ROLEBIND: TRAINING VISION-LANGUAGE-ACTION POLICIES WITH TASK-CONDITIONED ROLE–INSTANCE SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-Language-Action (VLA) models have demonstrated strong capabilities in general-purpose robotic manipulation by jointly modeling visual observations, language instructions, and robotic actions. However, existing VLA models primarily rely on action supervision for policy learning, and there is a lack of explicit constraints on the correspondence between task semantics and physical entities. Robotic manipulation requires not only the identification of relevant entities but also an understanding of their roles in the current task. To address this issue, we propose RoleBind—a training method for implementing the binding of task roles to object instances within existing VLA policies. In addition to the original action learning objective, RoleBind introduces role-instance binding supervision, which explicitly aligns task roles (such as manipulated objects and target objects) with their corresponding visual instances in a shared representation space, thereby enhancing the VLA's representation of task-relevant entities and their role relationships. Role labels and instance region information are used only during the training phase. We applied RoleBind to different action modeling methods on LIBERO and further evaluated its generalization ability in out-of-distribution scenarios. Experimental results show that RoleBind increases the average success rate of StarVLA-OFT from 98.0% to 98.55%, an improvement of 0.55 percentage points, and yields a further performance improvement of 0.8 percentage points on StarVLA-π. These results demonstrate that RoleBind can improve task success rates without altering the original inference process, and also provide a viable approach for explicitly linking task semantics to physical entities in VLA.

1 INTRODUCTION

Vision-Language-Action (VLA) models provide an important paradigm for general-purpose robotic manipulation by jointly modeling vision, language, and robotic actions. Research such as RT-2, OpenVLA, and π_0 has advanced this field in the areas of large-scale vision-language pretraining, open-ended VLA modeling, and continuous action generation, respectively (Zitkovich et al., 2023; Kim et al., 2024; Black et al., 2024). Although these methods possess strong semantic understanding and control capabilities, their robot training still relies primarily on action supervision; consequently, the correspondence between linguistic semantics and specific physical entities in a scene typically needs to be learned implicitly. For robotic manipulation, merely identifying task-relevant objects is insufficient; the strategy should also determine the task roles assigned to different objects based on the current instruction. For example, for the task "Put the bowl on the plate", the bowl is the manipulated object and the plate is the target object; however, in another task, the plate may itself become the manipulated object. Therefore, models should learn not a static word–object correspondence, but a task-dependent association between semantic roles and the corresponding visual object instances.

Existing research on language-conditioned robotic manipulation has established the importance of visual grounding. For example, CLIPort combines the semantic "what" with the spatial "where" (Shridhar et al., 2022), while VIMA uses interleaved visual and language prompts to describe different manipulation tasks (Jiang et al., 2023). Recent work has further explored explicit supervision to improve task-relevant representations in VLAs. GuidedVLA uses auxiliary signals to special-

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.

ize different attention heads in object grounding, spatial geometry, and temporal skill information (Jia et al., 2026). GIFT employs training-time constraints involving geometry, affordances, and goal regions to encourage intermediate features to retain control-relevant structure (Zheng et al., 2026). These studies suggest that final action supervision alone may be insufficient to fully learn the intermediate structure relevant to task execution. We focus on a more specific question: can the correspondence between linguistic task roles and physical instances in the current scene be supervised directly? To this end, we propose RoleBind, a training-time auxiliary supervision method that explicitly associates task-defined semantic roles with their corresponding visual object instances in existing VLA policies. During training, RoleBind uses task-role labels and the corresponding visual instance information to construct binding targets. This encourages task-role representations to align with the correct object instances. As shown in Figure 1, Rolebind takes effect only during training and does not alter the original inference process.

