GAZE2ACT: GAZE-GUIDED VISION-LANGUAGE-ACTION POLICIES FOR INTERACTIVE ROBOT MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Human–robot interaction requires clear communication of task goals and intended actions. Vision-language-action (VLA) policies support task specification through language. However, specifying exactly which object to manipulate and where to act often requires lengthy descriptions, and even detailed instructions can be difficult for VLA policies to ground accurately in the scene. We investigate human gaze as a complementary interaction channel for expressing and updating spatial intent during manipulation. To support this human-in-the-loop interaction, we propose Gaze2Act, a framework that integrates gaze into VLA-based robot manipulation. Gaze2Act grounds first-person selections in the robot view through marker-free cross-view matching, preserving the intended object and interaction location. The resulting semantic and spatial cues guide both visual interpretation and action generation. Users can select objects and interaction regions and update targets during execution without sustained fixation. Our evaluation covers 17 real-robot tasks on a Unitree G1 humanoid, with Gaze2Act achieving state-of-the-art intent accuracy and task success against the evaluated VLA baselines. Interface comparisons and a user study demonstrate that gaze supports natural, low-burden human-in-the-loop interaction.

1 INTRODUCTION

Human-in-the-loop robot manipulation requires users to specify which object the robot should manipulate, indicate where to interact with it, and revise the target when needed. For example, a user may select one cup among visually similar candidates or indicate a specific interaction region on an object (Mo et al., 2021). A practical interface should allow users to provide this spatial guidance without repeatedly formulating detailed descriptions.

Vision-Language-Action (VLA) models (Black et al., 2024; Bjorck et al., 2025; Kim et al., 2024) represent a promising step toward bridging perception and action through a unified language interface. However, most existing VLA systems rely on language as the primary channel for specifying intent, implicitly assuming that intent can be fully described in words. Although language conveys task goals, specifying a particular instance or interaction region through words alone can require detailed descriptions that are cumbersome for users to provide. Gaze offers a direct way to communicate these spatial choices by looking at the intended object or region.

Recent work attempts to mitigate this limitation by translating language into explicit spatial representations, such as bounding boxes or segmentation masks, generated by vision-language models or language-prompted detectors (Huang et al., 2025; Li et al., 2025a; Yu et al., 2025; Wang et al., 2026). However, these approaches still inherit the intrinsic ambiguity of language. Linguists have found that most of expressions are descriptive, i.e., implicit referring expressions (Levinson, 1983; Krahmer & Van Deemter, 2012), which makes it difficult for robots to identify the correct object (Jiang et al., 2026). Even with explicit spatial representations, the intended object or region must still be inferred from the user's words. This motivates a complementary interface through which users can directly specify spatial targets and update them when needed.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

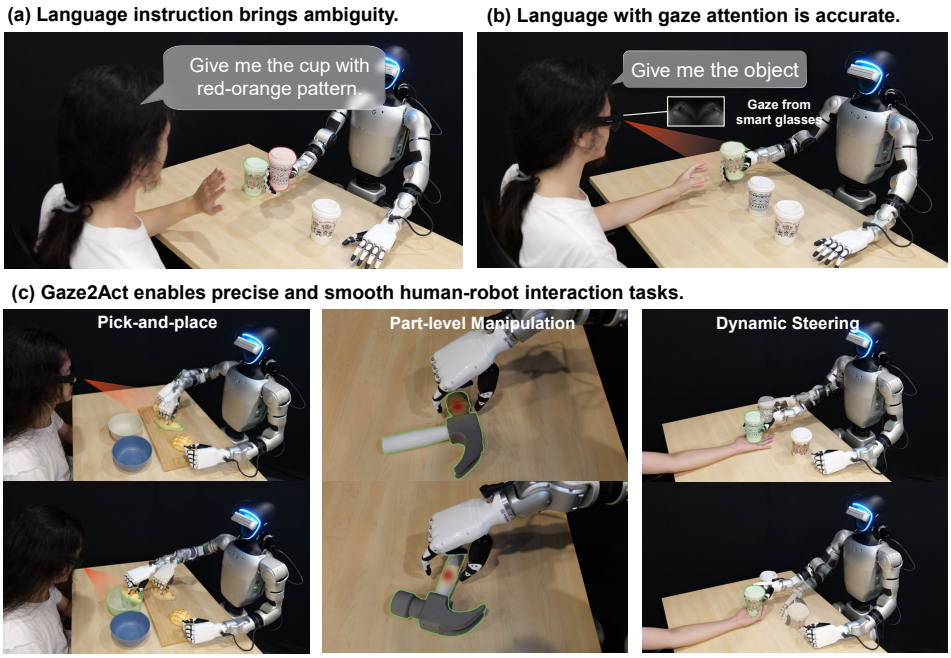


Figure 1: Motivation of gaze in human-robot interaction and new capabilities enabled by Gaze2Act.

Human gaze provides such an interface. Decades of cognitive studies show that gaze precedes and guides manipulation, with the eyes fixating on the target before the hand begins to move (Johansson et al., 2001; Flanagan & Johansson, 2003). This eye-leads-hand coordination motivates gaze as a cue for manipulation intent. In a human-in-the-loop setting, users can look at an object or interaction region to specify where the robot should act, complementing language-based task instructions, as illustrated in Fig. 1. Beyond its behavioral significance, gaze is also becoming increasingly accessible as a practical interaction modality. With eye tracking rapidly integrated into AR/VR and XR glasses from major industry players, gaze is emerging as a new paradigm for human-machine interaction, with growing real-world adoption. These properties motivate gaze as an interface through which users can participate in selecting and updating the spatial targets of robot manipulation. This raises a key question: *How can human gaze reliably and seamlessly guide VLA-based robot manipulation in a human-in-the-loop setting?*

To answer this question, we propose Gaze2Act, a framework that enables users to guide VLA-based manipulation through gaze in real time. Gaze2Act uses language to specify the high-level task objective, while leveraging gaze to resolve target ambiguity and provide fine-grained spatial cues during execution. A central challenge is that human gaze is observed from a first-person perspective and cannot be directly aligned with the robot's viewpoint. We address this by introducing a marker-free cross-view semantic matching mechanism based on visual foundation models, which projects gaze into the robot's observation space without requiring camera calibration or external markers. Once aligned, Gaze2Act integrates gaze-conditioned intent into the policy through perception-level prompting and action-level grounding. This allows the robot to attend to relevant regions and execute precise, dynamically updated interactions, while preserving the prior knowledge of the pretrained policy. Our contributions are summarized as follows:

- We investigate **human gaze as a human-in-the-loop interface for VLA policies**, enabling users to precisely specify target objects, interaction locations, and placement destinations, and reliably guide robot behavior as their intent changes during manipulation.
- We propose **Gaze2Act, which grounds first-person gaze in the robot view** through marker-free cross-view matching, preserving the selected object identity and intended interaction location to provide semantic and spatial guidance for VLA perception and action generation.
- **Experiments across 8 task categories and 17 tasks on a Unitree G1** demonstrate improvements in object disambiguation, part-level interaction, and target switching, while interface comparisons and a user study support natural, low-burden interaction.

2 Related Work

2.1 Gaze-Guided Robot Manipulation

Gaze has been used both to improve manipulation policy learning and to support online robot guidance. GazeVLA uses gaze annotations in human egocentric demonstrations to supervise intention prediction alongside action learning (Li et al., 2026), while Gaze-Regularized VLA aligns visual attention with predicted gaze heatmaps to emphasize task-relevant regions (Pani & Yang, 2026). Neither receives the user's current gaze as an interaction input at deployment. Online gaze interfaces support assisted manipulation (Shafti et al., 2019; Tay et al., 2026), with GAMMA translating gaze into skill selection and parameterized robot API calls. Gaze2Act learns to condition action generation on gaze-grounded semantic and spatial cues during training, while using the user's gaze at inference to support human-in-the-loop guidance and intent updates.

