VISUAL INSTRUCTION CHANNELS FOR GENERALIST ROBOT POLICIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models offer a capable foundation for robot control, but their usefulness depends on the ability to accurately steer them, that is to specify what a robot should do. Natural language provides a flexible, yet sometimes ambiguous and inaccurate, interface, further limited by the reliability of language-steering. We perform an in-depth analysis of visual instructions as conditioning, individual and complementary to language, supplying them through the policy's existing image inputs without architectural modification. In simulation, we vary the signal's shape (point, box, mask), its delivery format (overlay, blur, blackout, separate image), and its quality (view coverage and signal delay). We find that quality matters most: a signal updated at every observation substantially outperforms the static, first-frame signal often used in prior work. Shape and format effects depend on the backbone and task, with masks and boxes providing reliable overlay choices across the evaluated policies. On three real-robot tasks, visual instructions improve task performance over language alone in every case. These results highlight visual instructions as a practical channel for communicating task intent to generalist robot policies, one that can outperform language on its own and is strongest in combination with it.

1 INTRODUCTION

Vision-Language-Action (VLA) models have emerged as a promising path toward generalist robot policies: a single network that maps camera observations and instructions to complex motor behaviour (Brohan et al., 2023; Kim et al., 2024; Black et al., 2024; Physical Intelligence et al., 2025; Gemini Robotics Team et al., 2025). Their usefulness, however, depends on accurately specifying what a robot should do. Recent evaluations show that VLAs frequently ignore or shortcut the instruction when the scene alone hints at a familiar task (Fei et al., 2025; Fang et al., 2026), that fine-tuning on a narrow task distribution further reduces responsiveness to novel instructions (Huang et al., 2026), and that switching instructions mid-rollout is unreliable even for current models (Chen et al., 2026). Language is also inherently ambiguous when several objects could match a description, and even a precise description gives no guarantee that the policy resolves the reference correctly.

Visual guidance can make task-relevant objects and locations explicit to a robot policy. Since VLAs already consume multiple image streams, we can supply visual signals through the policy's existing inputs without architectural modification. Prior work has conditioned manipulation policies on visual markers of the target (Stone et al., 2023; Huang et al., 2025; Yu et al., 2026; Liang et al., 2026; Li et al., 2025a; Wang et al., 2026), but in each case the marker is bundled with a dedicated architecture. Often it is supplied only once, on the first observation (Yu et al., 2026; Huang et al., 2025; Liang et al., 2026; Stone et al., 2023). We investigate how to present visual instructions to a pretrained policy without modifying its architecture, varying the signal format and quality, and whether the gains can be retained without introducing latency. We show that keeping the signal temporally aligned with the observation is the crucial factor. As Figure 1 illustrates, a signal updated at every observation, rather than fixed from the first one, is a substantial and largely free source of improvement, beyond the choice of signal shape and delivery format. We therefore conduct a systematic study of how to deliver a visual instruction to a pretrained VLA, and confirm the results on real hardware under a blinded, randomised-block protocol.

Our contributions are:

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.

- We show that visual instructions supplied through a pretrained VLA's existing image inputs, with no architectural modification, substantially improve over a language-only baseline trained on the same data and budget. The improvement holds across four backbones, three simulated environments, and three real-robot tasks, extending the case made by prior work with dedicated architectures to the simplest possible integration.
- We systematically compare signal delivery across three shapes and four delivery formats. The best shape depends on the backbone and task, with masks and boxes reliable across the evaluated policies. Delivery format matters more: overlay and background blur tend to outperform formats that remove image content (background blackout) or add a separate image input.
- We show that a signal updated at every observation substantially outperforms the static, first-frame signal often used in prior work, and that this benefit can be retained with no added inference-time latency.
- We validate the central design-space findings on real hardware, across three tasks and two robot embodiments, under a blinded, randomised-block evaluation protocol.

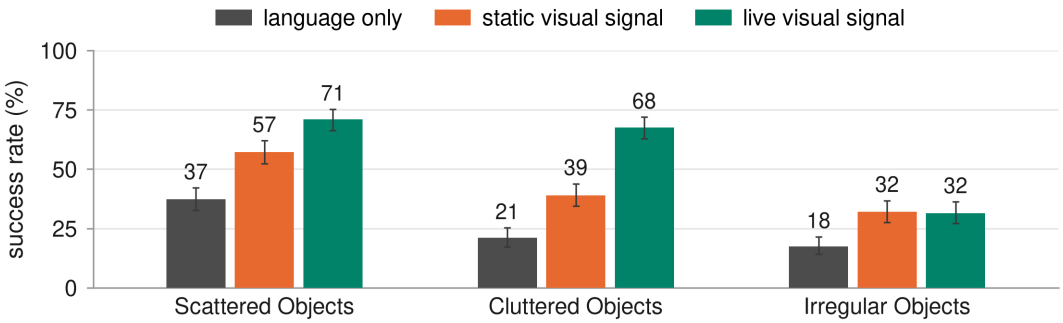


Figure 1: Signal quality as the dominant factor. Success rate with 95% Wilson confidence intervals in three simulated environments for a language-only baseline (grey), a visual instruction computed once on the first observation (orange), and a visual instruction updated at every observation (green). In Irregular Objects, where grasping is the main difficulty, the two visual instructions perform alike.