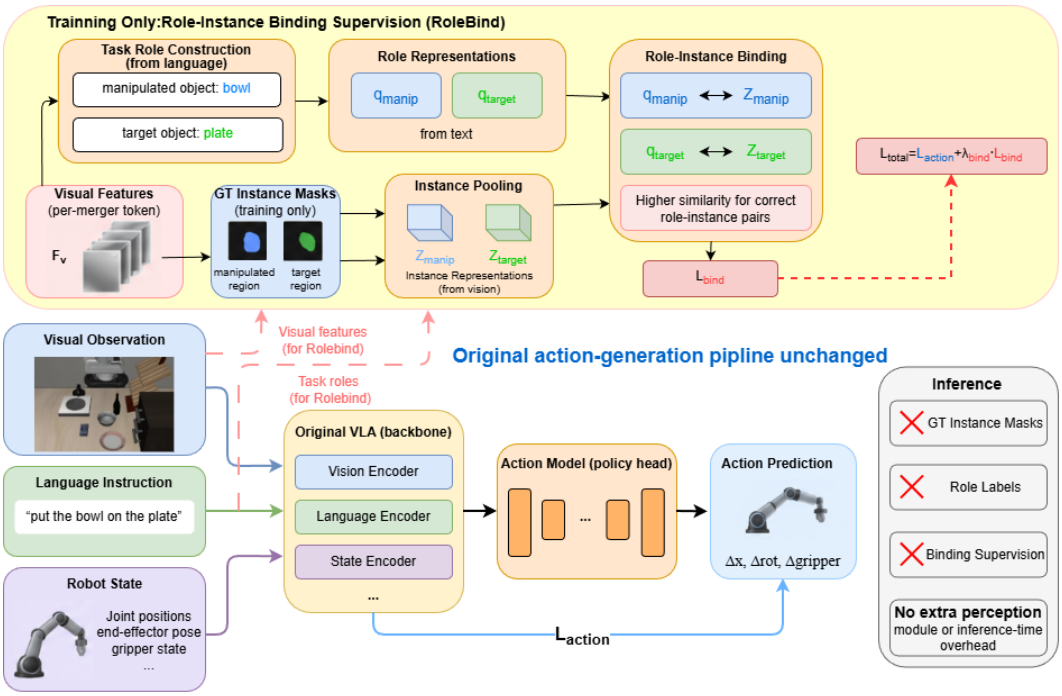


Figure 1: Overview of RoleBind. Role–instance binding supervision is used only during training, while the original action-generation pipeline remains unchanged.

We evaluate RoleBind across different action modeling approaches and robotic manipulation tasks, including standard LIBERO and distribution-shift evaluations on LIBERO-Plus. The main contributions of this paper are as follows:

(1) We focus on the correspondence between task-defined semantic roles and visual object instances in VLA grounding.

(2) We propose RoleBind, an auxiliary training objective that explicitly supervises the association between task roles and their corresponding visual instances alongside the original action objective.

(3) RoleBind requires additional supervision only during training and leaves the original architecture and inference procedure unchanged, enabling its use across different VLA policies.

2 RELATED WORK

2.1 VISION-LANGUAGE-ACTION MODELS AND VISUAL GROUNDING

VLA models extend the semantic capabilities of large-scale vision-language models to robotic control by jointly modeling visual observations, language instructions, and robot states. RT-2 represents

actions as tokens and jointly trains on vision-language and robotic manipulation data (Zitkovich et al., 2023); OpenVLA provides an open, large-scale pretrained VLA and demonstrates strong capabilities in multitask manipulation and language grounding (Kim et al., 2024); and π_0 employs flow matching to generate continuous robot actions (Black et al., 2024). Despite these architectural differences, policy learning is still driven primarily by action supervision from robot demonstrations, while task-specific grounding between language and scene entities is often learned implicitly as part of end-to-end action prediction.

