest that improving UMI data efficiency requires both collection-time coverage guidance and the exploitation of complementary egocentric observations.

To address these issues, we first present *EgoGuide*, an integrated robot-free data collection system that combines multi-view demonstration collection and online data quality guidance. EgoGuide targets the data diversity and coverage bottleneck by making the demonstrator aware of the data quality of the current visual-geometric state. We augment the handheld UMI interface with AR-based egocentric sensing and spatial tracking. Multiple modalities stream synchronously to a workstation, which computes an online data coverage score from wrist and head/egocentric visual-geometric information. The score is rendered in the AR interface to encourage adjustments to the initial hand pose, object arrangement, viewpoint, or workspace configuration before recording an episode. EgoGuide also allows and encourages recording from the middle of a task, extending data quality control beyond the early stages.

For robust egocentric policy learning, we introduce the *Gated Egocentric Residual Policy* (GERP), which addresses the observability bottleneck by using head/egocentric context to correct wrist-view control *on demand*. We argue that human head movement is often unintentional, so rather than building an active egocentric camera with imitation learning [2, 3, 4], we propose to learn a robust policy that can generalize to a fixed egocentric camera, aligning with realistic robot deployment. GERP keeps a wrist-view policy as a stable base, conditions an egocentric branch on the head/egocentric image and the wrist pose relative to the head frame, and gates the egocentric action candidate with the base action. This preserves wrist-view control in ordinary states while allowing head/egocentric context to modify the action when local observations are ambiguous or incomplete.

We evaluate on real-world manipulation tasks covering standard UMI-style settings as well as failure cases involving occlusion and limited visual coverage. On regular long-horizon tasks, our online collection guidance substantially improves success with the same dataset size and, on Pepper Sorting, reaches comparable success using only **50%** as many demonstrations. The egocentric residual policy further improves performance under occlusion. These results support the central hypothesis of this paper: reliable scaling of UMI requires both collecting more informative demonstrations and learning to use heterogeneous observations.

2 Related Work

Robot-Free Demonstration Collection. Large-scale robot data collection remains a bottleneck for robot learning. Teleoperation systems such as ALOHA [5], Mobile ALOHA [6], and GELLO [7] improve the accessibility of robot-native data collection. Recently, Universal Manipulation Interface (UMI) [1] introduced a portable gripper-like device that allows low-cost in-the-wild data collection. Subsequent works extend this paradigm. FastUMI improves portability and collection throughput [8]. exUMI [9], TacUMI [10], ViTaMIn [11], and FreeTacMan [12] incorporate tactile or force-related sensing for contact-rich manipulation. MV-UMI [13] and UMI-3D [14] enhance visual or spatial observability. DexUMI [15], HoMMI [16], and HuMI [17] extend toward dexterous or whole-body manipulation. Building on this, we study how to improve demonstration quality through online guidance.

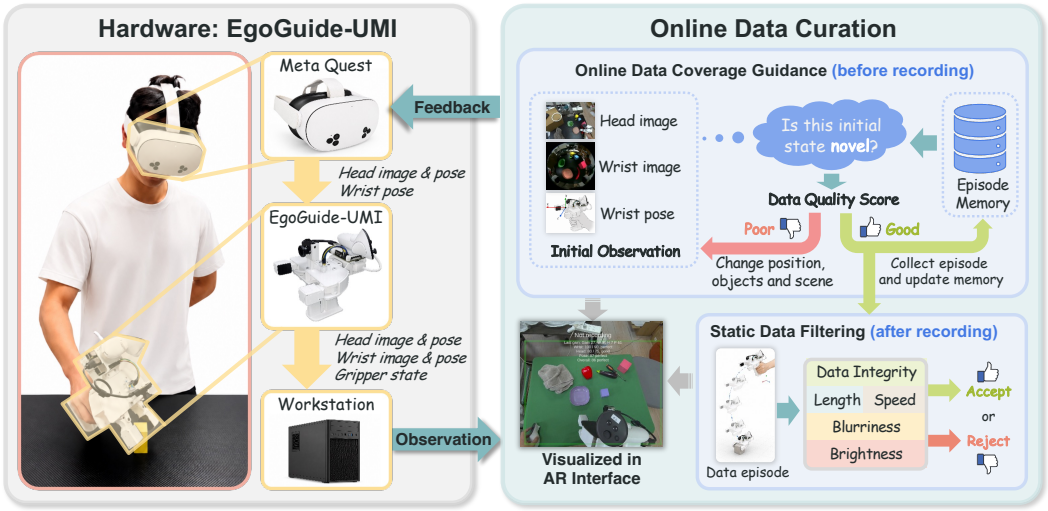


Figure 2: Overview of the EgoGuide hardware system. EgoGuide-UMI records wrist and head/egocentric views, gripper proprioception, and spatial poses, while the online curation module computes coverage scores and sends AR guidance back to the demonstrator.

Egocentric vision is also exploited in UMI systems. Vision in Action (ViA) [2] first proposes to record UMI data with an additional egocentric camera and learns an active robot head policy from human demonstrations. ActiveUMI [4] also introduces head-mounted observations with active head deployment. EgoMI [18] studies active egocentric perception in whole-body manipulation. Unlike methods that directly model or execute active head motion, we use egocentric observations as auxiliary context in a fixed egocentric camera configuration.

Robot Data Quality Evaluation. Imitation learning is sensitive not only to dataset scale, but also quality. Interactive imitation-learning methods such as DAgger [19], SafeDAgger [20], and HG-DAgger [21] collect additional supervision in states visited by the learned policy. RoboPocket [22] brings the DAgger strategy to the UMI system for online data curation. Offline studies such as RoboMimic [23], RoboTurk [24], and RoboNet [25] further show that policy performance depends on demonstration quality and diversity. DemInf [26] estimates trajectory utility from state diversity and action predictability. Demo-SCORE [27] uses online robot experience to identify useful demonstrations. CUPID [28] estimates the contribution of individual demonstrations using influence functions. These methods filter or select demonstrations after data collection, while we enable guidance **during** robot-free data collection.

3 EgoGuide System

EgoGuide is designed as a closed-loop robot-free collection system, including both UMI hardware design and an online data curation method. Fig. 2 summarizes the hardware and guidance pipeline.

3.1 EgoGuide-UMI Hardware

We first propose our hardware design **EgoGuide-UMI**, based on the open-source exUMI [9].

Handheld gripper interface. EgoGuide-UMI follows the UMI-style principle of collecting robot-free demonstrations with an easy-to-fabricate hand-held gripper. A rotary sensor mounted on the gripper joint measures the opening width g and sends it to an on-device Raspberry Pi controller. We mount an industrial board-level fisheye camera near the gripper to obtain the wrist image I^W so that the wrist stream is captured directly by the controller with lower latency.