2 Related Work

Additive visual guidance for manipulation policies. Several works condition a policy on a visual marker of the target. MOO (Stone et al., 2023) appends a single-pixel marker from an open-vocabulary detector to the observation of an RT-1 policy; RoboGround (Huang et al., 2025) feeds target and placement masks from a fine-tuned GLaMM into a GR-1 reproduction trained from scratch, through channel concatenation and a mask-guided perceiver; MaskWAM (Yu et al., 2026) anchors a world-action model on a first-frame SAM 3 mask and trains it to propagate that mask forward; PixelVLA (Liang et al., 2026) adds a prompt encoder, a multiscale pixel-aware encoder, and a new action decoder to the base VLA, trained in two stages on a 160K-episode auto-annotated dataset. In each case, the signal comes with its own integration mechanism, so its properties are never varied independently of the architecture. MOO, RoboGround, and MaskWAM supply the signal once. ControlVLA (Li et al., 2025a) and VP-VLA (Wang et al., 2026) update it during the episode. None varies freshness or view coverage as a factor. We keep the policy architecture unmodified, supply the signal through its existing image inputs, treat format and quality as the experimental variables, and check that the conclusions carry over to real hardware.

Subtractive interventions. BYOVLA (Hancock et al., 2024) and OBEYED-VLA (Vo et al., 2026) also improve VLAs without architectural modification, but by removing content: BYOVLA inpaints task-irrelevant regions at test time, and OBEYED-VLA fine-tunes on observations in which everything outside VLM-selected regions is blanked and the remainder replaced by estimated depth. Both re-segment the selected regions at every step, and both ablate components of their own pipeline (the

sensitivity probe and per-step re-probing in BYOVLA; cross-view matching and depth in OBEYED-VLA), but neither varies the signal itself: the intervention is always removal, the view coverage is fixed, and freshness is never treated as a factor. Removal also leaves the policy with no intact RGB to fall back on; OBEYED-VLA's own failure analysis reports that when occlusion breaks base-view selection, the policy has no recourse even when the wrist view is clean. An additive signal alongside intact RGB avoids this dependency: the policy always retains the full observation, so a segmentation error weakens the hint without corrupting what the policy sees. Our sweep spans both families: blackout and blur are the subtractive end of our delivery axis, overlay is the additive end. We also vary view coverage and temporal freshness.

Other visual instruction channels. Visual instructions can carry content other than a target reference. RT-Trajectory (Gu et al., 2023) and MolmoAct (Lee et al., 2025) condition on a sketched or predicted end-effector path; goal-image conditioning (Myers et al., 2023; Octo Model Team et al., 2024) specifies a desired end state; Interleave-VLA (Fan et al., 2025) places a reference image of the target inside the language instruction rather than in the observation. These channels specify *what* to achieve or *how* to move, and are complementary to the per-observation reference signal we study. Set-of-Mark prompting (Yang et al., 2023), MOKA (Liu et al., 2024), and CrayonRobo (Li et al., 2025b) overlay marks on images to elicit grounding from a VLM planner, whose output is then executed by a separate controller; our signal is consumed directly by the low-level policy at every step.

3 METHODOLOGY

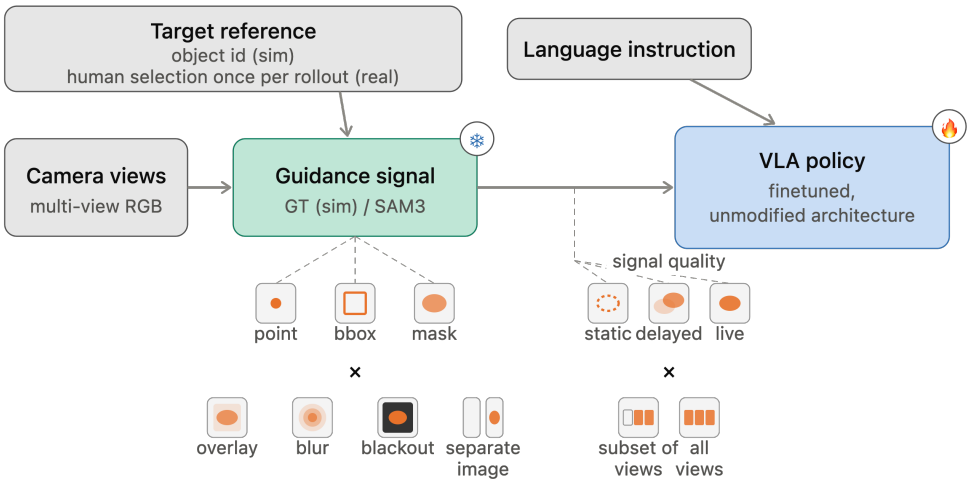


Figure 2: Overview of our study. A visual instruction is produced from the camera views as a guidance signal and passed to the policy through its existing image inputs. The policy architecture is unmodified. In simulation, we control both the format (e.g., point, bbox, mask) of the signal and its quality (e.g., delay). We find that quality is the dominant factor. We then validate the resulting configuration on real robots.

Figure 2 shows the overview of our work. Given camera observations, a guidance signal marking the target is produced, then composited onto the RGB input. The resulting conditioned image is passed to a pretrained VLA policy through its existing image inputs, exactly as an unconditioned RGB frame would be, with no architectural modification, auxiliary loss, or dedicated fusion module. The policy is fine-tuned end-to-end on demonstrations augmented with this signal. Unless stated otherwise, each condition we compare is a separately fine-tuned policy: the signal is applied identically during training and evaluation, so a difference between two conditions reflects what the policy can do when trained and deployed with that signal, not a mismatch between the two.