Visual grounding has been central to language-conditioned robotic manipulation. CLIPort combines semantic representations with spatial manipulation capabilities to perform language-conditioned manipulation (Shridhar et al., 2022). VIMA represents tasks and goals through multimodal prompts that interleave visual and textual inputs (Jiang et al., 2023). KITE explicitly grounds language instructions to 2D image keypoints and executes keypoint-conditioned manipulation skills, providing an object-centric interface between language and action (Sundaresan et al., 2023). MOO further leverages pretrained vision-language models to associate open-vocabulary object descriptions with visual observations for open-world manipulation (Stone et al., 2023). Together, these works demonstrate the importance of connecting linguistic semantics to visual entities. RoleBind focuses on a more specific point of this grounding problem: determining which visual instance corresponds to each task-defined semantic role under the current instruction. In this setting, the relevance and role of an object depend not only on its visual identity, but also on the current task.

2.2 EXPLICIT TASK-RELEVANT SUPERVISION IN VLAS

Recent work has increasingly introduced explicit intermediate structure or auxiliary supervision. CoT-VLA predicts future visual goals before generating actions, introducing explicit intermediate visual reasoning into VLA policy learning (Zhao et al., 2025). GuidedVLA directly supervises selected attention heads in the action decoder to specialize in object grounding, spatial geometry, and temporal skill reasoning (Jia et al., 2026). ProGAL-VLA constructs entity-centric representations and introduces a grounding-alignment contrastive objective to associate symbolic subgoals with grounded entities (Darabi & Trivedi, 2026). Wulff et al. similarly introduce explicit language-action alignment for hierarchical VLAs through contrastive supervision between language descriptions and corresponding action trajectories (Wulff et al., 2026). Collectively, these approaches suggest that task-relevant grounding and structure need not emerge solely from end-to-end action supervision.

RoleBind differs from these approaches primarily in the target and granularity of its supervision. CoT-VLA introduces future visual-goal reasoning, GuidedVLA specializes attention heads according to predefined functional factors, ProGAL-VLA combines entity-centric grounding with prospective reasoning, and explicit language-action alignment focuses on consistency between linguistic descriptions and action trajectories. RoleBind instead directly supervises the association between task-defined semantic roles and their corresponding visual instances. This binding objective is jointly optimized with the original action objective during training. The additional role labels, instance masks, and binding branch are removed at inference time, leaving the base VLA's action architecture and inference inputs unchanged.

3 ROLEBIND

3.1 PROBLEM SETUP AND METHOD OVERVIEW

Given a visual observation I_t, proprioceptive state s_t, and natural-language instruction l, a VLA policy π_θ predicts an action or action sequence A_t:

$$A_t = \pi_\theta(I_t, s_t, l). \tag{1}$$

Standard VLA training optimizes an action objective $\mathcal{L}_{\text{action}}$, but does not specify which physical instance should correspond to each task-dependent semantic role in the instruction. RoleBind adds this supervision during training. For each instruction, we consider a set of task roles $\mathcal{R} = \{r_1, \ldots, r_k\}$, where each role r is associated with a corresponding object instance o_r in the current scene. In our experiments, we focus on two common roles: the manipulated object and the target object.

As shown in Figure 1, RoleBind adds a binding branch to the standard VLA training pipeline. This branch aligns task-role representations with their corresponding visual instances. Role labels and instance regions are used only during training to construct the binding objective. The original action-generation pathway remains unchanged, and no additional information is required at inference time.

3.2 TASK-ROLE AND VISUAL INSTANCE REPRESENTATIONS

Task-Role Representations For each LIBERO task, we construct task-level metadata that specifies the semantic names of the manipulated object and the target object. During training, the corresponding role information is retrieved according to the task associated with each episode. We combine the role type with the object semantics to form a short role description. For example, for a task involving a bowl and a plate, the two role descriptions are "manipulated object: bowl" and "target object: plate".

These descriptions are encoded by the language model in the base VLA to obtain the role representations, denoted by q_{manip} and q_{target}. Including the role type distinguishes the function of an object in the current task from its visual category alone. The same object category may therefore be associated with different roles across different tasks. RoleBind uses these task-dependent role representations rather than learning a fixed mapping from object categories to visual regions.