Egocentric sensing and spatial tracking. We use a Meta Quest headset for head/egocentric sensing and spatial tracking. It provides the head-camera image I^H through the Unity Passthrough API and the head pose T^H; throughout the paper, H denotes this head-mounted egocentric camera. A Quest controller attached to the gripper provides the wrist pose T^W. The AR passthrough view matches the demonstrator's view, giving global context beyond the local wrist camera, and the controller buttons are used to control recording.

Synchronized wireless streaming. We synchronize all modalities over UDP. The headset streams I^H, T^H, T^W, and the recording state at 72 Hz to the Raspberry Pi controller. The controller captures I^W and gripper width g at 20 Hz, aligns them to the closest headset timestamp, and sends the combined packet to a workstation. In our setup, the workstation uses an Intel Core i5-13600KF CPU and an NVIDIA RTX 4070 GPU to store episodes, compute online coverage scores, and return AR guidance. The system runs on a standard 1000 M WLAN router with cross-modality synchronization within 20 ms (less than camera exposure time) and end-to-end workstation latency within 100 ms.

Each aligned observation is $o = \{I^W, I^H, T^W, T^H, g\}, \quad T^W, T^H \in SE(3)$, where I^W and I^H are the wrist-view and head/egocentric images, T^W and T^H are the wrist and head poses in the shared collection world frame, and g denotes the gripper width.

3.2 Online Data Curation

Beyond recording demonstrations, EgoGuide uses the live EgoGuide-UMI stream to improve data quality by guiding initial states online and filtering episodes afterward.

Online data coverage guidance. Before each episode, EgoGuide estimates whether the current state expands the dataset coverage. Since robust UMI policies rely on diverse demonstrations, we define a **data coverage score** to quantify how much the current collection state occupies an under-explored region of the existing dataset. This score is computed over three complementary signals. Wrist images reflect the diversity of policy inputs, including hand-object appearance and contact configuration. Wrist poses capture geometric and action-side diversity, since UMI demonstrations supervise end-effector motion and a narrow pose distribution usually induces a narrow action distribution. Head/egocentric images capture object arrangement, workspace layout, and contextual cues from a global viewpoint that may be invisible to the wrist camera.

For image modalities, EgoGuide extracts normalized visual features from the wrist image I^W and the head/egocentric image I^H, then compares each feature to a view-specific feature memory. For view $m \in \{W, H\}$ and encoder e, the visual similarity is the average cosine similarity to the k nearest features in the corresponding memory:

$$z^{m,e} = \bar{\phi}_e(I^m), \qquad s_{m,e} = \frac{1}{k} \sum_{j \in \mathrm{NN}_k(z^{m,e}, \mathcal{M}_{m,e})} (z^{m,e})^\top z_j^{m,e}. \tag{1}$$

We compute this score with both DINOv2 [29] and CLIP [30] so the guidance signal captures complementary visual cues from local appearance and geometry to object- and scene-level semantics.

For geometric coverage, EgoGuide uses the wrist pose T^W. We represent it by translation and quaternion orientation, and compute a nearest-neighbor pose similarity in a memory $\mathcal{M}_p$ using a normalized translation-rotation distance. High similarity indicates a repeated wrist configuration, while low similarity indicates an under-explored region of the pose space.

All similarity scores are converted into novelty percentiles with respect to the current memories, so higher values consistently mean higher coverage gain. This normalization is important for user feedback: raw similarity scores have encoder- and modality-dependent ranges, often with limited perceptual contrast, whereas demonstrators mainly need the relative novelty of the current state. We aggregate the two encoder-specific image scores within each view and display three 0–100 guidance values in the AR interface: wrist-view novelty, egocentric-context novelty, and wrist-pose novelty. The estimator runs at 2 Hz, allowing the demonstrator to adjust the initial hand pose, object arrangement, viewpoint, or workspace layout before recording an episode. Since these initial conditions

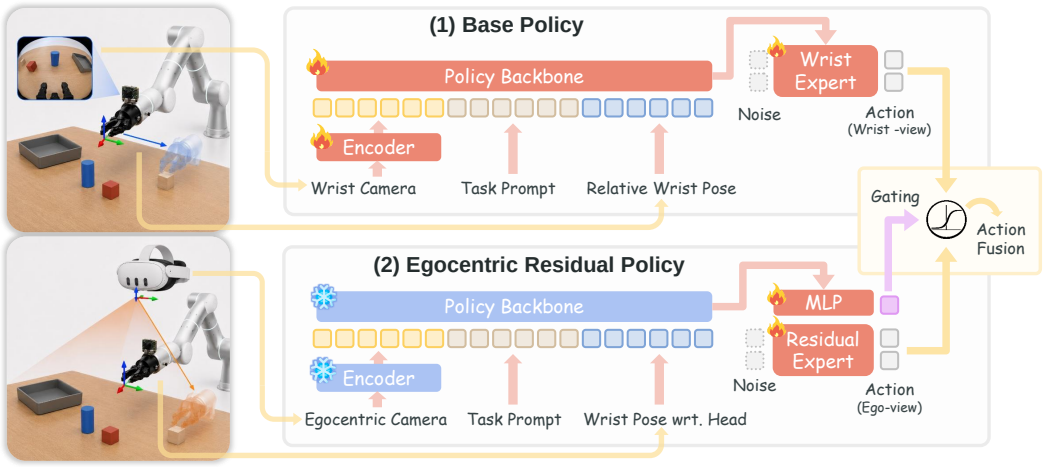


Figure 3: Overview of **G**ated **E**gocentric **R**esidual **P**olicy (**GERP**). A wrist-only base policy predicts a nominal action from I^W. An independent egocentric residual branch receives the head/egocentric image I^H and the wrist pose with respect to the head frame, then predicts an residual action in the same wrist-relative action space. A learned gate blends this candidate with the base action.

affect approach direction, contact sequence, occlusion patterns, and recovery behavior, the feedback improves trajectory diversity without on-policy rollouts like DAgger [19].

Partial demonstration. Because initial-state guidance mainly affects early-stage variation, EgoGuide also supports starting a new recording from the middle of a task. The demonstrator can start collection from an underrepresented intermediate state or subtask. This gives demonstrators more flexibility, turns later-stage states into controllable starts, and improves coverage beyond the beginning of the task. We show the efficacy of partial demonstration in Sec. 5.

Static data filtering. After an episode ends, EgoGuide applies deterministic quality checks before the sample enters the memory. The system rejects an episode if: (1) missing any required modality, (2) is too short, (3) wrist or head motion contains implausible jumps, (4) the images are severely blurred or outside a normal brightness range. Blur is measured by the Laplacian variance of the gray-scale image, and pose discontinuities are detected using linear and angular velocity thresholds. These checks remove sensor failures and physically inconsistent trajectories while avoiding a learned filter that could bias the dataset. Empirically, this stage discards around 2%–5% of collected episodes. Accepted episodes are downsampled and added to the memory.