We organise our study in two stages. We first run an extensive investigation in simulation, where mask quality, coverage, and timing can be controlled precisely, and where the cost of a training

run allows a sweep over the design space. The bulk of our experiments use $\pi_{0.5}$ (Physical Intelligence et al., 2025) as the base policy. To check that our findings are not an artefact of this one backbone, we additionally run a smaller, targeted comparison across π_0 (Black et al., 2024), GR00T N1.7 (NVIDIA et al., 2025; NVIDIA, 2026), and SmolVLA (Shukor et al., 2025). Finally, we take the most important findings to real hardware and design a careful evaluation protocol that preserves as much of the paired structure of simulated evaluation as physical constraints allow.

3.1 Visual Signal Design Space

Prior work has shown that some form of visual guidance can improve manipulation policies. In many works, however, the visual signal was delivered in a simplified way (e.g., static view from the first rollout observation), relying on the architecture to cope with that. We treat the guidance signal itself as a design space and systematically study the importance of this design space. We consider two categories of features: format and quality.

3.1.1 Visual Signal Format

The guidance signal can take various shapes, and each shape can be translated into a visual instruction in various ways. We consider three shapes: a point, a bounding box, and a segmentation mask. We then analyse each shape in terms of the delivery format (skipping obviously bad ones, e.g., point blackout):

- overlaying the signal on the RGB inputs at opacity $\alpha = 0.75$;
- supplying the signal as a separate image alongside the camera views;
- blacking out the background;
- blurring the background;

During format experiments, we assume the guidance signal is available in each camera view and is temporally and spatially consistent with the ground truth (available in simulation).

3.1.2 Visual Signal Quality

Even after determining format, signal quality can affect performance. We therefore analyse this aspect, holding format and shape fixed and varying two dimensions: view coverage and signal delay.

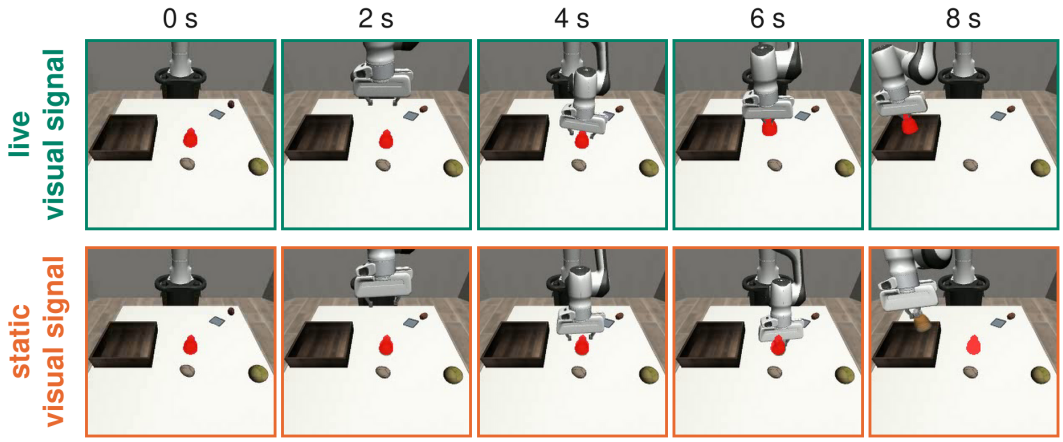


Figure 3: What the policy sees under the two extremes of the signal-delay axis. Top: a live signal, recomputed at every observation. Bottom: a static signal, computed once from the first observation and held fixed while the scene changes. The delayed condition lies between the two.

View coverage. We compare providing the mask in every camera view against providing it only in a subset of views, the two wrist cameras or the side camera (with the remaining views left as plain RGB), to test whether guidance needs to be spatially exhaustive or whether a few well-placed views suffice.

Signal delay. The visual signal can be recomputed at every observation to track the target as the scene changes, or computed once and held fixed (the static case used by prior work). We sweep this axis between these two extremes (Figure 3): a live signal (recomputed at every observation), a static signal (computed once from the first observation and never updated), and signals generated from an observation up to 10 s in the past.

The last point matters in practice. Generating a fresh mask takes time: SAM 3(Carion et al., 2025), for example. needs about 75 ms per image on an RTX 5090 GPU, and a policy can observe several camera views at every step. takes time. A moderate delay can be absorbed by computing the signal ahead of time, without adding inference-time latency. We test whether such a delay still captures most of the benefit of a live signal, or whether performance degrades as delay grows.

4 Experiments and Results

4.1 Environments

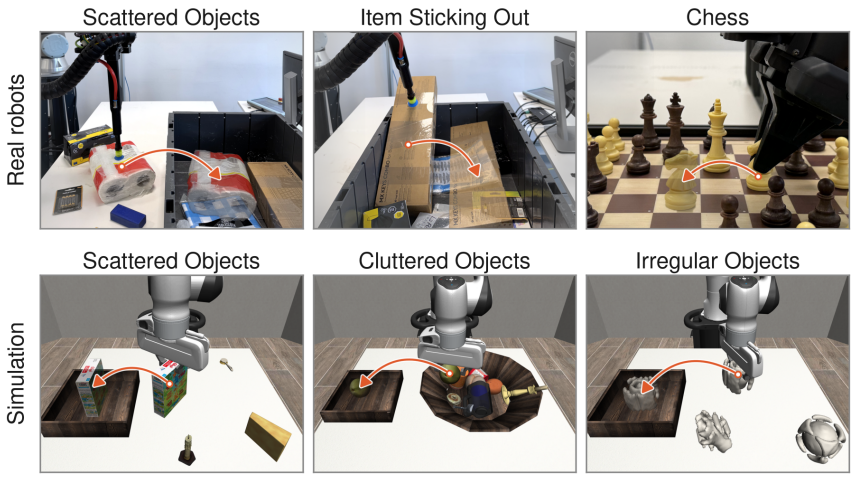


Figure 4: Environments. Top: the three real-world environments, Scattered Objects, Item Sticking Out, and Chess. Bottom: the three simulated environments, Scattered Objects, Cluttered Objects, and Irregular Objects. In all environments except Chess, test objects were unseen during training.