Visual Instance Representations The visual supervision is generated offline from the LIBERO simulator. For each training episode, we replay the corresponding simulator trajectory and obtain instance-level segmentation from the main camera. Using the task metadata, we retain only the binary masks corresponding to the manipulated object and the target object. These masks are used only during training. The RGB observations, multiview inputs, language instructions, robot states, and actions used by the original VLA remain unchanged.

Specifically, we extract visual tokens from the final layer of the visual encoder before spatial merging. In Qwen3-VL, this stage preserves fine-grained spatial information (Bai et al., 2025), allowing the training-time masks to associate object regions with visual tokens. For role r, let P_r denote the set of visual tokens covered by its corresponding mask. Let v_i denote the visual feature of token i. The visual instance representation is obtained by average pooling over the valid tokens:

$$z_r = \frac{1}{|P_r|} \sum_{i \in P_r} v_i. \tag{2}$$

The masks are not prediction targets and RoleBind does not require an additional segmentation model during policy learning. They are used only to identify which existing visual features correspond to the physical instance associated with each task role. If the relevant object is fully occluded and no valid mask region is available in the current frame, that role is excluded from the binding loss to avoid invalid supervision.

3.3 ROLE–INSTANCE BINDING OBJECTIVE

Because language role representations and visual instance representations originate from different feature spaces, RoleBind uses two lightweight projection layers to map them into a shared representation space:

$$\begin{aligned} \hat{q}_r &= \text{Normalize}(W_q(q_r)), \\ \hat{z}_r &= \text{Normalize}(W_z(z_r)). \end{aligned} \tag{3}$$

where W_q and W_z denote the text and visual projectors, respectively. We then use the similarity between the normalized representations to measure the match between task role r and visual instance j:

$$s(r, j) = \hat{q}_r \cdot \hat{z}_j / \tau \tag{4}$$

where τ is a temperature parameter.

Inspired by vision-language contrastive learning (Radford et al., 2021), we treat the correct role–instance pair as a positive match and use other valid, distinct role regions as competing matches. For role r, the binding loss is written as:

$$L_{\text{bind}}(r) = -\log\left[\frac{\exp(s(r,r))}{\sum_j \exp(s(r,j))}\right] \tag{5}$$

The final L_{bind} is the mean binding loss over all valid task roles. This objective directly increases the similarity of correct role–instance pairs while decreasing the relative probability of matching a role to an incorrect visual region.

In some tasks, the manipulated object and target object refer to the same physical region. In these cases, we do not treat them as negative pairs, which avoids conflicting supervision. Only roles with valid visual regions contribute to the loss. Unlike general vision–text alignment, RoleBind does not simply align object names with visual regions. Instead, it aligns each object with the task-specific role it assumes in the current instruction, thereby explicitly modeling task-conditioned role–instance correspondence.

3.4 JOINT TRAINING AND INFERENCE

RoleBind preserves the original VLA action learning objective and adds role binding as auxiliary supervision during training. The final objective is:

$$L_{\text{total}} = L_{\text{action}} + \lambda_{\text{bind}} \times L_{\text{bind}} \tag{6}$$

where L_{action} is exactly the original action loss of the base VLA, allowing RoleBind to be combined with different action modeling approaches. For example, a regression-based action model can retain its original regression objective, while a flow-matching-based action model continues to use its original flow-matching objective. RoleBind simply adds the same role-binding supervision to either objective.

All additional information used by RoleBind, including task-role labels, instance masks, and role–instance correspondences, is available only during training. At inference time, the binding branch and its supervision are removed, and the policy retains the same input–output interface as the base VLA, taking visual observations, language instructions, and robot states as input to predict actions. RoleBind therefore requires no instance segmentation model at deployment, adds no new visual inputs, and does not alter the action-generation process. Its purpose is to improve visual–language correspondence in existing VLAs through more explicit structural supervision during training, rather than to introduce a new inference-time perception module.