2.2 Explicit Spatial Conditioning for VLA Policies

Explicit spatial conditions complement language by specifying where a manipulation policy should act. RoboGround translates language instructions into masks of target objects and placement regions through a grounded vision-language model (Huang et al., 2025). ControlVLA obtains language-conditioned object masks through text-conditioned detection and segmentation, encoding them as additional policy conditions (Li et al., 2025a). VP-VLA and Point-VLA guide action generation through visual prompts, with targets inferred from language by a planner or specified through user-drawn boxes and pointing gestures, respectively (Wang et al., 2026; Yu et al., 2025). Gaze2Act grounds first-person gaze in the robot view and uses the resulting semantic and spatial cues to guide both visual interpretation and action generation.

3 Method

3.1 Overview

A standard VLA policy π_θ maps a language instruction l and visual observation o_t to an action sequence $a_t = \pi_\theta(l, o_t, s_t)$, where s_t denotes the proprioceptive state. Gaze2Act introduces a first-person gaze coordinate $g_t \in \mathbb{R}^2$ (provided by AR/XR glasses) as an additional conditioning signal, enabling users to update the target referent through voice-triggered gaze during execution. We implement Gaze2Act on top of GROOT N1.5 (Bjorck et al., 2025), a representative VLA backbone, in which a frozen Eagle2 VLM (Li et al., 2025b) encodes (l, o_t, s_t) into multimodal tokens $\mathbf{z} \in \mathbb{R}^{N\times d}$ that condition a DiT (Peebles & Xie, 2023) action head through cross-attention. As illustrated in Fig. 2, gaze conditioning proceeds in three stages. Cross-view gaze grounding (Section 3.2) maps g_t to a target mask m_t and fixation point p_t in the robot view. Perception-level gaze prompting (Section 3.3) renders (m_t, p_t) onto the observation as visual prompts: object contours give coarse-grained target specification, and gaze-centered heatmaps produce fine-grained interaction cues. Action-level gaze conditioning (Section 3.4) guides denoising through decoupled gaze-token cross-attention.

3.2 Cross-View Gaze Grounding

To ground the first-person gaze coordinates g_t in the robot observation o_t, we propose a multi-granularity matching framework that is training-free and calibration-free, as illustrated in Fig. 2 (a). The matching process operates sequentially across coarse and fine granularities to identify the fixated target mask m_t and fixation point p_t, respectively. We adopt DINOv3 (Siméoni et al., 2025) ViT-L/16 and extract features from five uniformly sampled transformer layers ($\{0, 5, 10, 15, 20\}$). The patch tokens from these layers are then L2-normalized and averaged to obtain the final feature map $\phi(\cdot) \in \mathbb{R}^{H\times W\times D}$. This multi-layer aggregation preserves both low-level texture and high-level semantic cues, improving robustness under viewpoint and appearance variations. In deployment, gaze grounding is voice-triggered, and the selected target is tracked until the next trigger, removing the need for continuous fixation on the intended target during interaction.

Coarse-grained matching. Given g_t as a point prompt, we first employ SAM3 (Carion et al., 2025) to segment the reference object from the egocentric image o_t^{ego}, producing a reference mask M^{ref}.

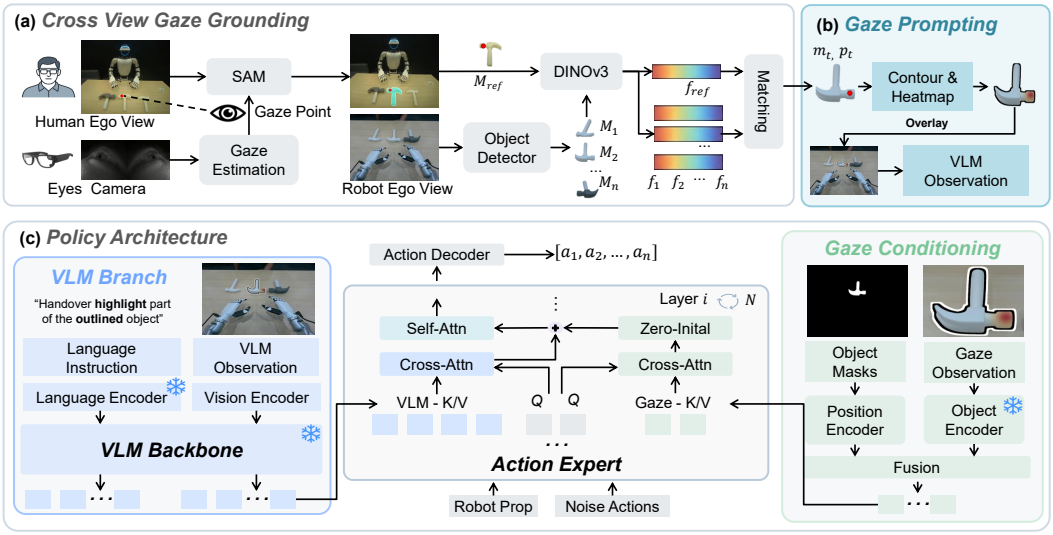


Figure 2: **Overview of Gaze2Act. (a) Cross-view gaze grounding** aligns the operator's first-person gaze with the robot view and produces gaze-grounded target cues. **(b) Perception-level gaze prompting** renders the grounded gaze cues as visual prompts on the robot observation, including object contours and gaze heatmaps. **(c) Policy architecture with action-level gaze conditioning** integrates gaze-conditioned intent into the Vision-Language-Action (VLA) policy. The Vision-Language Model (VLM) branch encodes the language instruction and gaze-prompted observation, while the gaze-conditioning branch injects gaze-grounded spatial tokens into the Diffusion Transformer (DiT) action expert through a zero-initialized decoupled cross-attention branch.

SAM3 is then applied in automatic mode on the robot observation o_t to generate a set of candidate masks $\{M_k\}_{k=1}^{K}$. The patch features within each mask are averaged to obtain $\mathbf{f}_{\text{ref}}$ and $\mathbf{f}_k$, and the mask with the highest cosine similarity to the reference feature is then selected:

$$\mathbf{f}_{\text{ref}} = \frac{1}{|M^{\text{ref}}|} \sum_{p \in M^{\text{ref}}} \phi(o_t^{\text{ego}})[p], \quad \mathbf{f}_k = \frac{1}{|M_k|} \sum_{p \in M_k} \phi(o_t)[p], \quad m_t = \arg\max_{M_k} \cos(\mathbf{f}_{\text{ref}}, \mathbf{f}_k). \tag{1}$$

Fine-grained matching. To localize the target part within the selected object, we perform a second-stage matching conditioned on the coarse mask m_t. Since the point prompt g_t provides only a sparse indication of the target part location, we extract a $(2R{+}1) \times (2R{+}1)$ neighborhood around its corresponding patch on o_t^{ego} and average the features to obtain a robust reference vector $\mathbf{f}_{\text{ref}}^{\text{fine}}$. This vector is compared against the dense features of o_t via cosine similarity to identify the corresponding point, with the argmax restricted to the coarse-grained mask m_t obtained in the previous step:

$$p_t = \arg\max_{(i,j) \in m_t} \cos\big(\mathbf{f}_{\text{ref}}^{\text{fine}}, \phi(o_t)[i,j]\big). \tag{2}$$

This constraint prevents the part-level match from being distracted by visually similar objects elsewhere in the scene. Together, the two stages output (m_t, p_t) as a shared spatial condition for perception (Section 3.3) and action generation (Section 3.4).

3.3 Perception-Level Gaze Prompting

Given the target representation (m_t, p_t) from Section 3.2, we first overlay gaze-derived cues onto the robot observation at the perception level, as illustrated in Fig. 2 (b). Object contours provide coarse-grained target specification, while gaze-centered heatmaps provide fine-grained interaction cues. These signals are directly rendered on the robot observation image, allowing the policy to receive spatial guidance through the visual channel.

Contour overlay. To provide object-level grounding, a colored contour is drawn along the boundary of m_t on o_t to produce o_t', with different colors distinguishing different targets in multi-target tasks.

The language instructions can be category-agnostic (e.g., "pick up the outlined object"), with object identity fully grounded in the visual modality rather than the language input.