4 Algorithm

Online coverage guidance improves UMI demonstration diversity, but challenging tasks still require a policy that can use the head/egocentric observations collected by EgoGuide. Directly conditioning the full action policy on a moving human viewpoint can be brittle because the head pose is not a controlled robot action, the view can be noisy or redundant, and the visible hand-held device introduces embodiment mismatch. We therefore propose **G**ated **E**gocentric **R**esidual **P**olicy (**GERP**), which treats egocentric perception as a gated alternative action cue rather than a replacement for wrist-view control. The wrist branch provides the default local manipulation behavior, while the egocentric branch proposes a complete action candidate when broader scene context is useful. Unlike active-perception methods that imitate or execute head motion [2, 4, 18], GERP does not model the head camera as an action output, and can be deployed with a fixed calibrated egocentric camera. Fig. 3 summarizes the design.

Stage 1: wrist-only base policy. Let $\mathbf{A}^\star$ denote the ground-truth demonstration action chunk used as supervision. Following the UMI convention, it is expressed in a wrist-relative action space to the

Task	EgoGuide	100 Demos	200 Demos	300 Demos	400 Demos
Pick Cube	✗	25% / 30.0%	40% / 42.5%	50% / 55.0%	70% / 70.0%
	✓	**30% / 47.5%**	**65% / 75.0%**	**95% / 97.5%**	**100% / 100.0%**
Pepper Sorting	✗	0% / 0.0%	10% / 12.5%	15% / 45.0%	50% / 57.5%
	✓	0% / **7.5%**	**50% / 60.0%**	**60% / 60.0%**	**75% / 77.5%**
Garlic Storage	✗	0% / 0.0%	0% / 0.0%	0% / 18.8%	0% / 16.3%
	✓	0% / **23.8%**	**15% / 38.8%**	**35% / 53.8%**	**50% / 61.3%**

Table 1: Performance scaling curves of different data collection strategies with or without EgoGuide. We report success rate and progress score as "SR / TPS".

current wrist pose, together with the gripper command. All predicted actions including the base action $\mathbf{A}^b$, the egocentric action candidate $\mathbf{A}^r$, and the fused action $\hat{\mathbf{A}}$, use this same action space.

In the first stage, we train a standard wrist-view diffusion policy. Given the wrist image, the current wrist pose, and the task instruction ℓ, the base policy predicts a nominal action chunk:

$$\mathbf{A}^b = \pi_b(I^W, T^W, \ell), \tag{2}$$

where the recorded wrist pose T^W is expressed in the shared collection/world frame. The base policy is trained with the standard flow-matching action objective toward $\mathbf{A}^\star$, and later serves as a stable local manipulation prior.

Stage 2: independent egocentric residual policy. In the second stage, the base policy is frozen and we add an independent residual action expert. The egocentric branch should interpret the wrist configuration in the coordinate system of the image it observes, so we express the current wrist pose in the current head frame as $T^{H\leftarrow W} = (T^H)^{-1}T^W$, where T^W and T^H are both recorded in the shared collection/world frame. This gives the wrist pose in the head-camera frame and aligns the pose input with I^H. Conditioned on the head/egocentric image and this head-frame wrist pose, the residual branch predicts a complete action candidate $\mathbf{A}^r$ in the same wrist-relative action space as $\mathbf{A}^b$, along with a scalar gate α. The final predicted action is formed by gated blending:

$$T^{H\leftarrow W} = (T^H)^{-1}T^W, \qquad (\mathbf{A}^r, \alpha) = \pi_r(I^H, T^{H\leftarrow W}, \ell), \qquad \hat{\mathbf{A}} = (1-\alpha)\mathbf{A}^b + \alpha\mathbf{A}^r. \tag{3}$$

This coordinate transform is only used as residual input. It does not change the action target: during training, $\mathbf{A}^r$ is supervised by the same demonstration action chunk $\mathbf{A}^\star$ as the base policy. Thus the base action, egocentric action candidate, and final action share one coordinate convention and can be mixed directly. The gate controls how much the policy moves from the stable wrist-view base action toward the egocentric candidate.

Training objective. Since the base policy is fixed in stage 2, the residual action expert directly learns the full ground-truth action $\mathbf{A}^\star$ rather than the difference $\mathbf{A}^\star - \mathbf{A}^b$. We train this branch with the same flow-matching objective, and also supervise the composed action:

$$\mathcal{L}_{\mathrm{GERP}} = \lambda_{\mathrm{res}}\mathcal{L}_{\mathrm{FM}}(\pi_r; \mathbf{A}^\star) + \lambda_{\mathrm{act}} \left\|(1-\alpha)\mathbf{A}^b + \alpha\mathbf{A}^r - \mathbf{A}^\star\right\|_2^2. \tag{4}$$

The flow-matching loss teaches the egocentric branch to generate action from head/egocentric context, while the composed-action loss trains the gate to combine this candidate with the wrist-view base action. We use a curriculum schedule for stable optimization: starts with only the residual branch loss ($\lambda_{\mathrm{act}} = 0$), then linearly increases to equal weighting ($\lambda_{\mathrm{act}} = 1$), and finally continues training with this balanced objective. At inference, GERP computes the wrist-view base action, the egocentric action candidate and the gating value, then executes the fused action in Equation (3).

5 Experiments

5.1 Experimental Setup

Robot deployment. All policies are evaluated on a Flexiv Rizon 4 robot with a Grav gripper. We mount the fisheye camera and Meta Quest controller in the same configuration as EgoGuide-UMI,

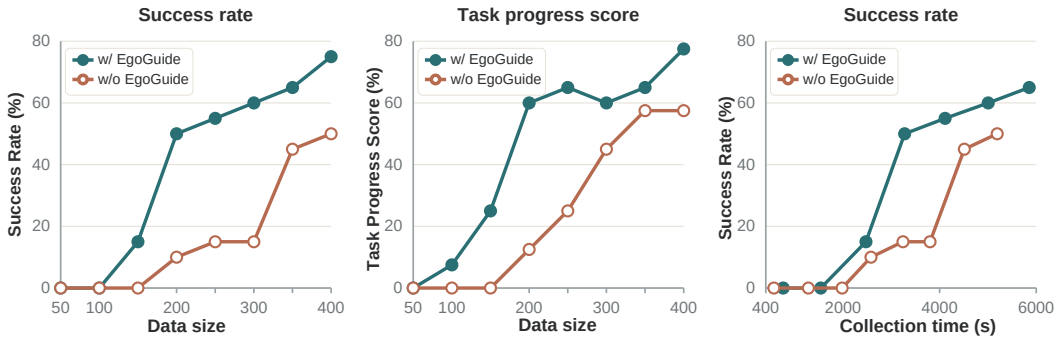


Figure 4: Data scaling comparison between unguided and EgoGuide-guided collection. For each dataset size, we train the same $\pi_{0.5}$ policy and report both success rate and progress score.