4 EXPERIMENTS

Our experiments investigate two central questions: (1) whether RoleBind improves robotic manipulation performance in existing VLAs; (2) whether these gains persist across different action modeling approaches. We further analyze the robustness of RoleBind using a more challenging evaluation with out-of-distribution perturbations.

4.1 EXPERIMENTAL SETUP

Benchmarks We first conduct standard robotic manipulation evaluations on LIBERO. LIBERO comprises diverse language-conditioned robotic manipulation tasks. We use four task suites commonly adopted in current VLA research: LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long (Liu et al., 2023). These suites emphasize spatial relationships, object knowledge, task goals, and long-horizon manipulation, respectively.

For each task, we conduct 50 independent evaluation trials and define the success rate as the proportion of trials in which the task is completed successfully. All compared methods use the same initial task states, evaluation code, and number of trials to reduce the effects of evaluation randomness. We report the success rate for each of the four task suites and their average.

Given that strong VLAs already achieve high success rates on standard LIBERO, we evaluate the models' out-of-distribution robustness on LIBERO-Plus. While preserving the task structure of the original LIBERO, LIBERO-Plus systematically perturbs seven factors: object layouts, camera viewpoints, robot initial states, language instructions, lighting conditions, background textures, and sensor noise. It contains 10,030 test tasks and 21 fine-grained perturbation factors in total (Fei et al., 2025). RoleBind does not use LIBERO-Plus data for additional training. This benchmark is used solely to evaluate the generalization of policies trained on standard LIBERO under distribution shifts.

To better characterize the improvement under the near-saturated success rates on LIBERO, we additionally report the relative reduction in failure rate, defined as

$$\text{Failure Reduction} = \frac{F_{\text{baseline}} - F_{\text{RoleBind}}}{F_{\text{baseline}}}, \tag{7}$$

where $F = 1 - \text{Success Rate}$.

Model instantiations To assess whether RoleBind depends on a particular action-generation architecture, we select two substantially different action modeling approaches within a unified VLA framework. Our experiments use a Qwen3-VL vision-language backbone connected to StarVLA-OFT and StarVLA-π-style action heads through StarVLA's modular interface. StarVLA supports interchangeable action decoding approaches with a shared vision-language backbone and training infrastructure, enabling comparisons between action models with fewer additional variables (StarVLA Community, 2026).

StarVLA-OFT uses parallel continuous action prediction and action chunking, directly predicting future action sequences through L1 regression. Its design is consistent with the efficient continuous action modeling approach of OpenVLA-OFT (Kim et al., 2025). StarVLA-π uses a generative action model based on flow matching to iteratively generate continuous action sequences (Black et al., 2024). The substantial difference between the two action-generation mechanisms allows us to assess whether RoleBind is merely an optimization technique tailored to a specific action head.

For each action model, the Baseline and RoleBind use the same base checkpoint and training setup. The base VLA is trained with the same data, schedule, action horizon, and trainable parameters in both settings. RoleBind only adds two lightweight projection layers for the binding objective.

Table 1: Training Configuration

Model	Steps	Batch	Horizon	Backbone LR	Action LR	Projector LR	λ_{bind}	τ	Bind Dim.
OFT	10k	4	8	10^{-7}	10^{-6}	10^{-4}	10^{-3}	0.07	256
π	10k	4	8	10^{-7}	10^{-6}	10^{-4}	10^{-3}	0.07	256

4.2 MAIN RESULTS ON LIBERO

For each method, checkpoints are selected using the same evaluation protocol. Table 2 presents the main experimental results of RoleBind on standard LIBERO.

Table 2: Success rates (%) on the four LIBERO task suites.

Model	Spatial	Object	Goal	Long	Average
StarVLA-π	99.2	98.6	96.2	96	97.5
StarVLA-Oft	99	99.6	98.6	94.8	98
RoleBind-StarVLA-π	99.2	99.4	98.4	96.2	**98.3**
RoleBind-StarVLA-Oft	98.8	99.6	99.2	96.6	**98.55**

For the StarVLA-OFT action model, the Baseline achieves an average success rate of 98% across the four task suites. After adding RoleBind, the success rate has increased to 98.55%, an absolute improvement of 0.55 percentage points. The largest improvement occurs on LIBERO-Long, where the success rate increases from 94.8% to 96.6%.