Gaze heatmap. While p_t provides a precise localization signal, directly using a single point is brittle for visual policy learning. We therefore convert it into a continuous spatial heatmap to provide a soft and robust guidance signal. Specifically, a two-dimensional isotropic Gaussian heatmap $\mathcal{H}_t$ is superimposed within the m_t region, centered at p_t:

$$\mathcal{H}_t(x, y) = \exp\left(-\frac{(x - p_t^x)^2 + (y - p_t^y)^2}{2\sigma^2}\right) \cdot \mathbf{1}[(x, y) \in m_t], \tag{3}$$

where σ is the scale hyperparameter of the Gaussian kernel. During manipulation, gaze provides fine-grained spatial guidance only in the pre-contact phase. The two types of visual prompts are selected based on the spatial granularity required by the task: fine-grained, part-level interactions rely on heatmap prompting, whereas coarse object-level actions are sufficiently guided by contour cues alone. Appendix A.3 provides further details on this design.

3.4 Action-Level Gaze Conditioning

Perception-level gaze prompting makes the gaze-selected region visible to the VLM, but using gaze only as an image overlay leaves the spatial condition implicit. We therefore introduce action-level gaze conditioning that encodes the grounded mask and fixation point as a compact spatial token and injects it directly into the DiT action head, as described in Fig. 2 (c). This structured conditioning complements the visual prompt by explicitly incorporating gaze into the denoising process.

Gaze token construction. The representation (m_t, p_t) is decoupled into position and object pathways and encoded separately. The position pathway uses a two-dimensional sinusoidal code, computed at p_t when $p_t \in m_t$ and at the center of m_t otherwise, yielding $\mathbf{z}_t^{\text{pos}} \in \mathbb{R}^{d_p}$. The object pathway uses a frozen 29M-parameter DINOv3 ViT-S+/16 (Siméoni et al., 2025) as the object encoder. It encodes the target crop from the bounding box of m_t, resized to 224×224, and outputs the CLS feature $\mathbf{z}_t^{\text{obj}} \in \mathbb{R}^{d_v}$. The two pathways are concatenated and projected to the backbone dimension through a linear layer W_{fuse}, forming a single gaze-grounded spatial token (gaze token) $\mathbf{c}_t^{\text{gaze}} \in \mathbb{R}^d$:

$$\mathbf{c}_t^{\text{gaze}} = W_{\text{fuse}}\,[\mathbf{z}_t^{\text{pos}};\, \mathbf{z}_t^{\text{obj}}] + b_{\text{fuse}}. \tag{4}$$

Decoupled cross-attention. Within each DiT block, Gaze2Act adds an independent cross-attention path in parallel with the original vision-language path (Ye et al., 2023). The two paths share the same action query but use separate key-value projections, allowing the original path to preserve global vision-language context while the new path injects gaze-derived spatial constraints. Self-attention and the feed-forward network (FFN) remain unchanged from the pretrained backbone. Both branches share the same Query $\mathbf{Q} = W_Q\,\mathbf{h}$, where $\mathbf{h}$ is the noisy action and robot proprioception hidden representation. Their Keys and Values, however, come from separate sources: the original path projects VLM tokens $\mathbf{z}$ through pre-trained weights W_K, W_V to obtain $(\mathbf{K}, \mathbf{V})$, while the new path projects the gaze token $\mathbf{c}^{\text{gaze}}$ through newly added weights $W_K^{\text{gaze}}, W_V^{\text{gaze}}$ to obtain $(\mathbf{K}^{\text{gaze}}, \mathbf{V}^{\text{gaze}})$. The outputs of both paths are summed as a residual into the cross-attention sub-module output:

$$\mathbf{h}_{\text{xattn}} = \mathbf{h} + \underbrace{\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V})}_{\text{vision-language condition}} + \underbrace{\text{Attn}(\mathbf{Q}, \mathbf{K}^{\text{gaze}}, \mathbf{V}^{\text{gaze}})}_{\text{gaze spatial condition}}. \tag{5}$$

The resulting $\mathbf{h}_{\text{xattn}}$ then passes through the original FFN sub-module to the next layer. The original path handles global language-vision context, and the new path handles gaze-derived spatial constraints, both acting at every denoising step and every layer.

Stabilized gaze injection. To preserve the pretrained action prior at the start of fine-tuning, we zero-initialize the output projection of the gaze cross-attention branch while using standard initialization for the other projections. With this design, the added gaze branch is initially a no-op to the backbone output, and its influence is learned gradually from the manipulation objective. This design incurs only 4.95% additional parameters; see Appendix A.4.

4 EXPERIMENTS

We evaluate Gaze2Act on a Unitree G1 humanoid robot across eight categories and 17 tasks: 15 static manipulation tasks, one dynamic target-switch task, and one additional table-organization task for interface comparison. The experiments address three questions:

(1) Can gaze guide manipulation of the intended objects and interaction regions, and support target updates during execution? (Sec. 4.2)
(2) How does gaze compare with other interfaces in target selection accuracy, interaction time, and user workload? (Sec. 4.3)
(3) How should gaze be represented and integrated into the policy to guide action generation? (Sec. 4.4)

4.1 EXPERIMENTAL DESIGN

Datasets and Tasks. We evaluate 15 static manipulation tasks covering object selection, compositional target specification, and part-level interaction, together with a dynamic intent steering task and an additional table-organization task for interface comparison. We construct training demonstrations for each task group, covering multiple scene layouts and instruction variants. The same trajectories are used for all methods, but converted into method-specific supervision. For Gaze2Act, target masks and interaction points are annotated offline in robot-view demonstrations to construct visual prompts and gaze tokens, without first-person recordings or eye-tracking hardware. At inference time, gaze is obtained from Meta Aria glasses (Engel et al., 2023) and grounded online into the robot view. By contrast, language-conditioned baselines use the most specific, unambiguous instruction available in each scene. See Appendix C for task details.

Baselines. We compare against Vanilla GROOT, RoboGround (Huang et al., 2025), and ControlVLA (Li et al., 2025a). For object-level tasks, RoboGround uses its original GLaMM (Rasheed et al., 2024) segmentation pipeline, while ControlVLA uses Grounding DINO (Liu et al., 2024) and SAM2 (Ravi et al., 2025). Their original pipelines are less effective at part-level segmentation, so we use SAM3 (Carion et al., 2025) with explicit part-language prompts for all five part-level tasks, retaining each method's mask-integration mechanism. All methods use the same GROOT N1.5 backbone. Intent Accuracy is evaluated from robot behavior, not segmentation output. Appendix A provides implementation and evaluation details.

4.2 MAIN RESULTS

Table 1 evaluates object- and part-level guidance across 15 static tasks, while Fig. 4 evaluates target updates during execution with an unchanged language instruction.

Object and Compositional Intent. Gaze2Act achieves 93.0% intent accuracy and 89.0% task success across the ten object-level and compositional tasks, compared with 59.6%/42.4% for RoboGround and 68.0%/46.4% for ControlVLA. Gaze supplies the intended spatial referent alongside the task instruction, supporting selection among similar instances and specifying both the manipulated object and its placement target. Success remains 86–88% on unseen objects and 84–86% on transparent objects; the two compositional tasks reach 94% and 84% success.

Part Intent. Gaze2Act achieves 80.4% intent accuracy and 72.4% task success across the five part-level tasks. On cup-body handover and cup-handle pouring, it reaches 88% and 80% success, respectively, compared with 52–68% for the strengthened SAM3-based baselines. These tasks show how gaze can specify an interaction region and, under the shared cup instruction, distinguish manipulation modes. Hammer success is 62–68%, with fine-grained grasping still limited. Appendix A details behavioral scoring and SAM3 neck localization.

Dynamic Intent Steering. The user selects a new target through a voice trigger after execution has begun, while the language instruction remains unchanged. Gaze2Act succeeds in 14/30 trials, compared with 4/30 for RoboGround and 5/30 for ControlVLA, whose language-conditioned mask generators are updated after the switch. Figure 4 illustrates a successful redirection. Despite improved target switching, success remains below 50%, indicating that online redirection remains difficult.