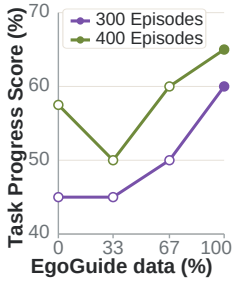


Figure 5: Mixing regular data and EgoGuide data.

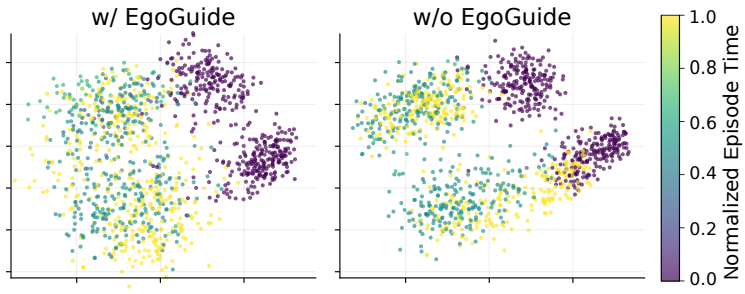


Figure 6: t-SNE visualization of wrist-camera CLIP features.

and use a fixed Meta Quest headset as the head/egocentric camera at a third-person viewpoint close to the demonstrator's viewing direction. Data collection and robot evaluation use disjoint rooms and scene layouts, while the fixed setup keeps camera intrinsics and relative 6D pose consistent with training. We run 20 trials per policy with object-location randomization.

Policy implementation. We use $\pi_{0.5}$ as the base model. Images are resized to 224×224, and the action horizon is $K = 16$. We fine-tune the model for 30 K steps with batch size 128 and cosine learning rate from $2.5e-5$ to $2.5e-6$. For GERP, we first train the wrist-view base policy, then freeze it and train the egocentric branch for 30 K steps. The composed-action loss is introduced after a 15 K-step egocentric warm-up, linearly ramped over 10 K steps, and kept for the remaining steps.

Metrics and tasks. We report both binary success rate (SR) and task progress score (TPS). The task progress captures partial completion of subtasks. The tasks and defined subgoals are as follows:

1. **Pick Cube**: Pick up a yellow cube (TPS $= 50\%$) and put it in a box (TPS $= 100\%$).
2. **Pepper Sorting**: Pick up a green pepper (TPS $= 25\%$) and put it in the green tray (TPS $= 50\%$), then pick up a chili pepper (TPS $= 75\%$) and put it in the red tray (TPS $= 100\%$).
3. **Garlic Storage**: Open the top drawer (TPS $= 25\%$), pick up the garlic (TPS $= 50\%$), put it in the drawer (TPS $= 75\%$), then close the drawer (TPS $= 100\%$).
4. **Rubik's Cube**: Grasp the Rubik's cube (TPS $= 50\%$) and rotate $90°$ (TPS $= 100\%$).

5.2 Comparison on Online Data Curation

We first evaluate whether EgoGuide improves the utility of collected demonstrations, using the wrist-only $\pi_{0.5}$ policy for all experiments. For each task, users collect two datasets with the same target size: one using standard unguided collection and one using EgoGuide feedback. Each session contains 100 samples; we reset the collection location and EgoGuide memory between sessions, while keeping background, scene, and lighting matched for fair comparison.

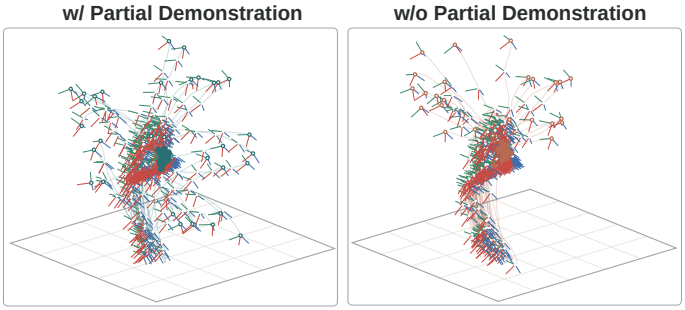


Figure 7: Trajectory distribution of EgoGuide with and without partial demonstration (25 episodes each). Partial demonstration substantially increases data coverage.

Task	#Demos	Wrist Only	Wrist+Ego Direct	GERP
Pick Cube	200	65% / 75.0%	70% / 75.0%	**80% / 90.0%**
Pepper Sorting	400	75% / 77.5%	65% / 72.5%	**80% / 87.5%**
Garlic Storage	400	50% / 61.3%	50% / **72.5%**	**55%** / 70.0%
Rubik's Cube	300	30% / 37.5%	70% / 77.5%	**80% / 82.5%**

Table 2: Comparison of egocentric policies. We report "SR / TPS". Wrist+Ego Direct feeds wrist and head/egocentric images directly into $\pi_{0.5}$. Best results are bold and second-best are underlined.

Performance and efficiency comparison. Tab. 1 and Fig. 4 show that EgoGuide improves policy performance and reduces the required data. On Pepper Sorting with 200 demonstrations, for example, EgoGuide raises success from 10% to 50% and reaches 50% success using only half the data. The gains also appear on long-horizon tasks, even though EgoGuide only guides collection starts. EgoGuide largely preserves UMI collection efficiency: real-time guidance adds 4.3 s per sample on average, but the total-time scaling curve in Fig. 4 remains better than unguided collection. Fig. 5 further shows that mixing more EgoGuide data into a regular dataset steadily improves performance, indicating consistent data quality.

Dataset distribution. We visualize camera image features to compare diversity and coverage. Fig. 6 shows CLIP wrist-image features as an example: EgoGuide covers a larger feature-space region. Measured by feature covariance trace, EgoGuide improves variance by 5%, 4%, 3%, and 4% across the two camera views and two feature models.

Partial demonstration. Fig. 7 compares EgoGuide with and without the partial-demonstration option described in Sec. 3.2. Mid-task starts prevent trajectories from collapsing to similar later states and substantially increase spatial coverage.

5.3 Egocentric Residual Policy Evaluation

We compare GERP with *Wrist Only*, the default UMI setting, and *Wrist+Ego Direct*, which treats the egocentric camera as a regular head-view input and feeds both camera views to the policy. Tab. 2 shows that on Pepper Sorting, GERP improves success rate and progress score by 5%–10% over *Wrist Only*. *Wrist+Ego Direct* can sometimes degrade performance because the moving egocentric view mismatches policy pretraining and distracts training, while GERP uses it as a gated residual cue. We also visualize the gating value throughout the rollout process in the appendix.

6 Conclusion

We present EgoGuide, a robot-free demonstration collection system that augments UMI-style data with synchronized wrist and head/egocentric observations. Its online coverage guidance and partial-demonstration support help demonstrators collect more diverse and useful trajectories, while static filtering removes common sensor failures. We also introduce GERP, which uses the head/egocentric view as a gated residual action cue on top of a stable wrist-view base policy. Together, EgoGuide and GERP improve data efficiency and policy robustness in real-world manipulation.

References

[1] 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.