For the StarVLA-π action model, the Baseline achieves an average success rate of 97.5%, compared with 98.3% for RoleBind, a change of 0.8 percentage points. Despite the different action modeling mechanisms of StarVLA-OFT and StarVLA-π, the overall positive trend for both indicates that the benefits of role–instance supervision are not restricted to a particular action decoding architecture.

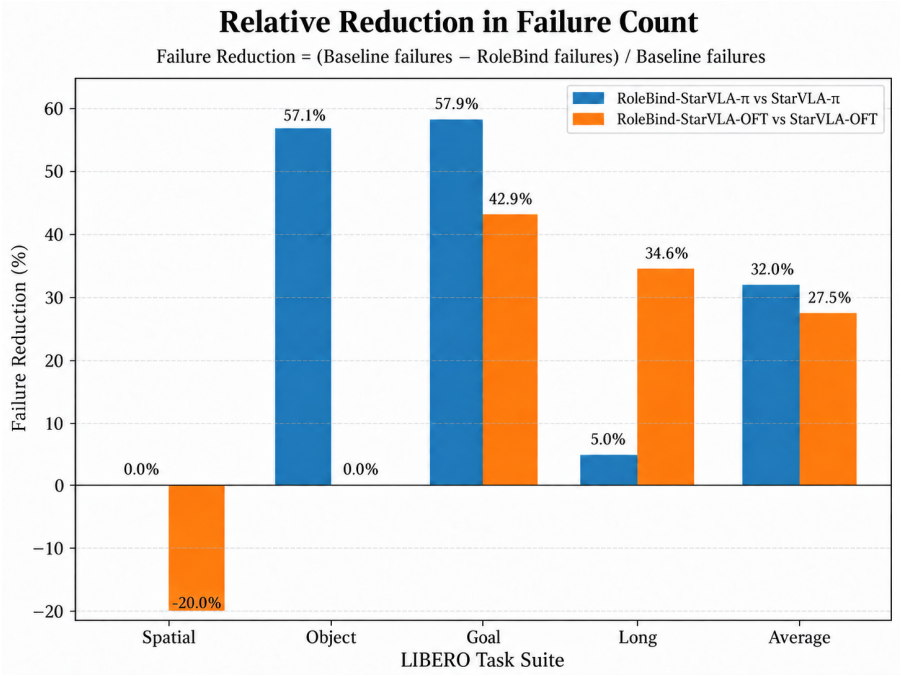


Figure 2: Relative reduction in failure rate of RoleBind over the corresponding StarVLA baselines on the four LIBERO task suites. Positive values indicate fewer failures after applying RoleBind, while negative values indicate increased failures.

As shown in Figure 2, RoleBind reduces the average failure rate by 32.0% for StarVLA-π and 27.5% for StarVLA-OFT. The reduction is particularly pronounced on Object and Goal for StarVLA-π, and on Goal and Long for StarVLA-OFT. The only clear degradation occurs on OFT-Spatial, where the failure rate increases by 20.0%. These results are consistent with the motivation of RoleBind, suggesting that explicit role–instance supervision can benefit different downstream action models.

4.3 GENERALIZATION UNDER DISTRIBUTION SHIFTS

Standard LIBERO primarily evaluates robotic manipulation scenarios that are relatively close to the training distribution. One motivation for RoleBind is to encourage more explicit associations between task semantics and visual instances. We evaluate whether its benefits persist under distribution shifts using LIBERO-Plus.