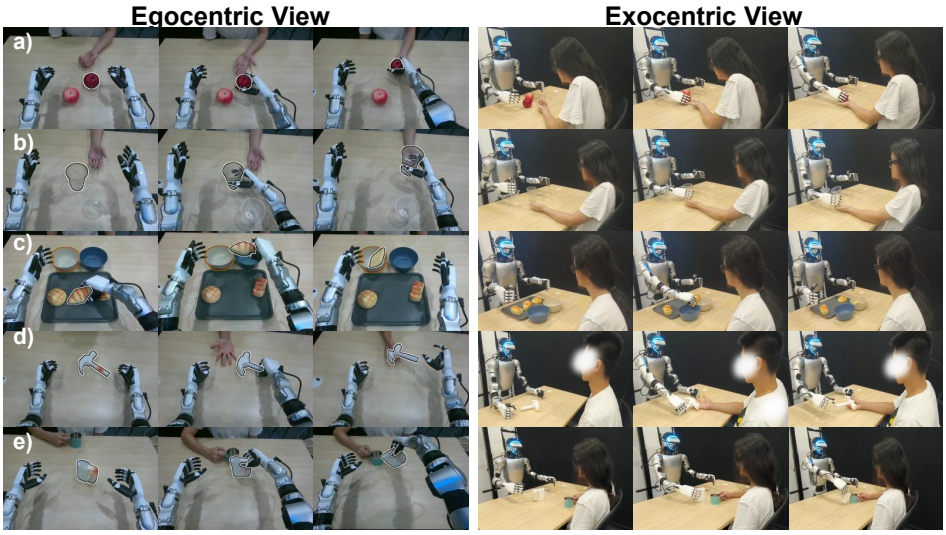


Figure 3: Experimental demonstrations across representative tasks. Each row shows one task instance, with the human egocentric view on the left and the corresponding exocentric robot execution on the right. The rows correspond to (a) Ambiguous Obj., (b) Transparent Obj., (c) Compositional, (d) Subpart Grasp, and (e) Part-cond. Act. The robot follows coarse- and fine-grained grounded gaze cues to complete different tasks.

Table 1: Main results across 15 manipulation tasks, with 50 trials per task (%). Int. = Intent Accuracy: action attempts on the specified object at the specified location. Suc. = Task Success: completion of the instructed manipulation.

		Vanilla GROOT		RoboGround		ControlVLA		Gaze2Act (Ours)	
Category	Task	Int.	Suc.	Int.	Suc.	Int.	Suc.	Int.	Suc.
Ambiguous Obj.	Cup	44	44	86	72	82	62	92	92
	Bread	28	18	36	32	40	32	98	96
	Fruits	30	20	66	56	80	44	100	94
Unseen Obj.	Cup	44	36	84	74	76	62	90	88
	Bread	38	16	44	28	82	54	96	86
	Fruits	48	26	76	40	90	54	94	86
Transparent Obj.	Cup	30	24	56	32	64	42	88	86
	Bottle	20	14	32	24	40	28	88	84
Compositional	Pick bread place bowl	30	26	38	34	42	34	96	94
	Pick paper ball place bin	24	18	78	32	84	52	88	84
Subpart Grasp	Hammer (handle)	40	24	64	34	68	48	80	62
	Hammer (head)	36	18	58	42	66	44	76	64
	Hammer (neck)	42	22	60	42	62	40	70	68
Part-cond. Act.	Cup (handover)	40	22	80	62	84	68	90	88
	Cup (pour)	44	20	76	52	74	60	86	80
Object-level Avg.		33.6	24.2	59.6	42.4	68.0	46.4	**93.0**	**89.0**
Part-level Avg.		40.4	21.2	67.6	46.4	70.8	52.0	**80.4**	**72.4**
Overall Avg.		35.9	23.2	62.3	43.7	68.9	48.3	**88.8**	**83.5**

Cross-view grounding robustness. We evaluate the grounding module independently under viewpoint changes, similar distractors, occlusion, user motion, and gaze perturbations (Appendix B.1). Across 750 trials, grounding achieves 86.9% accuracy and retains 92–96% accuracy at 90° and 180° viewpoint gaps without added distractors or occlusion. The failure analysis links the largest accuracy drops to loss of shared distinctive target evidence under combined clutter and occlusion, with gaze leaving the target providing a separate failure mode.

4.3 Interface Comparison and User Study

We compare Detailed Language, Mouse click, Finger pointing, and Gaze2Act on a two-stage table-organization task, placing a user-specified object into a box at each stage. The task also includes a subset of trials with objects not used in task-specific training. Ten participants each complete five

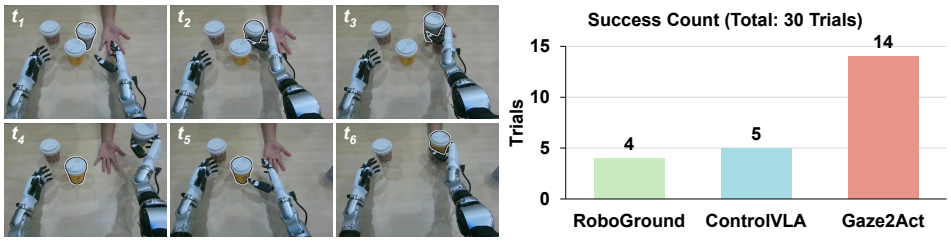


Figure 4: Dynamic intent steering. **Left**: a real-robot rollout (t_1–t_6); the robot first moves toward one cup, the target changes during execution, and the robot redirects the action to hand over the newly intended cup. **Right**: success counts of the three methods over 30 target-switch trials.

Table 2: Interface comparison and user study. Correct Target Selections and Task Progress are computed over 100 indications per interface. The remaining values are participant means ± sample standard deviations. Workload is an adapted NASA-TLX score on a 1–7 scale; lower is better. Naturalness and Control use 1–7 scales; higher is better.

Interface	Target Selections ↑	Task Progress ↑	Interaction Time ↓	Workload ↓	Naturalness ↑	Control ↑
Detailed Language	56%	42%	2.38 ± 0.43 s	4.2 ± 1.0	3.7 ± 1.5	3.6 ± 1.2
Mouse click	**90%**	**74%**	5.11 ± 0.78 s	3.5 ± 1.0	1.7 ± 0.7	5.2 ± 1.4
Finger pointing	60%	46%	2.15 ± 0.35 s	3.7 ± 0.4	5.8 ± 0.7	2.7 ± 1.2
Gaze2Act	86%	72%	$\mathbf{1.66 \pm 0.24}$ s	$\mathbf{2.0 \pm 0.5}$	$\mathbf{6.1 \pm 0.7}$	$\mathbf{5.3 \pm 0.7}$

trials per interface while performing an ongoing manual assembly activity. All interfaces use the same VLA and controller and ultimately produce the same robot-view mask–point representation. Human Interaction Time measures user-side target communication, excluding model processing and robot execution. Appendices B.2 and B.3 detail the protocol, questionnaire, and item-level ratings; Fig. 6 shows the setup.

As shown in Table 2, Gaze2Act improves target selection accuracy over Detailed Language and Finger pointing (86% vs. 56% and 60%), with corresponding gains in task progress. Mouse click achieves the highest selection accuracy and task progress, but requires the longest interaction time. Gaze2Act approaches its performance (86%/72% vs. 90%/74%) while reducing mean interaction time from 5.11 s to 1.66 s. The user study complements these task-level results: gaze receives the lowest mean workload and highest mean naturalness, with a control rating close to Mouse click. Finger pointing also receives a high naturalness rating, but its lower selection accuracy and perceived control suggest that an intuitive gesture alone does not ensure reliable robot guidance. Together, these results support gaze as a low-burden interface that combines accurate target communication with natural interaction. Because all interfaces share the same policy, controller, and robot-view mask–point representation, the observed differences arise from how each interface conveys the target to the robot.