We implement three simulation and three real-world environments. Figure 4 presents the example images of each environment. Except for Chess, the target is always an object unseen during training.

Simulation. All three are pick-and-place tasks in which the goal is to place a target object in a dedicated bin, but they differ in how hard the target is to single out and grasp.

- Scattered Objects. Five items placed without clutter, i.e., far enough apart that the target can be grasped without interacting with other objects.
- Cluttered Objects. The same goal, but a random pile of objects is created in a bowl. The target is always somewhere on top of the pile; we enforce this by dropping the target last, at a random position, once the rest of the pile is in place.
- Irregular Objects. The same goal, but the items are procedurally generated shapes that are hard to grasp and require long textual descriptions to identify.

Real world.

- Scattered Objects. At least 5 items placed on a tabletop, with the goal of placing the target in a dedicated bin. This is the counterpart of the simulated task with the same name.
- Item Sticking Out (ISO). Items are placed in a bin such that one protrudes above its rim, and the goal is to place that item fully inside. The task is inspired by production deployments,

where such edge cases can block a system until resolved. It also serves as a control for the visual signal: by construction, exactly one item protrudes, so the target can be identified from the scene alone, and the instruction carries no referential ambiguity. Any benefit from visual guidance here must come from the subset of instances where the target is hard to perceive, for example when it is not visible in the wrist cameras at the start of the episode, rather than from resolving which object is meant.

- Chess. The goal is to move a chess piece from a source square to a target square. The instruction therefore specifies two locations. Accordingly, the visual instruction marks both the source and the target square, in different colours. In this task, we additionally evaluate visual instructions provided without any textual instruction, to test whether the visual and language signals are complementary.

4.2 Results

In simulation, every result is a policy after 60k training steps, evaluated on 400 episodes, except for the training curves in Figure 7; error bars and shaded regions are 95% Wilson intervals. Table 2 in Appendix A lists the data and evaluation budget of every environment, including the real robots.

4.2.1 Visual Signal Format

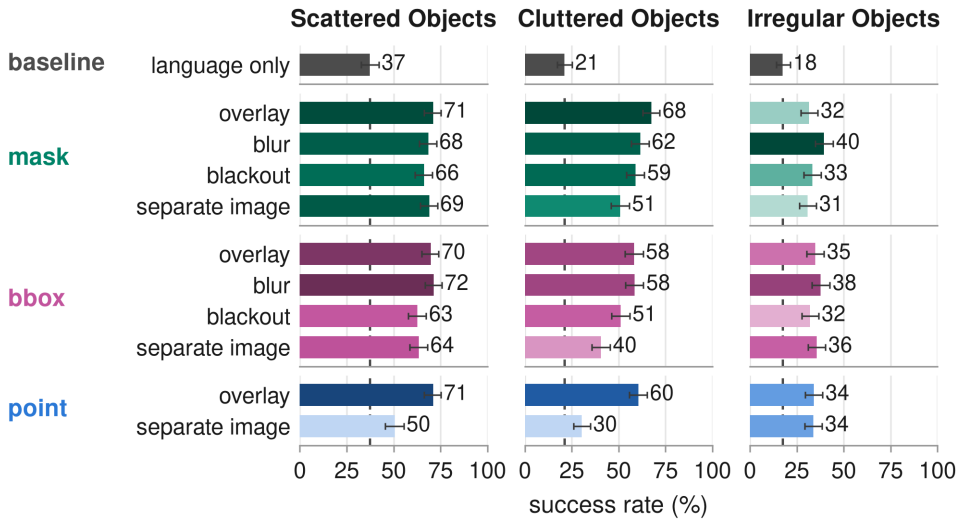


Figure 5: Signal format sweep. Success rate for signal shape and delivery format in each simulated environment. The language-only baseline is shown at the top and marked by the dashed line.

Figure 5 presents the success rate for each combination of signal shape (point, bounding box, mask) and delivery format (overlay, blur, blackout, separate image). In Scattered Objects, shape matters little, as point, bounding box, and mask perform comparably. In Cluttered Objects, the mask leads by at least 8 points. Format matters more, as formats that preserve the policy's pretrained visual distribution (overlaying, blur) tend to outperform formats that alter that distribution more drastically (blackout, separate image). In Irregular Object, where grasping is important, blur tends to outperform other formats. Nevertheless, all conditions substantially outperform the unconditioned baseline, confirming that visual guidance helps a pretrained VLA. Based on this sweep, due to computational reasons, we standardise on mask overlay for all further experiments.

Figure 6 repeats the shape comparison for three further backbones, π_0, GR00T, and SmolVLA, with the delivery format fixed to overlay. Visual guidance improves over the language-only baseline for every policy in every environment, in most cases by a wide margin, so the effect is not specific to $\pi_{0.5}$. The shape comparison, however, is not uniform across policies. For GR00T and SmolVLA, a point is consistently the weakest signal and a mask or box the strongest.