LIBERO-Plus systematically changes visual and task conditions while keeping task objectives largely unchanged. These changes include Object Layout, Camera, Robot Initial State, Language, Lighting, Background, and Sensor Noise. Object Layout is of particular interest because it changes the positions of target objects and introduces distractor objects. This tests whether the model can associate task-defined roles with the correct visual instances under distribution shifts. Across all LIBERO-Plus test tasks, oft_Baseline achieves an overall success rate of 75%, compared with 75.83% for RoleBind, a change of 0.83 percentage points. For StarVLA-π, the Baseline achieves an overall success rate of 77%, compared with 77.96% for RoleBind, a change of 0.96 percentage points.

The largest gains from RoleBind occur on Robot Initial States and Language Instructions, respectively. The gains are not limited to object-layout perturbations, suggesting that the effect of RoleBind may extend to other distribution shifts. The specific effects differ across models with different action modeling approaches, but the overall trend in success rate is positive.

Table 3: Success rate (%) on LIBERO-Plus

Model	StarVLA-OFT (Baseline)	RoleBind-StarVLA-OFT	StarVLA-π (Baseline)	RoleBind-StarVLA-π
Camera Viewpoints	47	47.53	64.3	63.41
Robot Initial States	60.1	62.13	57.2	57.1
Language Instructions	87	88.16	82.8	85.23
Light Conditions	96.3	97.02	94.2	94.31
Background Textures	95.3	97.21	94	95.54
Sensor Noise	73.1	72.2	79.6	81.64
Objects Layout	79.2	79.93	78.2	78.56
Total	75	**75.83**	77	**77.96**

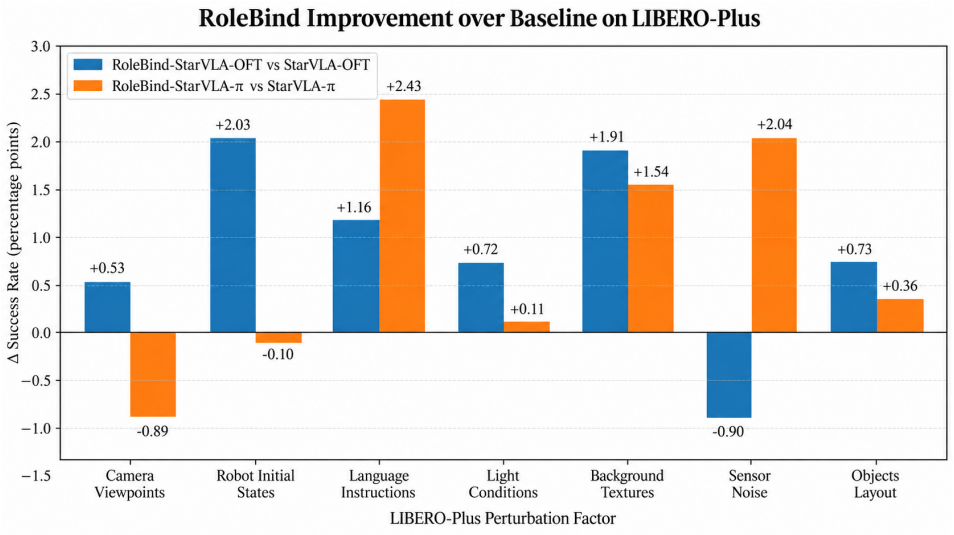


Figure 3: RoleBind improvement over the corresponding baselines on LIBERO-Plus, measured in percentage points.

4.4 ABLATION STUDIES

Finally, we analyze the effect of a key design choice in RoleBind: the binding weight. We vary λ_{bind} to investigate how the strength of binding supervision affects policy learning. Taking StarVLA-Oft as an example, the model reduces to the original Baseline when $\lambda_{\text{bind}} = 0$. Increasing λ_{bind} to 10^{-3} raises the success rate from 98% to 98.55%; further increasing the binding weight to 10^{-2} decreases the success rate to 98.3%. This pattern indicates that role binding is better suited to serving as auxiliary supervision for the original action learning objective than to dominating the training process. The specific results are shown in Table 4.