4.4 Ablation Studies

4.4.1 Gaze Input Representation

We compare first-person images with a filtered gaze-point overlay against Gaze2Act on table organization (Appendix B.4). Gaze2Act improves target selection from 78% to 86% and task success from 60% to 72%. Direct input requires sustained fixation, yet small gaze and head movements still shift the prompt after filtering, particularly for small targets. Viewpoint differences further complicate relating this cue to robot actions. Gaze2Act uses first-person gaze only to select or update the target, then grounds and tracks it in the robot view. Segmentation delineates the selected object from surrounding distractors, providing object-centered semantic and spatial guidance in the policy's own observation without requiring continued fixation.

4.4.2 Policy Design

Ablation Setup. We ablate the policy design on Pick Bread Place Bowl and Hammer (handle/head/neck), which together separate the two roles of gaze. Pick Bread Place Bowl tests whether gaze selects and binds the correct object and placement target. Hammer tests whether gaze preserves

Table 3: Policy design ablation. Entries report successful trials out of 60. "Gaze prompting only" uses a contour for Pick Bread Place Bowl and a contour plus gaze heatmap for Hammer.

Variant	Pick Bread Place Bowl	Hammer (handle/head/neck)
Baseline	17/60	15/60
Gaze prompting only	34/60	28/60
Gaze conditioning only (random init)	24/60	17/60
Gaze conditioning only (zero init)	40/60	19/60
Gaze2Act (full)	**55/60**	**39/60**

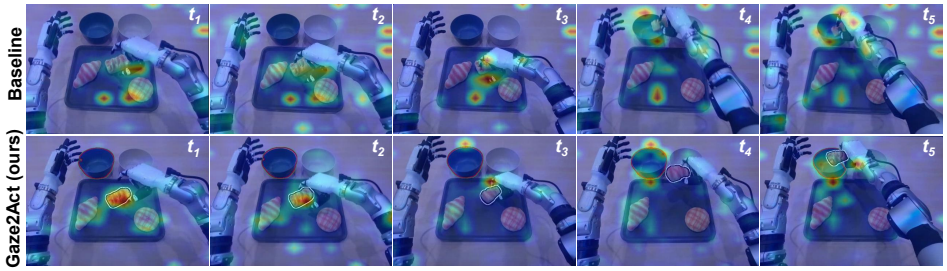


Figure 5: Attention visualization for Pick Bread Place Bowl. Top: Vanilla GROOT with "pick cream bun place blue bowl." Bottom: Gaze2Act with "pick outlined bread place outlined bowl."

the intended contact point. Pick Bread Place Bowl is evaluated over 60 trials per variant. Hammer uses 60 trials per variant in total, with 20 trials for each of the handle, head, and neck.

Complementary Gaze Pathways. Table 3 isolates three design choices: perception-level gaze prompting, action-level gaze conditioning, and stabilized injection. The ablation reveals a clear division of roles. For Hammer, visual prompting is the stronger single pathway, increasing the number of successful trials from 15 to 28, while action-level conditioning alone reaches 19 with zero initialization. For Pick Bread Place Bowl, action-level conditioning is more effective, reaching 40 successful trials compared with 34 from visual prompting alone. The full model performs best on both tasks, with 55 successful trials on Pick Bread Place Bowl and 39 on Hammer, showing that the two pathways provide complementary information. Finally, random initialization weakens action-level conditioning, especially on Pick Bread Place Bowl, where performance drops from 40 to 24 successful trials. These results support stabilized gaze injection into pretrained policies.

Attention Analysis. Fig. 5 shows the attention visualization of a Pick Bread Place Bowl task, where both the bread and the bowl have distractors. Vanilla GROOT receives the specific instruction "pick cream bun, place blue bowl," but its attention remains dispersed across the scene, with the grasp often landing near object edges and becoming less stable. Gaze2Act instead uses the generic instruction "pick outlined bread place outlined bowl" and obtains both referents from gaze. Perception-level gaze prompting marks the selected regions in the observation, while action-level gaze conditioning keeps these spatial cues explicit during denoising. Together, they produce a more concentrated spatial focus and a more stable grasp and place trajectory.

5 Conclusion

We presented Gaze2Act for interactive VLA manipulation through gaze. Our results show that gaze can communicate not only target identity but also intended interaction locations, while allowing users to revise robot behavior during execution. The ablations demonstrate complementary benefits from visual prompting and action-level conditioning, and the interface study shows that accurate target communication can be achieved with low interaction burden. Gaze thus serves as an explicit channel for human guidance rather than solely a source of training supervision. The current system requires eye-tracking glasses, and grounding under occlusion and reliable redirection during execution remain limitations. Overall, these findings support gaze-based human-in-the-loop interaction as a useful setting for guiding VLA policies.

AI USE STATEMENT

Generative AI tools were used only for language editing, including grammar, spelling, and word choice. The authors reviewed all AI-assisted text and take responsibility for the final content of the paper.

ETHICS STATEMENT

The method uses first-person video and gaze measurements, which can contain sensitive information about users and their surroundings. The paper therefore highlights privacy-preserving data handling and physical safety guardrails as requirements for real-world deployment. The reported evaluation uses aggregate task-level outcomes rather than personally identifying results.

REPRODUCIBILITY STATEMENT

The appendix documents the robot and gaze hardware, event-triggered interaction protocol, baseline implementations, prompt construction, training settings, evaluation protocol, task definitions, and demonstration counts used in the experiments. These details complement the method description and are intended to support reproduction of the reported results.

REFERENCES

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023.

J. Randall Flanagan and Roland S. Johansson. Action plans used in action observation. *Nature*, 424 (6950):769–771, 2003.

Sandra G. Hart and Lowell E. Staveland. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati (eds.), *Human Mental Workload*, volume 52 of *Advances in Psychology*, pp. 139–183. North-Holland, 1988.

Haifeng Huang, Xinyi Chen, Yilun Chen, Hao Li, Xiaoshen Han, Zehan Wang, Tai Wang, Jiangmiao Pang, and Zhou Zhao. Roboground: Robotic manipulation with grounded vision-language priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22540–22550, 2025.

Chenxi Jiang, Chuhao Zhou, and Jianfei Yang. REI-bench: Can embodied agents understand vague human instructions in task planning? In *The Fourteenth International Conference on Learning Representations*, 2026.

Roland S. Johansson, Göran Westling, Anders Bäckström, and J. Randall Flanagan. Eye–hand coordination in object manipulation. *Journal of Neuroscience*, 21(17):6917–6932, 2001.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *8th Annual Conference on Robot Learning*, 2024.

Emiel Krahmer and Kees Van Deemter. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218, 2012.

Stephen C. Levinson. *Pragmatics*. Cambridge University Press, 1983.

Chengyang Li, Kaiyi Xiong, Yuan Xu, Lei Qian, Yizhou Wang, and Wentao Zhu. Gazevla: Learning human intention for robotic manipulation. *arXiv preprint arXiv:2604.22615*, 2026.

Puhao Li, Yingying Wu, Ziheng Xi, Wanlin Li, Yuzhe Huang, Zhiyuan Zhang, Yinghan Chen, Jianan Wang, Song-Chun Zhu, Tengyu Liu, and Siyuan Huang. ControlVLA: Few-shot object-centric adaptation for pre-trained vision-language-action models. In *9th Annual Conference on Robot Learning*, 2025a.

Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, et al. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *arXiv preprint arXiv:2501.14818*, 2025b.

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

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Computer Vision – ECCV 2024*, volume 15105 of *Lecture Notes in Computer Science*, pp. 43–63. Springer, 2024.

Kaichun Mo, Leonidas J. Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2Act: From pixels to actions for articulated 3d objects. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 6793–6803, 2021.

Anupam Pani and Yanchao Yang. Gaze-regularized vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2603.23202*, 2026.

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

Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M. Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S. Khan. Glamm: Pixel grounding large multimodal model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13009–13018, June 2024.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. SAM 2: Segment anything in images and videos. In *The Thirteenth International Conference on Learning Representations*, 2025.

Ali Shafti, Pavel Orlov, and A Aldo Faisal. Gaze-based, context-aware robotic system for assisted reaching and grasping. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 863–869. IEEE, 2019.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Tracey Yee Hsin Tay, Xu Yan, Jonathan Ouyang, Daniel Wu, William Jiang, Jonathan Kao, and Yuchen Cui. Intent at a glance: Gaze-guided robotic manipulation via foundation models. *arXiv preprint arXiv:2601.05336*, 2026.