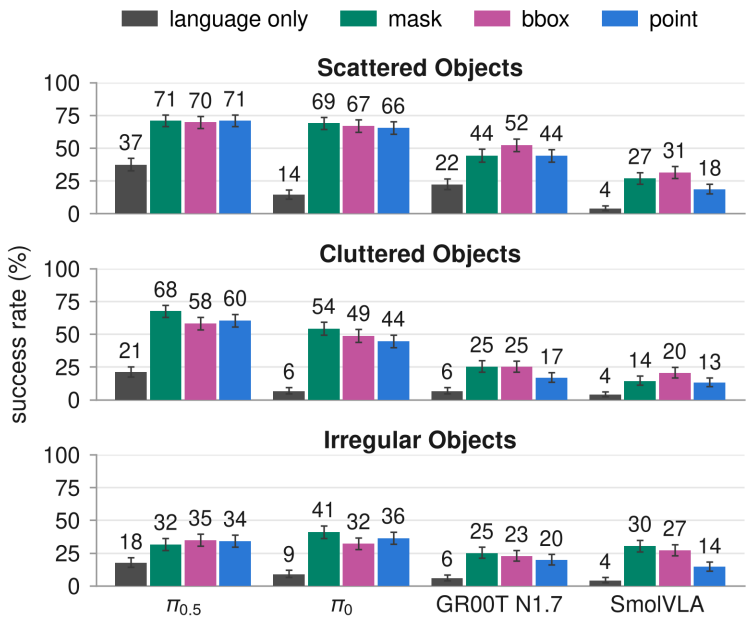


Figure 6: Visual guidance across policies. Success rate across visual signal shapes in each simulated environment for four VLA backbones. The format is held fixed at overlay.

4.2.2 Visual Signal Quality

Section 3.1.2 posed two questions about signal quality: whether guidance must be available in every camera view, and whether it must be temporally up to date. Figure 7 compares providing the mask in every camera view with providing it in a selected subset of views. Providing guidance in the wrist views improves performance over side-view-only guidance on Scattered and Cluttered Objects. Wrist-only guidance approaches all-view performance, while Irregular Objects shows no consistent advantage from broader view coverage. These results indicate that a single well-placed view does not fully capture the benefit of the guidance signal, and suggest that the policy focuses mostly on the wrist cameras' view.

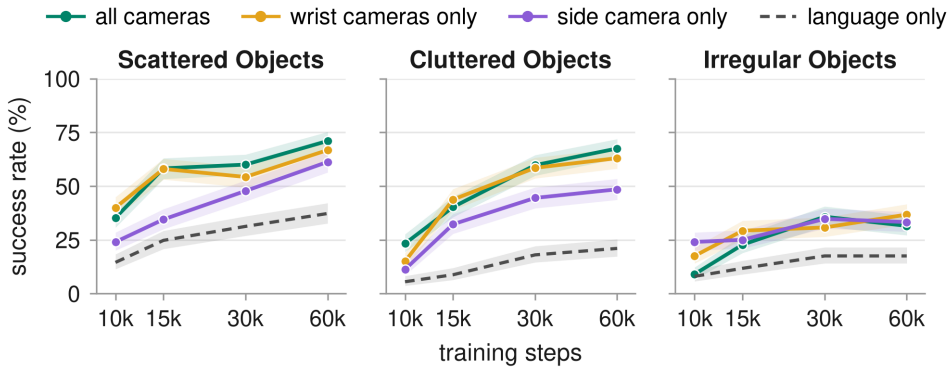


Figure 7: View coverage. Success rate in the three simulated environments when the visual instruction is provided in every camera view versus in selected views, with the remaining views left as plain RGB. Format and delay are held fixed at mask overlay and the live signal. Each point is one checkpoint. Shaded regions are 95% Wilson intervals.

Prior work often used static conditioning from a single observation. Figure 8 shows how delay of the visual signal impacts performance. The static condition is a separately trained policy, consistent with the rule we use throughout the paper. The delayed conditions are not: they evaluate the policy trained with a live signal while feeding it a signal computed from an earlier observation. This

measures how much staleness a live-trained policy tolerates at inference, which is the practically relevant question.

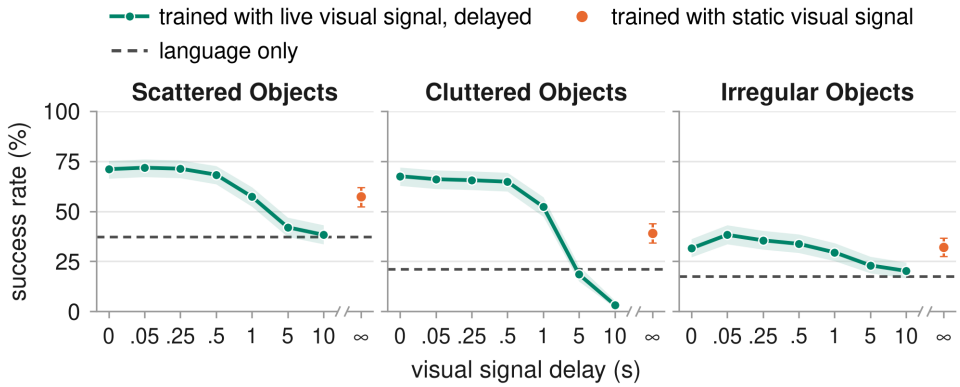


Figure 8: Signal delay. Success rate as a function of the delay, in seconds (non-uniform scale), between the observation used to compute the visual instruction and the observation it is paired with, in the three simulated environments. A delay of zero is the live signal; the static, first-frame signal corresponds to ∞. Delayed points use the policy trained with the live signal, except the static case with a separately trained policy. Format and view coverage are held fixed.

Table 1: Success rate (%) of the live-trained $\pi_{0.5}$ policy at short signal delays, and of the static policy. SAM 3 segments an image in about 75 ms on an RTX 5090, so a mask computed asynchronously for all three cameras lags the image by roughly 0.1-0.25 s, where success stays close to that of the live signal.