[2] H. Xiong, X. Xu, J. Wu, Y. Hou, J. Bohg, and S. Song. Vision in action: Learning active perception from human demonstrations. *arXiv preprint arXiv:2506.15666*, 2025.

[3] Y. Zou, C. Shi, W. Yu, H. Xue, J. Lv, Y. Pan, C. Wen, and C. Lu. Activeglasses: Learning manipulation with active vision from ego-centric human demonstration. *arXiv preprint arXiv:2604.08534*, 2026.

[4] Q. Zeng, C. Li, J. S. John, Z. Zhou, J. Wen, G. Feng, Y. Zhu, and Y. Xu. Activeumi: Robotic manipulation with active perception from robot-free human demonstrations. *arXiv preprint arXiv:2510.01607*, 2025.

[5] 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.

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

[7] 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.

[8] Z. Zhaxizhuoma, K. Liu, C. Guan, Z. Jia, Z. Wu, X. Liu, T. Wang, S. Liang, P. Chen, P. Zhang, et al. Fastumi: A scalable and hardware-independent universal manipulation interface with dataset. In *Conference on Robot Learning*, pages 3069–3093. PMLR, 2025.

[9] Y. Xu, L. Wei, P. An, Q. Zhang, and Y.-L. Li. exumi: Extensible robot teaching system with action-aware task-agnostic tactile representation. *arXiv preprint arXiv:2509.14688*, 2025.

[10] T. Cheng, K. Chen, L. Chen, L. Zhang, Y. Zhang, Y. Ling, M. Hamad, Z. Bing, F. Wu, K. Sharma, et al. Tacumi: A multi-modal universal manipulation interface for contact-rich tasks. *arXiv preprint arXiv:2601.14550*, 2026.

[11] F. Liu, C. Li, Y. Qin, J. Xu, P. Abbeel, and R. Chen. Vitamin: Learning contact-rich tasks through robot-free visuo-tactile manipulation interface. *arXiv preprint arXiv:2504.06156*, 2025.

[12] L. Wu, C. Yu, J. Ren, L. Chen, Y. Jiang, R. Huang, G. Gu, and H. Li. Freetacman: Robot-free visuo-tactile data collection system for contact-rich manipulation. *arXiv preprint arXiv:2506.01941*, 2025.

[13] O. Rayyan, J. Abanes, M. Hafez, A. Tzes, and F. Abu-Dakka. Mv-umi: A scalable multi-view interface for cross-embodiment learning. *arXiv preprint arXiv:2509.18757*, 2025.

[14] Z. Wang. Umi-3d: Extending universal manipulation interface from vision-limited to 3d spatial perception. *arXiv preprint arXiv:2604.14089*, 2026.

[15] M. Xu, H. Zhang, Y. Hou, Z. Xu, L. Fan, M. Veloso, and S. Song. Dexumi: Using human hand as the universal manipulation interface for dexterous manipulation. *arXiv preprint arXiv:2505.21864*, 2025.

[16] X. Xu, J. Park, H. Zhang, E. Cousineau, A. Bhat, J. Barreiros, D. Wang, and S. Song. Hommi: Learning whole-body mobile manipulation from human demonstrations. *arXiv preprint arXiv:2603.03243*, 2026.

[17] R. Nai, B. Zheng, J. Zhao, H. Zhu, S. Dai, Z. Chen, Y. Hu, Y. Hu, T. Zhang, C. Wen, et al. Humanoid manipulation interface: Humanoid whole-body manipulation from robot-free demonstrations. *arXiv preprint arXiv:2602.06643*, 2026.

[18] J. Yu, Y. Shentu, D. Wu, P. Abbeel, K. Goldberg, and P. Wu. Egomi: Learning active vision and whole-body manipulation from egocentric human demonstrations. *arXiv preprint arXiv:2511.00153*, 2025.

[19] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[20] J. Zhang and K. Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.

[21] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

[22] J. Fang, W. Chen, H. Xue, F. Zhou, T. Le, Y. Wang, Y. Zhang, J. Lv, C. Wen, and C. Lu. Robopocket: Improve robot policies instantly with your phone. *arXiv preprint arXiv:2603.05504*, 2026.

[23] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. *arXiv preprint arXiv:2108.03298*, 2021.

[24] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[25] S. Dasari, F. Ebert, S. Tian, S. Nair, B. Bucher, K. Schmeckpeper, S. Singh, S. Levine, and C. Finn. Robonet: Large-scale multi-robot learning. *arXiv preprint arXiv:1910.11215*, 2019.

[26] J. Hejna, S. Mirchandani, A. Balakrishna, A. Xie, A. Wahid, J. Tompson, P. Sanketi, D. Shah, C. Devin, and D. Sadigh. Robot data curation with mutual information estimators. *arXiv preprint arXiv:2502.08623*, 2025.

[27] A. S. Chen, A. M. Lessing, Y. Liu, and C. Finn. Curating demonstrations using online experience. *arXiv preprint arXiv:2503.03707*, 2025.

[28] C. Agia, R. Sinha, J. Yang, R. Antonova, M. Pavone, H. Nishimura, M. Itkina, and J. Bohg. Cupid: Curating data your robot loves with influence functions. *arXiv preprint arXiv:2506.19121*, 2025.

[29] 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.

[30] 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.

A Implementation Details

A.1 Task Details

We provide additional details on the evaluation randomization protocol. For each trial, objects are manually reset within predefined ranges while ensuring reachability and avoiding initial object overlap. All methods on the same task are evaluated under the same randomization protocol. Once execution starts, no human correction is allowed unless a safety stop is required.

- **Pick Cube.** The cube position and the target box are randomized within a tabletop region of approximately 30 cm $\times$ 30 cm. Its yaw orientation is sampled within 45 degrees.
- **Pepper Sorting.** The green pepper and chili pepper and the two trays are independently randomized within a region of approximately 40 cm $\times$ 40 cm. Their yaw orientations are randomized.
- **Garlic Storage.** The garlic position is randomized within approximately 20 cm $\times$ 40 cm. The drawer cabinet is always facing the robot and shifting left to right within 40cm. The drawer always starts from a closed state.
- **Rubik's Cube Rotation.** The cube position is randomized within approximately 30 cm $\times$ 30 cm. Its yaw orientation is sampled within 20 degrees. The blue side always faces upward.

A trial terminates when the task succeeds, the end effector deviates from regular working area for 5 seconds, the object becomes unrecoverable, or the robot triggers a safety stop. Safety-stop trials are counted as failures.

A.2 Data Collection Statistics

Tab. 3 summarizes the datasets after synchronization, static filtering, and downsampling. We report both episode-level and frame-level statistics.