Zixuan Wang, Yuxin Chen, Yuqi Liu, Jinhui Ye, Pengguang Chen, Changsheng Lu, Shu Liu, and Jiaya Jia. Vp-vla: Visual prompting as an interface for vision-language-action models. *arXiv preprint arXiv:2603.22003*, 2026.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Hang Yu, Juntu Zhao, Yufeng Liu, Kaiyu Li, Cheng Ma, Di Zhang, Yingdong Hu, Guang Chen, Junyuan Xie, Junliang Guo, et al. Point what you mean: Visually grounded instruction policy. *arXiv preprint arXiv:2512.18933*, 2025.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *IEEE International Conference on Computer Vision (ICCV)*, pp. 3836–3847, 2023.

A Experimental Details

A.1 Robot Platform and Gaze Interface

All real-robot experiments are conducted on a Unitree G1 humanoid robot. The policy observes the scene through a RealSense D455 RGB camera mounted on the robot. At inference time, the operator wears Meta Aria glasses, and first-person gaze coordinates are obtained with an open-source gaze estimation model. Gaze2Act follows an event-triggered interaction protocol. A voice keyword activates target selection; the current gaze point is used as an interactive point prompt to segment the intended target in the egocentric view, and the selected target is then tracked over time by SAM3. Therefore, the system does not re-run gaze-based target selection at every frame. A new voice keyword triggers target reselection and cross-view matching for Dynamic Intent Steering.

A.2 Baseline Implementations

All four methods use the same GROOT N1.5 backbone and are trained under the same computing environment and training schedule (Section A.4). They share robot demonstration trajectories but use method-specific language conditioning, spatial supervision, and preprocessing.

Vanilla GROOT. The original GROOT N1.5 policy without any explicit spatial conditioning. The Eagle2 VLM consumes the language instruction directly, and the DiT action head receives only the language-vision tokens through standard cross-attention.

RoboGround. We follow Huang et al. (Huang et al., 2025) and use the publicly released GLaMM (Rasheed et al., 2024) checkpoint as the mask generator. The masks are concatenated channel-wise with the observation and processed through a Grounded Perceiver, a token resampler whose attention is guided by the masks, before being passed to the DiT action head. During training, the policy receives ground-truth object masks from data annotation. At the inference stage, GLaMM takes the robot observation and a specific language instruction as input, generates the target masks once at episode onset, and the masks are reused throughout execution.

ControlVLA. We follow Li et al. (Li et al., 2025a) and use Grounding DINO (Liu et al., 2024) and SAM2 (Ravi et al., 2025) as the mask generator. The masks are encoded into object-centric features and injected into the DiT action head via ControlNet-style (Zhang et al., 2023) fine-tuning. During training, the policy receives ground-truth object masks from data annotation. At inference, Grounding DINO and SAM2 take the robot observation and a specific language instruction as input to generate the target masks.

Part-aware baseline segmentation. The original RoboGround and ControlVLA segmentation pipelines are less effective at part-level segmentation. We therefore use SAM3 (Carion et al., 2025) with explicit part-language prompts for all hammer and cup part-level tasks, retaining each method's training setup and mask-integration mechanism.

SAM3 neck-localization limitation. SAM3 can return a mask covering both the hammer head and neck rather than isolating the neck. Intent Accuracy is determined by the robot's attempted interaction location, not by mask coverage.

Behavior-based intent scoring. All methods are scored on the first action attempt at the specified object and location. Vanilla GROOT receives explicit part-language instructions and is scored identically. A correct-region grasp attempt followed by slipping is correct intent but failed task execution.

Backbone unification. The original RoboGround uses a GR-1 policy backbone and the original ControlVLA uses a Diffusion Policy (Chi et al., 2025) backbone. To enable a fair comparison with Gaze2Act, we re-implement both methods on the GROOT N1.5 backbone, preserving each method's mask-integration mechanism and original object-level segmentation pipeline. All training hyperparameters and the computing environment are unified with Gaze2Act.

Language description protocol. For language-conditioned baselines (Vanilla GROOT, RoboGround, and ControlVLA), we provide the most specific unambiguous language description available in each scene, e.g., "handover the red patterned cup with white lid." Gaze2Act uses category-generic templates, e.g., "handover the outlined object," so gaze supplies object identity, contact points, and target updates.

A.3 Gaze Prompt Design

This section explains how the visual prompt form is chosen for different task categories. The choice follows the spatial granularity required by the task. Object-level and compositional tasks require target selection, while subpart and part-conditioned tasks require finer interaction cues.

Object-level prompts. For Ambiguous Obj., Unseen Obj., Transparent Obj., and Compositional tasks, the intent is to select the correct object or placement target among distractors. In these settings, the contour prompt is sufficient because it marks the boundary of the gaze-selected referent while leaving the object appearance visible. We do not add a heatmap because these tasks do not require a within-object interaction location. Adding a local point cue could unnecessarily bias the policy toward an arbitrary region on the selected object.

Part-level prompts. For Subpart Grasp and Part-cond. Act., the intent is not only which object to manipulate, but also where on the object the robot should interact. We therefore combine the contour prompt with a gaze-centered heatmap. The contour provides object-level grounding, and the heatmap provides the fine-grained interaction cue within the selected object. Different fixations on the same object specify different grasp regions or manipulation modes.

Pre-contact fine-grained guidance. During manipulation, fine-grained gaze cues are most useful before contact, when the fixation still indicates the intended interaction region. We therefore render the heatmap only when the grounded fixation remains inside the current object mask, i.e., $p_t \in m_t$. When this condition is not satisfied, Gaze2Act omits the heatmap and keeps the contour prompt. This avoids emphasizing a fixation point that is no longer supported by the visible object region, which can occur when the dexterous hand occludes the target during contact. The gaze token follows the same validity check by using p_t when valid and the mask center otherwise.

A.4 Training and Implementation Details

Training setup. Each task group is trained for 20,000 steps on a single NVIDIA RTX PRO 6000 GPU with 96 GB memory and batch size 80. We use AdamW with $\beta_1 = 0.95$, $\beta_2 = 0.999$, weight decay 10^{-5}, learning rate 1×10^{-4}, cosine decay, 5 % warmup, and bf16 mixed precision. The action head uses flow matching (Lipman et al., 2022) time sampling with $\alpha = 1.5$ and $\beta = 1.0$, action horizon $H = 50$, and 4-step Euler integration at inference. The Eagle2 language model, text encoder, and DINOv3 encoders are frozen. Trainable modules include the vision encoder, DiT action head, mask control encoder, and action encoder/decoder. During training, visual prompts and gaze tokens are generated from masks and interaction points annotated offline in the robot view; at inference, they are generated from first-person gaze through online cross-view grounding.

Lightweight moudule. Compared with vanilla GR00T N1.5, Gaze2Act increases the parameter count from 2.414B to 2.533B, adding only 119.5M parameters (4.95%). Most added parameters belong to the frozen 29M-parameter DINOv3 object encoder and gaze-conditioning layers.

A.5 Evaluation Protocol

Each static task in Table 1 is evaluated over 50 trials, with objects rearranged between trials. Intent Accuracy scores action attempts on the specified object at the specified location. Task Success scores completion of the instructed manipulation. The same behavioral criterion applies to all methods (Appendix A.2). Dynamic Intent Steering uses 30 trials per method. The policy-design ablation uses 60 trials per variant for each task group.

Table 4: Cross-view grounding accuracy under viewpoint gaps, similar distractors, and occlusion. Each cell reports 25 trials.

Scene condition	Same-side	Adjacent-side	Opposite-side
Sparse	100%	96%	96%
Sparse + similar distractors	96%	92%	92%
Sparse + single-view occlusion	100%	92%	92%
Sparse + dual-view occlusion	92%	88%	92%
Sparse + distractors + dual occlusion	88%	80%	80%
Dense	96%	92%	96%
Dense + similar distractors	88%	88%	84%
Dense + single-view occlusion	88%	80%	80%
Dense + dual-view occlusion	84%	72%	72%
Dense + distractors + dual occlusion	80%	64%	68%

Table 5: Cross-view grounding under user motion, posture changes, and gaze perturbations. Each condition contains 25 trials.