Signal delay	Scattered Objects	Cluttered Objects	Irregular Objects
0 s (live)	71	68	32
0.05 s	72	66	38
0.25 s	71	66	36
0.5 s	68	65	34
static (∞)	57	39	32

We see that a static signal causes a major drop in performance. Reasonably delayed masks, however, recover nearly all of the live condition's performance, degrading only as the delay grows large. Since a moderate delay can be absorbed by computing the mask ahead of the observation, the benefit of an up-to-date signal is available without adding inference-time latency. The runtime of SAM 3 falls within this range (Table 1). At the largest delays, the live-trained policy falls below the static policy, even though a static signal is the limit of infinite delay. We attribute this to the training–evaluation mismatch noted above: the static policy was trained on a stale signal and has learned to treat it as a coarse hint, whereas the live-trained policy was never exposed to staleness and is misled when the signal disagrees with the scene. The episode lengths are consistent with this. In every environment, the policy trained with a live signal becomes slower when its signal is delayed: in Cluttered Objects at a 10 s delay, its successful episodes take about 50% longer than without delay. The static policy does not slow down in this way, even though its signal is the most outdated of all. Because it was trained with an outdated signal, it succeeds almost as fast as the live policy does without delay. The comparison that matters in practice is at short delays, where the live-trained policy loses little success and barely any speed, and clearly outperforms the static one.

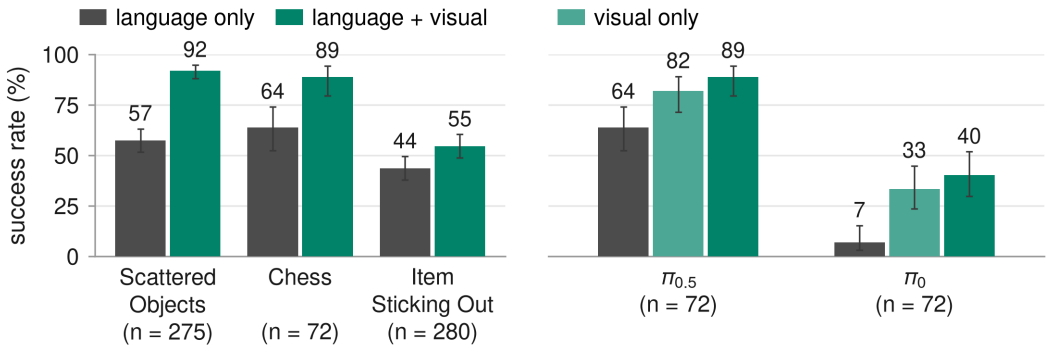
Taken together, the signal quality results indicate that, among the factors we study, the quality often matters most. This is our prime result: much of the benefit visual guidance can offer goes unclaimed by a signal that does not properly track the scene.

4.3 Real world experiments

We evaluate on three real-robot environments described in Section 4.1: Scattered Objects, Item Sticking Out (ISO), and Chess. All real-world policies use the mask-overlay format selected in sim-

ulation, with masks produced online by SAM 3 (Carion et al., 2025), and all three environments are evaluated under the randomised-block protocol: every candidate is evaluated on the same environment instances, in randomised order, with the operator blind to model identity.

Visual guidance improves performance on all three tasks. Figure 9a reports success rate for the policy with visual guidance and the language-only baseline in each environment. Visual guidance outperforms the baseline in every environment, with the largest margin on Scattered Objects and the smallest on ISO. The ordering matches what the task descriptions predict: the gain is largest where the instruction has the most to disambiguate, and smallest on ISO, where the target is identifiable from the scene alone, and the visual signal helps only on the harder-to-perceive instances. We read this as evidence that the benefit of visual instruction scales with how much reference resolution a task demands.



(a) Aggregate results

(b) Chess conditioning

Figure 9: Real-world results. (a) Success rate of $\pi_{0.5}$ with visual guidance and the language-only baseline on the three real-robot tasks; n per model is shown under each task. (b) Chess under three conditions, language only, visual only, and language + visual, for $\pi_{0.5}$ and π_0

Figure 9b shows more detailed results for Chess for both $\pi_{0.5}$ and π_0. Combining the visual signal with language performs best, followed by the visual signal alone, with language alone clearly worst. This ordering holds for both policies. With 72 rollouts per condition, the lead of the combination over the visual signal alone is within the confidence intervals, but the consistent ordering is what we would expect if visual guidance complements language rather than replacing it.

5 Conclusion

We studied visual instructions as a conditioning channel for pretrained VLA policies, supplied through the policy's existing image inputs with no architectural changes. Our central finding is that signal quality is an important, controllable factor for visual-conditioning interfaces. A signal updated at every observation substantially outperforms the static, first-frame signal that much prior work has relied on. A moderately delayed signal retains most of that benefit, so it can be obtained without adding inference-time latency. Providing the signal in wrist camera views also matters. Signal shapes perform similarly across much of the main overlay sweep, but the best choice depends on the backbone and task; masks and boxes are more reliable across the evaluated policies. Delivery formats that preserve the RGB observation tend to work better than those that don't. On real hardware, visual instructions improved success over language alone on all three tasks we evaluated. Visual instructions without language still outperform language alone, but the two channels are strongest in combination.

Limitations and Future Work. For computational reasons, the design-space sweep is conducted on $\pi_{0.5}$, with only a targeted check on other backbones, and the delay and view-coverage experiments use ground-truth masks in simulation. How policies degrade under realistic segmentation errors, and how to train for robustness to them, is the most direct extension of this work.