Task	EgoGuide	# Episodes	# Frames	Frames per Episode	Duration per Episode (s)	Rejected Episodes	Scenes
Pick Cube	✗	400	25,434	63.59	3.18	–	4
Pick Cube	✓	400	35,473	88.68	4.43	22	4
Pepper Sorting	✗	400	50,410	126.03	6.30	–	4
Pepper Sorting	✓	394	45,189	114.69	5.73	24	4
Garlic Storage	✗	400	82,597	206.49	10.32	–	4
Garlic Storage	✓	400	60,712	151.78	7.59	9	4

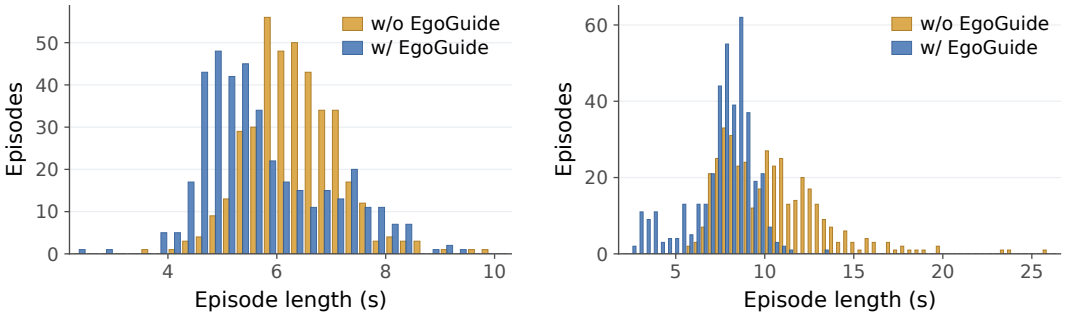
Table 3: Dataset statistics after synchronization, filtering, and downsampling. "Frames" denotes synchronized observation-action frames used for training.

Fig. 8 compares the episode-length distributions measured by the number of synchronized frames per episode. EgoGuide produces a broader distribution because the dataset contains both full demonstrations and partial demonstrations starting from intermediate states. This increases coverage of later-stage interactions and recovery behaviors that are less frequent in standard full-trajectory collection.

A.3 Training Details

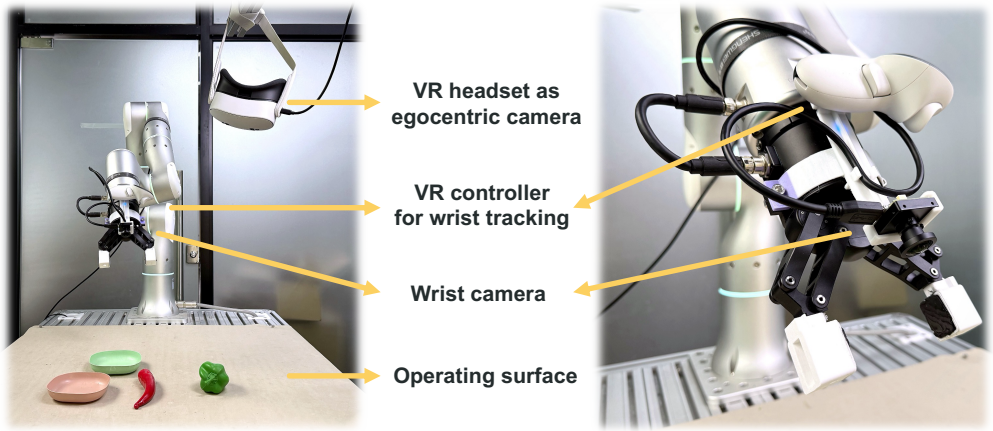
All datasets are split at the episode level, with a validation ratio of 3%. This avoids temporally adjacent frames from the same trajectory appearing in both splits. Each training sample contains an observation history of 1 frame for camera inputs and 2 frames for low-dim inputs, and an action chunk of length $K = 16$.

The action target follows the wrist-relative convention and consists of relative translation, relative rotation, and gripper command. Translation, rotation, and gripper targets are normalized using statistics from the training set. Episodes are not downsampled and keep 20 Hz at training.



(a) Pepper Sorting (b) Garlic Storage

Figure 8: Episode-length distribution after synchronization. EgoGuide includes both full and partial demonstrations, resulting in a broader frame-count distribution.



(a) Testing Environment (b) Wrist Bracket

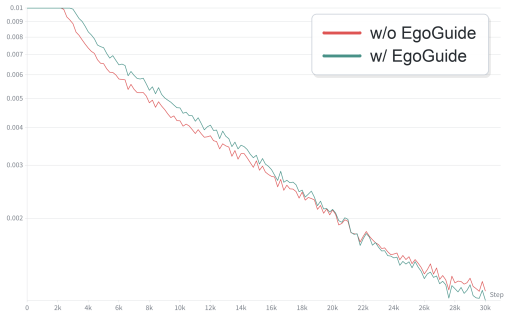
Figure 9: Robot deployment setup. The wrist camera is fixed to the robot end-effector with a custom bracket. For egocentric-policy experiments, an additional fixed egocentric camera provides global scene context.

For all experiments, we train the $\pi_{0.5}$-based policy on 4 NVIDIA H100 GPUs and 64-core CPU. A 30 K-step training usually takes 5 hours for wrist-only policies, and 3 hours for additional GERP residual policy. We use the checkpoints of last iteration for evaluation. The evaluation is on a workstation with one NVIDIA 4070 GPU and Intel i9 7900X CPU.

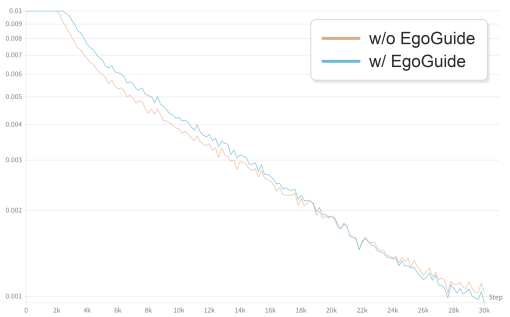
Fig. 9 shows the robot deployment setup and the wrist-camera mounting bracket. The wrist camera is rigidly attached to the robot end-effector using a fixed custom bracket. The bracket keeps the camera pose stable relative to the gripper and approximately matches the wrist-camera viewpoint used during robot-free UMI data collection.

For egocentric-policy experiments, the egocentric camera is fixed in the workspace rather than actively controlled. It is placed at approximately 70 cm height from the workspace desktop. This provides a stable global view while avoiding major occlusion from the robot arm.

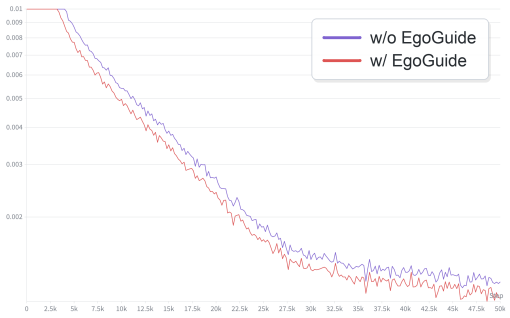
The policy runs at 10 Hz. Predicted actions are clipped by predefined translation, rotation, gripper, and workspace limits before execution, and then converted to joint control signals by the internal inverse-kinematics tool of FlexivRDK. Each trial starts from the same robot home pose, followed by task-specific scene randomization.