Test condition	Grounding accuracy ↑
Seated, stable fixation	96%
Natural head motion	88%
Lateral user shift	92%
Standing posture	80%
$\pm 5^\circ$ gaze pitch/yaw perturbation	72%

B Additional Grounding and Interface Evaluations

B.1 Cross-View Gaze Grounding

We evaluate cross-view gaze grounding independently of the downstream policy under viewpoint changes, visually similar distractors, and partial occlusion. Same-side, adjacent-side, and opposite-side configurations produce approximate view differences of 20–30°, 90°, and 180°, respectively. The tabletop scenes use relatively small everyday objects, including electronics, tools, and food items, to emulate cluttered household workspaces. Sparse scenes contain 5–6 objects, whereas dense scenes contain 10–15 objects. Similar distractors share shape, color, texture, material, or semantic category with the target. Occlusion covers approximately 10–40% of the target in either the egocentric view alone or both views. Five participants, with heights ranging from 1.60 to 1.83 m, complete five trials per condition, yielding 25 trials per table cell. A trial succeeds only when the robot-view match identifies the user-specified target.

Across the 750 trials in Table 4, grounding accuracy is 86.9% (652/750). Viewpoint difference alone has limited impact, with 92–96% accuracy at 90° and 180° without added distractors or occlusion. Accuracy decreases most when dense scenes combine similar distractors with dual-view occlusion, reaching 64% and 68% in the adjacent-side and opposite-side settings.

We further fix an opposite-side sparse scene with similar distractors and vary one user- or gaze-related factor at a time. Each condition contains 25 trials. Table 5 reports an overall accuracy of 85.6% (107/125). Natural head motion and lateral user shifts retain 88% and 92% accuracy. Standing changes the visible object surfaces across views, while a $\pm 5^\circ$ gaze perturbation can move the fixation outside the intended target.

Latency. On one NVIDIA GeForce RTX 5090 GPU, mean trigger-to-grounding latency is 1.14 s with 25–50 candidate masks and 1.47 s with 50–100 masks.

Failure analysis. The largest accuracy drops occur when multiple disturbances remove distinctive target evidence shared by the two views. Dense scenes introduce more competing candidates, while occlusion can hide the regions that distinguish otherwise similar objects. For example, a large viewpoint difference combined with occlusion may leave a patterned cup visible mainly through its white lid, with little of the distinctive pattern exposed. With these identifying features largely absent, the target can be confused with another white, circular object. This explains the pronounced decrease

to 64% and 68% when dense scenes combine similar distractors with dual-view occlusion (Table 4). Posture changes can similarly reduce shared evidence by exposing different object surfaces, with standing accuracy decreasing to 80% (Table 5).

A separate failure arises when the gaze point leaves the intended target. The $\pm 5^\circ$ perturbation can shift the point onto the background or a neighboring object, reducing accuracy to 72%. Unlike the cases above, this failure originates before cross-view matching: an incorrect egocentric selection can produce a reference mask for a neighboring object, so even an accurate cross-view correspondence retrieves the wrong target. Visual confirmation of the selected region before triggering could help detect such selection errors.

Takeaway. The results suggest that shared, distinctive target evidence is more informative about matching reliability than viewpoint angle alone. Matching tolerates large view differences when this evidence remains visible, but degrades when occlusion or differing visible surfaces remove it, or when gaze selects the wrong region.

B.2 Interface Comparison



Figure 6: Interface-study setup for the table-organization task: (a) gaze, (b) finger pointing, and (c) mouse click. Participants direct the robot to place selected objects into a box while performing a manual assembly activity.

We compare detailed language, mouse clicking, finger pointing, and gaze on a two-stage table-organization task. At each stage, the participant specifies an object for the robot to place into a box. Ten participants, aged 23–64, complete five trials per interface, with two target indications per trial, yielding 100 indications per interface. During each approximately one-minute trial, the participant remains engaged in a manual assembly activity while intermittently directing a nearby humanoid in a cluttered seven-object scene.

A subset of trials includes objects absent from the task-specific training data; the systematic unseen-object evaluation is reported separately in Section 4.2 (Unseen Obj.).

Table 6: Interface comparison on the two-stage table-organization task. Interaction time is the participant mean ± sample standard deviation after averaging five trials per participant.

Interface	Correct Target Selections ↑	Task Progress ↑	Interaction Time ↓
Detailed Language	56/100 (56%)	42/100 (42%)	2.38 ± 0.43 s
Mouse click	90/100 (90%)	74/100 (74%)	5.11 ± 0.78 s
Finger pointing	60/100 (60%)	46/100 (46%)	2.15 ± 0.35 s
Gaze2Act	86/100 (86%)	72/100 (72%)	$\mathbf{1.66 \pm 0.24}$ s

All interfaces use the same VLA, controller, visual prompting, and action conditioning, and each interface ultimately produces a robot-view mask and point. Detailed Language supplies SAM3 with an explicit object phrase. Mouse supplies a point prompt in the robot view. A closed-source vision-language model interprets Finger pointing and selects an object for SAM3 segmentation. Gaze transfers the user's fixation across views. Correct Target Selections measures whether the interface identifies the intended object at each stage. Task Progress is the fraction of completed stages.

Human Interaction Time measures approximate user-side target communication time, excluding model processing (including VLM inference for Finger pointing) and robot execution. Participants begin after a start cue, with variability in their responses and interaction timing. Table 6 summarizes target selection, task progress, and interaction time.

Finger pointing requires a VLM to infer the referent from the gesture direction. In cluttered scenes, nearby objects can lie along a similar direction, making this inference ambiguous from the robot view (Fig. 6). This is consistent with its lower selection accuracy despite relatively short interaction time. Both Finger pointing and Mouse click require users to free a hand, interrupting the ongoing manual assembly activity. Mouse additionally requires shifting attention to a screen, locating the target in the robot view, and operating the cursor. Although Mouse achieves the highest target-selection accuracy, these interaction steps accompany the longest communication time. Finger pointing and Mouse also receive higher mean workload scores than Gaze2Act (3.7 and 3.5 vs. 2.0; Appendix B.3). Gaze approaches Mouse in target selection and task progress while avoiding hand-based input and repeated screen-based selection.

B.3 User Study

After completing the trials, participants rate each interface once using an adapted questionnaire based on the six NASA-TLX workload dimensions (Hart & Staveland, 1988). Mental Demand, Physical Demand, Temporal Demand, reversed Performance, Effort, and Frustration are rated on 1–7 scales and averaged into an adapted workload score, where lower is better. Participants also rate Naturalness and Control on 1–7 scales, where higher is better. Table 7 reports the participant mean ± sample standard deviation for every item.

- Mental Demand (MD): "How mentally demanding was intent communication?"
- Physical Demand (PD): "How physically demanding was using the interface?"
- Temporal Demand (TD): "How much time pressure did you feel?"
- Performance (Perf.): "How successful were you in accomplishing the task goals?"
- Effort: "How hard did you have to work?"
- Frustration (Frust.): "How frustrated, stressed, or annoyed did you feel?"
- Naturalness: "Communicating intent through this interface felt natural."
- Control: "I felt in control of the robot's object selections."

The adapted workload scores are 4.2 ± 1.0 for Detailed Language, 3.5 ± 1.0 for Mouse, 3.7 ± 0.4 for Finger pointing, and 2.0 ± 0.5 for Gaze2Act. Gaze2Act yields the lowest workload and highest Naturalness, while its Control rating is comparable to Mouse. We evaluate these ratings in a hands-busy setting motivated by collaboration in which users' hands are occupied and pointing or clicking interrupts their ongoing activity, making low-burden target specification particularly useful.

Table 7: User-study ratings on 1–7 scales. Lower is better for the six workload items; higher is better for Naturalness and Control. Perf. denotes reversed Performance.