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing, for retrieval and discovery (e.g., finding related work), for research ideation or execution, and to draft sections of the paper. We have not used generative AI tools to generate synthetic datasets or to prove mathematical claims; they are not applicable to this work.

We have reviewed all AI-assisted work. We did not rely on LLMs to generate research ideas. We always verified everything proposed or suggested by an LLM. AI tools were mostly used to speed up code and figure generation, but everything was always verified by the authors. We take responsibility for the final content of this work, including text, claims or artefacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We will release the code and resources required for reproducibility.

REFERENCES

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.

Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023.

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.

Zhenyang Chen, Alan Tian, Liquan Wang, Benjamin Joffe, Yingyan Celine Lin, Yuxiao Chen, Siddharth Karamcheti, and Danfei Xu. ReSteer: Quantifying and refining the steerability of multitask robot policies. *arXiv preprint arXiv:2603.17300*, 2026.

Cunxin Fan, Xiaosong Jia, Yihang Sun, Yixiao Wang, Jianglan Wei, Ziyang Gong, Xiangyu Zhao, Masayoshi Tomizuka, Xue Yang, Junchi Yan, and Mingyu Ding. Interleave-vla: Enhancing robot manipulation with interleaved image-text instructions, 2025. URL https://arxiv.org/abs/2505.02152.

Yu Fang, Yuchun Feng, Dong Jing, Jiaqi Liu, Yue Yang, Zhenyu Wei, Daniel Szafir, and Mingyu Ding. When vision overrides language: Evaluating and mitigating counterfactual failures in VLAs. *arXiv preprint arXiv:2602.17659*, 2026.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Gemini Robotics Team et al. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *Google DeepMind technical report*, 2025.

Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, Priya Sundaresan, Peng Xu, Hao Su, Karol Hausman, Chelsea Finn, Quan Vuong, and Ted Xiao. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches, 2023. URL https://arxiv.org/abs/2311.01977.

Asher J. Hancock, Allen Z. Ren, and Anirudha Majumdar. Run-time observation interventions make vision-language-action models more visually robust, 2024. URL https://arxiv.org/abs/2410.01971.

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, 2025. URL https://arxiv.org/abs/2504.21530.

Suning Huang, Jiaqi Shao, Ke Wang, Qianzhong Chen, Jiankai Sun, Yanjiang Guo, Mac Schwager, and Jeannette Bohg. Breaking lock-in: Preserving steerability under low-data VLA post-training. *arXiv preprint arXiv:2604.23121*, 2026.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space, 2025. URL https://arxiv.org/abs/2508.07917.

Puhao Li, Yingying Wu, Ziheng Xi, Wanlin Li, Yuzhe Huang, Zhiyuan Zhang, Yinghan Chen, Jianan Wang, Song-Chun Zhu, Tengyu Liu, et al. ControlVLA: Few-shot object-centric adaptation for pre-trained vision-language-action models. In *Conference on Robot Learning (CoRL)*, 2025a. arXiv:2506.16211.

Xiaoqi Li, Lingyun Xu, Mingxu Zhang, Jiaming Liu, Yan Shen, Iaroslav Ponomarenko, Jiahui Xu, Liang Heng, Siyuan Huang, Shanghang Zhang, and Hao Dong. Crayonrobo: Object-centric prompt-driven vision-language-action model for robotic manipulation, 2025b. URL https://arxiv.org/abs/2505.02166.

Wenqi Liang, Gan Sun, Yao He, Jiahua Dong, Suyan Dai, Ivan Laptev, Salman Khan, and Yang Cong. Pixelvla: Advancing pixel-level understanding in vision-language-action model, 2026. URL https://arxiv.org/abs/2511.01571.

Fangchen Liu, Kuan Fang, Pieter Abbeel, and Sergey Levine. Moka: Open-world robotic manipulation through mark-based visual prompting. In *Robotics: Science and Systems (RSS)*, 2024. URL https://arxiv.org/abs/2403.03174.

Vivek Myers, Andre He, Kuan Fang, Homer Walke, Philippe Hansen-Estruch, Ching-An Cheng, Mihai Jalobeanu, Andrey Kolobov, Anca Dragan, and Sergey Levine. Goal representations for instruction following: A semi-supervised language interface to control. In *Conference on Robot Learning (CoRL)*, 2023. URL https://arxiv.org/abs/2307.00117.

NVIDIA. NVIDIA Isaac GR00T N1.7. https://huggingface.co/nvidia/GR00T-N1.7-3B, 2026. Model release; architecture described in NVIDIA et al. (2025).

NVIDIA, Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy, 2024. URL https://arxiv.org/abs/2405.12213.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, et al. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

Austin Stone, Ted Xiao, Yao Lu, Keerthana Gopalakrishnan, Kuang-Huei Lee, Quan Vuong, Paul Wohlhart, Sean Kirmani, Brianna Zitkovich, Fei Xia, Chelsea Finn, and Karol Hausman. Open-world object manipulation using pre-trained vision-language models. In *Conference on Robot Learning (CoRL)*, 2023. URL https://arxiv.org/abs/2303.00905.

Khoa Vo, Taisei Hanyu, Yuki Ikebe, Trong Thang Pham, Nhat Chung, Minh Nhat Vu, Duy Nguyen Ho Minh, Anh Nguyen, Anthony Gunderman, Chase Rainwater, and Ngan Le. Clutter-robust vision-language-action models through object-centric and geometry grounding, 2026. URL https://arxiv.org/abs/2512.22519.

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

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V, 2023. URL https://arxiv.org/abs/2310.11441.