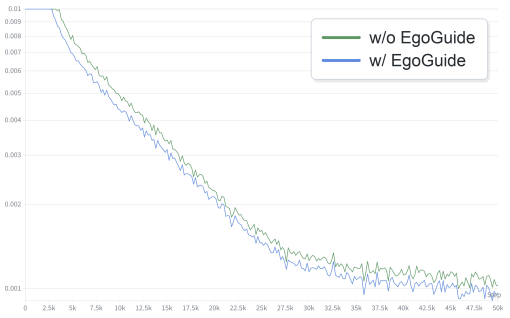
(a) Pepper Sorting, 400 samples



(b) Pepper Sorting, 300 samples



(c) Garlic Storage, 400 samples



(d) Garlic Storage, 300 samples

Figure 10: Examples of loss curves for various tasks and data scale. The loss values (Y-axis) are truncated and presented in log-scale for better visualization.

Method	Setting 1	Setting 2	Setting 3
Wrist+Ego Direct	65% / 72.5%	60% / 65.0%	40%/ 40.0%
GERP	80% / 87.5%	75% / 80.0%	70% / 80.0%

Table 4: Pepper Sorting performance under different fixed egocentric camera placements. Each entry reports success rate and task progress score as SR / TPS.

B Additional Comparisons

B.1 Loss Behaviors

Although the training loss does not directly reflect the final model performance or task success rate, it may still provide useful indications about the quality of the post-training data. As shown in Fig. 10, we present the post-training loss curves on the *Pepper Sorting* and *Garlic Storage* tasks. In the early stage of post-training, the loss decreases differently when trained with data of varying quality. However, in the later stage, the model trained with EgoGuide data generally converges to a lower loss, which is consistent with EgoGuide providing more informative training data. For *Pepper Sorting* task, the model trained without EgoGuide initially shows faster convergence. This may be because the non-EgoGuide data is more concentrated and easier to fit at the beginning; nevertheless, its loss soon saturates and fails to decrease further, while EgoGuide enables continued optimization and achieves a lower final convergence loss.

B.2 Sensitivity to Fixed Egocentric Camera Placement

We further evaluate the robustness of egocentric policies to different fixed head-camera placements on the Pepper Sorting task. During testing, the wrist-camera and robot setup and policy checkpoint are kept unchanged; only the fixed egocentric camera pose is varied. We compare under three

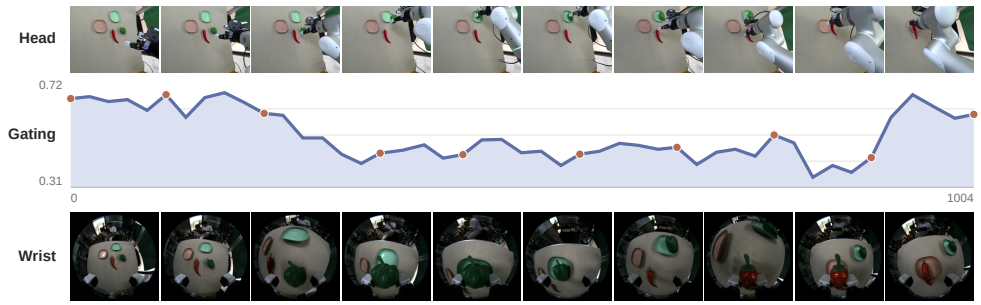


Figure 11: Gating behavior during execution.

camera placements in Tab. 4. Across different camera placements, GERP is more stable than the direct Wrist-Ego baseline.

C Visualizations

C.1 Visualization of Gating Behavior

Fig. 11 shows the gating value during evaluation. Notably, the gating value increases when the target object is occluded or missing from the wrist view (right side of the figure).

C.2 Additional Visualization of Feature Distribution

Fig. 12 (on the next page) presents additional t-SNE visualizations for all other combinations of camera modalities (wrist and egocentric) and feature encoders (CLIP and DINOv2) on Pepper Sorting task datasets. Similar to the main-paper result, EgoGuide data consistently exhibits broader feature-space coverage across all settings. The effect is observed in both local wrist-view observations and global egocentric observations, indicating that the online guidance mechanism increases diversity in both manipulation-centric and scene-level visual states.

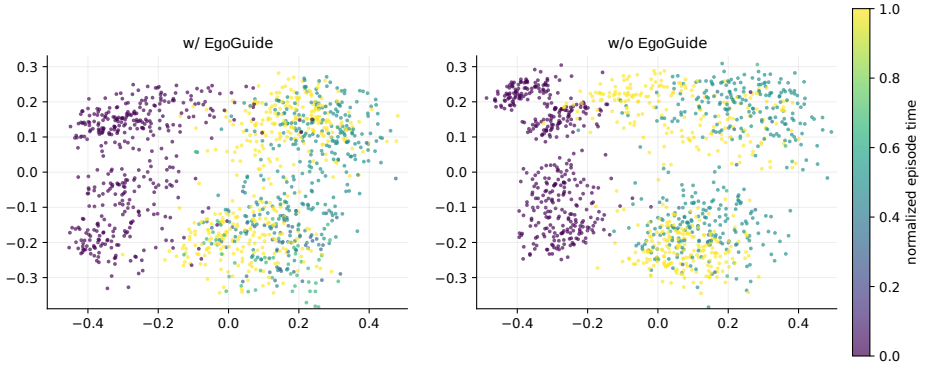
C.3 Qualitative Data Comparison

Fig. 13 (on the next two pages) compares representative demonstrations collected with and without EgoGuide. Samples collected with EgoGuide exhibit greater variation in viewpoints, object placements, hand poses, and scene configurations, whereas unguided collection tends to produce more repetitive observations. These qualitative results are consistent with the feature-space analysis and support our claim that EgoGuide improves dataset diversity by encouraging exploration of underrepresented states during collection.