Interface	MD ↓	PD ↓	TD ↓	Perf. ↓	Effort ↓	Frust. ↓	Natural. ↑	Control ↑
Detailed Language	5.0 ± 1.2	2.6 ± 2.0	4.9 ± 0.9	3.9 ± 1.1	5.1 ± 1.2	3.6 ± 0.8	3.7 ± 1.5	3.6 ± 1.2
Mouse click	3.8 ± 1.2	3.7 ± 0.9	3.3 ± 1.6	1.9 ± 0.7	4.3 ± 1.6	3.9 ± 1.3	1.7 ± 0.7	5.2 ± 1.4
Finger pointing	2.3 ± 1.2	2.6 ± 1.4	4.4 ± 1.2	5.3 ± 1.5	3.0 ± 1.2	4.5 ± 1.3	5.8 ± 0.7	2.7 ± 1.2
Gaze2Act	$\mathbf{1.8 \pm 1.0}$	$\mathbf{1.4 \pm 0.7}$	$\mathbf{2.3 \pm 0.7}$	2.4 ± 0.5	$\mathbf{2.0 \pm 0.7}$	$\mathbf{1.9 \pm 0.6}$	$\mathbf{6.1 \pm 0.7}$	$\mathbf{5.3 \pm 0.7}$

B.4 Gaze Input Representation

We compare two gaze-input pipelines on the two-stage table-organization task described in Appendix B.2. At each stage, the user selects an object for the robot to place into a box. The direct-input baseline uses first-person images with a filtered gaze-point overlay during training and inference. Gaze2Act instead grounds the selected object in the robot view and uses its segmentation-derived contour and mask–point condition to guide execution.

Table 8: Gaze-input comparison on table organization (%). Int.: correct target selection. Suc.: successful placement of the specified object into the box.

Gaze input	Int.	Suc.
First-person image + filtered gaze point	78	60
Gaze2Act	**86**	**72**

The direct-input baseline requires sustained fixation, and the overlaid point still moves slightly with gaze fluctuations and head motion after filtering. Small target regions are sensitive to these shifts, while the human and robot viewpoints differ. Gaze2Act uses the first-person view for target selection and updates, then tracks the grounded object in the robot view. Segmentation provides the object's extent and distinguishes it from surrounding distractors, rather than representing the target solely by a point. Table 8 compares the input pipelines as a whole. The grounding module is evaluated independently in Appendix B.1, and the policy-side gaze pathways are ablated in Section 4.4.2.

C Task Descriptions

This section details the motivation and setup of the eight task categories in Section 4.1.

(i) Ambiguous Obj. (object-level disambiguation): each scene contains two or three visually similar objects from the same category, such as cups, bread, or fruits. The language instruction does not provide a unique semantic handle for the target, so the task tests whether gaze can specify the intended instance among similar distractors.

(ii) Unseen Obj. (object-level generalization): this setting uses the same training demonstrations as Ambiguous Obj., but replaces the evaluation objects with unseen instances while preserving similar-object distractors. Unseen Cup uses new cup appearances, such as an orange-patterned yellow cup; Unseen Bread uses pineapple buns; and Unseen Fruits uses unseen fruit-like objects such as tomatoes and peaches. The task tests whether gaze grounding generalizes beyond the training appearances.

(iii) Transparent Obj. (object-level perceptual challenge): transparent cups and bottles lack clear visual boundaries in RGB images and are difficult for standard detectors to segment reliably, while the explicit contour of the visual prompt can reconstruct foreground cues in the observation image.

(iv) Compositional (combinatorial spatial specification): both the manipulation object and the placement target must be specified simultaneously, and a single language instruction struggles to encode two independent spatial targets without ambiguity.

(v) Subpart Grasp (part-level localization): the robot grasps the specified hammer part and hands the hammer to the human. Gaze2Act uses the same language instruction across parts, with the highlighted region specifying the grasp location.

Table 9: Training demonstrations and language conditioning for each task group.

Task group	Task variants	Demos	Baseline instruction	Gaze2Act instruction
Ambiguous / Unseen Obj.	Cup	56	handover the blue/pink patterned cup to the human.	handover the outlined object to the human.
Ambiguous / Unseen Obj.	Bread	50	handover the croissant/cream bun to the human.	handover the outlined object to the human.
Ambiguous / Unseen Obj.	Fruits	50	handover the dark red/light red fruit to the human.	handover the outlined object to the human.
Transparent Obj.	Bottle / cup	50	handover the transparent bottle/transparent cup to the human.	handover the outlined object to the human.
Compositional	Pick bread place bowl	89	pick the cream bun/croissant and place it in the blue or white bowl.	pick the outlined bread and place it in the outlined bowl.
Compositional	Pick paper ball place bin	96	pick the white/yellow paper ball and place it in the green/yellow bin.	pick the outlined paper ball and place it in the outlined bin.
Subpart Grasp	Hammer handle / head / neck	105	Hand over the hammer by grasping its handle/head/neck.	Hand over the outlined object by grasping the highlighted part.
Part-cond. Act.	Cup handover / pouring	100	handover the cup. / pour water with the cup.	use the outlined object at the highlighted region.
Dynamic Intent Steering	Target switch	57	handover the specified cup to the human.	handover the outlined object to the human.
Table Organization	Four object types	200	Shared policy with interface-specific target cues (Appendix B.2).	

(vi) Part-cond. Act. (part-level action): interacting with different regions of the same cup corresponds to different manipulation modes, such as handover or pouring; object-level masks cannot encode this part-level intent, requiring the gaze heatmap to provide sub-object resolution.

(vii) Dynamic Intent Steering: the operator looks from object A to object B and uses a new voice trigger to update the selected target during execution. The task tests whether the policy redirects its ongoing action toward the new target while the language instruction remains unchanged.

(viii) Table Organization (interface comparison): the robot places user-specified objects into a box over two stages. This additional task compares interaction interfaces using the same policy and controller (Appendix B.2).

D Training Demonstrations

Table 9 summarizes the robot demonstration trajectories used for training and the language conditioning used by different methods. All methods use the same trajectories within each task group, but the trajectories are converted into different conditioning formats according to each method. Language-conditioned baselines use the most specific instruction available in the scene, while Gaze2Act uses category-level instructions and receives the spatial referent from gaze.Specifically, our data collection is tailored to various task scenarios and distractor configurations to ensure comprehensive coverage. For example, in the bread-picking task group, we collected 25 trajectories for the croissant and 25 for the cream bun, resulting in a total of 50 trajectories for that category.

Ambiguous Obj. and Unseen Obj. share the same training demonstrations. The Unseen Obj. setting differs only at evaluation time, where the test objects are replaced by unseen instances. Unseen Cup uses cups with new appearances, such as an orange-patterned yellow cup. Unseen Bread uses pineapple buns. Unseen Fruits uses unseen fruit-like objects such as tomatoes and peaches. Other evaluation settings follow the task variants in Table 9.

For the table-organization task, we collect 50 demonstration trajectories each for an alcohol container, an iodine container, clear tape, and black tape, totaling 200 trajectories. Each demonstration places the specified object into a box. These trajectories train the shared policy used by all four interfaces.

For Gaze2Act, target masks and interaction points are annotated offline directly in robot-view demonstration videos. These annotations construct the visual prompts and gaze tokens during training, without first-person recordings, eye-tracking hardware, or cross-view matching. At inference time, the same mask–point representation is obtained by grounding first-person gaze in the robot view using the pipeline in Section 3.2.

E BROADER IMPACTS

This work advances intuitive human-robot collaboration by enabling gaze-driven VLA control. On the positive side, this paradigm can significantly lower the barrier to deploying assistive robots, potentially aiding individuals with limited physical mobility by allowing them to guide robots through natural visual attention. However, we also acknowledge potential negative societal impacts. The reliance on continuous egocentric vision and gaze tracking inherently raises privacy concerns regarding sensitive user data collection. Furthermore, in physical environments, misinterpretations of dynamic human intent could lead to unintended or unsafe robot behaviors. Deployment requires privacy-preserving data processing and physical safety guardrails.