Hanyang Yu, Haitao Lin, Jingbo Zhang, Wenyao Zhang, Chenghao Gu, Heng Li, and Ping Tan. Maskwam: Unifying mask prompting and prediction for world-action models, 2026. URL https://arxiv.org/abs/2606.13515.

A EXPERIMENTAL SETUP

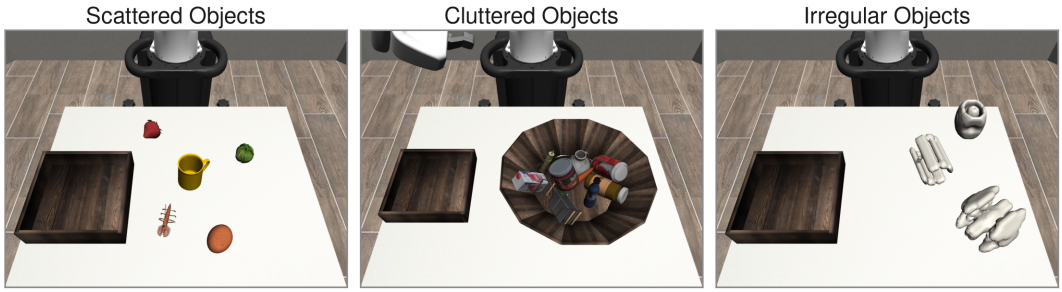
Table 2: Robots, demonstrations and evaluation episodes per environment.

Environment	Robot	Objects per scene	Demonstrations	Evaluation episodes
Simulation				
Scattered Objects	Franka Panda	5	800	400
Cluttered Objects	Franka Panda	10	800	400
Irregular Objects	Franka Panda	3	800	400
Real world				
Scattered Objects	UR10e	≥ 5	744	275
Item Sticking Out	UR10e	≥ 5	949	280
Chess	Trossen WidowX AI	–	1500	72

We use RTX 5090 GPUs for rollouts and H100, A100, or GH200 for fine-tuning.

A.1 SIMULATION

The policy receives the three camera images, the language instruction, and the end-effector and gripper state, and outputs end-effector poses and gripper commands at 20 Hz. An episode succeeds when the target is in the bin, and no other object is. It fails as soon as another object enters the bin, or after 35 s.



"Pick the cup and place it into the box"

"Pick the can and place it into the box"

"Pick the medium-sized squat shape, with 4 lobes standing out around it, its surface gently textured, more open frame than solid body, pierced by a few openings, widest right at the crown and place it into the box"

Figure 10: Simulated environments at the first frame of one evaluation episode each, with the instruction the policy received.

Figure 10 shows one scene from each environment. Scattered and Cluttered Objects use everyday objects from Objaverse, Lightwheel and AI-generated assets. Training uses one object from 40 categories. The evaluation targets are 200 new instances from the training categories and 200 objects from 100 unseen categories. In Irregular Objects, a category is a geometric description of a procedurally generated shape. Instructions follow the template "Pick the *target* and place it into the box", e.g., "Pick the smooth ball clasped by two flat curved arms that arch over it from one side and almost meet, leaving an open gap between each arm and the ball and place it into the box".

A.2 DEMONSTRATIONS AND TRAINING

We collected demonstrations by teleoperation using a VR headset, 20 per training object. We render all visual-signal variants from the same demonstrations using the simulator's ground-truth knowledge.

For each condition, we fine-tune the base pretrained policy checkpoint with a batch size of 16 and a single seed. Policies predict chunks of 30 actions and execute the first 10.

A.3 Evaluation

All conditions and checkpoints are evaluated on the same scenes. We use the same seed, so the starting poses, items present, etc. are identical.

A.4 Visual signal rendering

We draw a bounding box as a 2-pixel outline and a point as a 7-pixel disc. We consider everything outside the chosen shape as background. Overlay paints the shape in red, blackout sets the background to black, and blur applies a Gaussian blur to the background. The separate image adds one image input whose three channels carry signals from the three cameras.

B Real-robot evaluation

B.1 Setup

On Scattered Objects and Item Sticking Out, the UR10e uses a suction cup and is observed by a side camera and two wrist cameras. Both tasks share the same item sets: Figure 11 shows the items used to collect training demonstrations, and Figure 12 shows the evaluation items, none of which appear in the training set. We generate masks with SAM 3 and apply no manual correction. In Chess, the source square is highlighted in green and the target square in red.



Figure 11: Training items of the real-robot Scattered Objects and Item Sticking Out tasks.

B.2 Blinding and resets

Visual guidance is initialised once per instance, before any model in the round runs, and the guidance pipeline runs identically regardless of whether the current model uses masks or not. Rollout inference latency is equal across models. The person supervising the evaluation session therefore does not know whether the current rollout uses visual signal conditioning.

Between rollouts on the same scene, we restore item placements. In the real world, some aspects, such as poses of items the robot interacted with and ambient lighting, cannot be controlled exactly. These residual differences are shared across models in expectation due to order randomisation.



Figure 12: Evaluation items of the real-robot Scattered Objects and Item Sticking Out tasks. None of these items appears in the training set (Figure 11).

B.3 Termination and failure criteria

A rollout is terminated and counts as a failure after a limit of inferences is reached or upon a protective stop of the robot. The budget is 110 policy inferences for Scattered Objects and Item Sticking Out. For Scattered Objects, we also terminate the episode if an incorrect object is placed in the bin, matching the termination criteria of the respective simulation environment. For chess, we similarly end the episode if an incorrect piece is grabbed, the target piece is placed on an incorrect square, or the policy destroys the board state in any way.