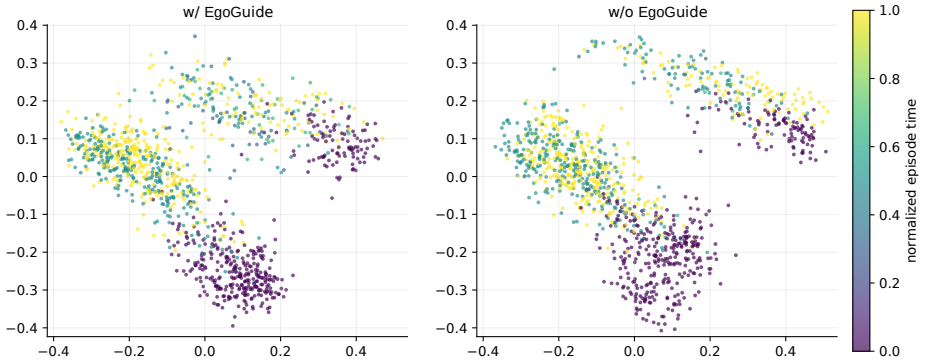
D Discussion

Q1: Why can guiding the initial state improve the diversity of the whole episode? Although EgoGuide provides guidance before recording, the initial state is a strong causal factor for the rest of the trajectory. Different initial hand poses, object layouts, and viewpoints lead to different approach directions, contact sequences, occlusion patterns, and recovery behaviors. Therefore, improving the coverage of initial visual-geometric states is a simple but effective way to induce more diverse full demonstrations, without requiring policy rollouts or robot interaction. Furthermore, the introduction of **partial demonstration** indicates that any intermediate state of a task could be regarded as an *initial state* for data collection, which allows controlling the data quality of the whole episode.

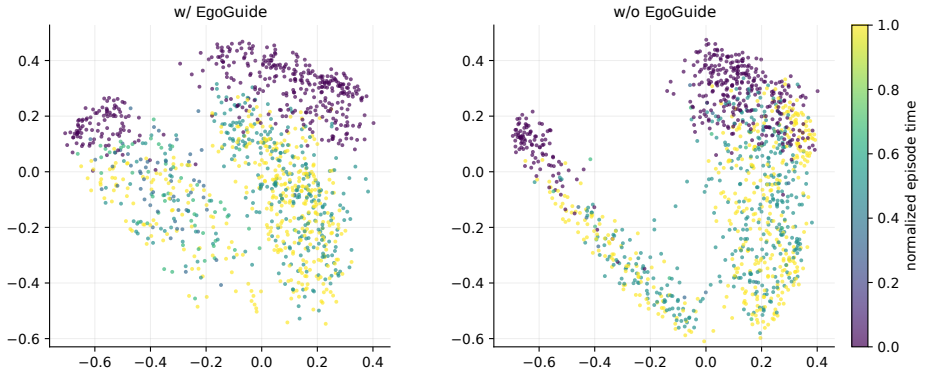
Q2: Does partial demonstration introduce more late-stage samples, thus result in data imbalance? Yes. Partial demonstration intentionally increases the coverage of intermediate and late-stage



(a) Wrist camera, DINO feature.



(b) Egocentric camera, CLIP feature.



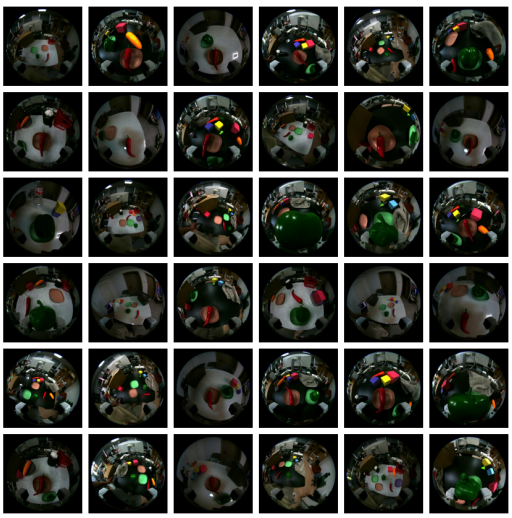
(c) Egocentric camera, DINO feature.

Figure 12: t-SNE visualization of feature distributions. EgoGuide enhances data diversity and unguided data is more concentrated in fewer regions of the projected feature space.

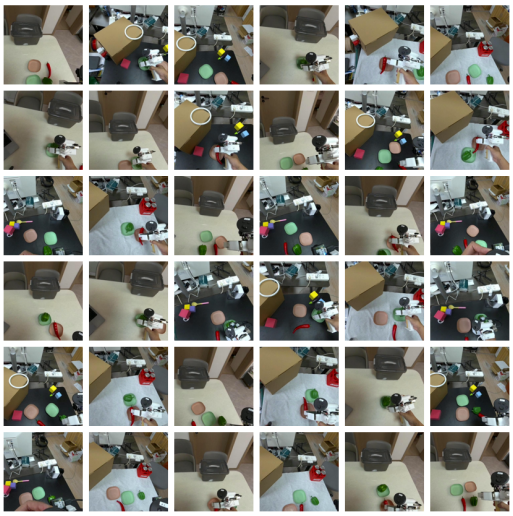
states. However, more data samples do not necessarily indicate training imbalance. In long-horizon tasks, full demonstrations often share similar later-stage configurations, so simply collecting more complete episodes may still leave these regions under-diversified. Therefore the late stages are harder to learn so more samples for late stages could help to train a temporally balanced model.

Q3: Why not use DAgger for data quality? DAgger is designed for a different setting: it requires a learned policy to roll out, visit its own states, and query expert corrections. EgoGuide targets the

(a) Wrist camera, w/o EgoGuide



(b) Wrist camera, w/ EgoGuide



(c) Egocentric camera, w/o EgoGuide

(d) Egocentric camera, w/ EgoGuide

Figure 13: Examples of collected episodes for Pepper Sorting task.

earlier robot-free data collection stage, where a reliable policy may not yet exist and where on-robot interaction is undesirable. Our goal is therefore training-free data improvement: provide online collection guidance before policy training, rather than relying on policy-dependent data aggregation.

Q4: Why use spatial wrist pose for EgoGuide instead of object-relative pose? The coverage score is used to detect whether the current collection state is redundant, not to define the final control representation. Spatial wrist pose is useful because it reflects action-side geometric diversity: repeated wrist poses often correspond to repeated approach directions and similar supervised end-effector motions. Object-relative pose could be informative, but it would require reliable object tracking or pose estimation in unconstrained robot-free scenes. EgoGuide instead combines wrist pose with wrist-view and egocentric visual features, which keeps the interface simple while still capturing both geometry and scene context.

Q5: Why is EgoGuide especially suitable for crowdsourced collection? Crowdsourcing can scale robot-free demonstration collection, but it also makes quality control harder because users differ in skill and recording behavior. And the crowdsourced demonstrators may unintentionally

produce repetitive, low-diversity, or shortcut-like demonstrations. EgoGuide addresses this with lightweight online feedback and deterministic post-recording checks. The AR novelty score tells users what kind of state is worth collecting, while static filtering removes common failures such as missing modalities, blur, abnormal brightness, and implausible pose jumps. This makes the collection protocol more standardized without requiring expert supervision for every episode.

E Limitation

Overhead of EgoGuide. Data collection with EgoGuide requires additional time for feature-memory computation (about 2 s) and scene and UMI pose adjustment (about 3 s). The feature computation could be further optimized by engineering efforts.

Ergonomics concerns. Our volunteers report fatigue after collecting data with an AR headset. We currently alleviate this by dividing dataset collection into short sessions with sufficient rest, and we expect the headset to be replaced by AR glasses in the future.