Table 4: Success rates (%) for Different Values of λ_{bind}

λ_{bind} (RoleBind-StarVLA-OFT)	Spatial	Object	Goal	Long	Average
0	99	99.6	98.6	94.8	98
10^{-2}	98.6	99.6	98.4	96.6	98.3
10^{-3}	98.8	99.6	99.2	96.6	**98.55**
10^{-4}	98.2	99.2	98.2	96.6	98.125

We also examine whether the benefit of RoleBind depends on the correctness of the role–instance correspondence. We construct a swapped variant by exchanging the manipulated-object and target-object role assignments while keeping the corresponding visual instance regions unchanged. This

preserves the same visual supervision while introducing an incorrect role–instance correspondence. The specific results are shown in Table 5.

Table 5: Success rates (%) of different role–instance binding variants on StarVLA-OFT.

Variant	Spatial	Object	Goal	Long	Average
Baseline	99.0	99.6	98.6	94.8	98.00
Swapped role–instance binding	98.4	99.4	98.4	95.2	97.85
Correct role–instance binding	98.8	99.6	99.2	96.6	**98.55**

The swapped variant achieves 97.85%, compared with 98.55% for the correct binding. This gap suggests that the benefit of RoleBind depends on the correctness of the role–instance correspondence, rather than simply on the presence of additional object-region supervision.

5 DISCUSSION AND LIMITATIONS

RoleBind improves the average success rate for both StarVLA-OFT and StarVLA-π action models, but the gains are not uniform across all task suites or distribution shifts. This suggests that task-conditioned role–instance grounding is only one factor affecting VLA performance. In tasks dominated by precise geometry, trajectory generation, grasp pose, or long-horizon control, correctly identifying the manipulated and target objects may not be sufficient for successful execution. RoleBind should therefore be viewed as a complement to action modeling rather than a replacement for it.

The results also indicate that the effect of role–instance supervision depends on the downstream action model and evaluation condition. While both action models benefit overall on LIBERO and LIBERO-Plus, some individual task suites or perturbation factors show little improvement or slight degradation. This suggests that stronger task grounding does not necessarily translate uniformly into better control performance.

RoleBind requires training-time correspondences between task roles and object regions. In LIBERO, these are obtained from simulator-generated instance masks, whereas real-world data may require manual annotation or automatic detection and segmentation. In addition, our current formulation considers only two roles, the manipulated object and the target object, and does not model richer or temporally changing roles. Finally, evaluation is limited to LIBERO and LIBERO-Plus; further validation on real-world robots and additional embodiments is needed.

6 CONCLUSION

We introduced RoleBind, a binding method for vision-language-action policies. By explicitly aligning task roles with their corresponding visual instances, RoleBind provides additional grounding supervision while preserving the original VLA architecture and inference procedure. Experiments on LIBERO show consistent overall improvements across both StarVLA-OFT and StarVLA-π action models, and evaluations on LIBERO-Plus further demonstrate positive overall gains under distribution shifts. These results suggest that role–instance supervision can improve manipulation performance while keeping the original inference pipeline unchanged. The additional grounding supervision is used only during training. We hope this work motivates further exploration of task-conditioned grounding supervision for more complex roles and real-world robotic manipulation.

REFERENCES

S. Bai, Y. Cai, R. Chen, et al. Qwen3-VL technical report. arXiv:2511.21631, 2025.

K. Black, N. Brown, D. Driess, et al. π_0: A vision-language-action flow model for general robot control, 2024.

Nastaran Darabi and Amit Ranjan Trivedi. Progal-vla: Grounded alignment through prospective reasoning in vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9013–9022, 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. URL https://arxiv.org/abs/2510.13626.

Xiaosong Jia, Bowen Yang, Zuhao Ge, Xian Nie, Yuchen Zhou, Cunxin Fan, Yufeng Li, Yilin Chai, Chao Jing, Zijian Liang, Qingwen Bu, Haidong Cao, Chao Wu, Qifeng Li, Zhenjie Yang, Chenhe Zhang, Hongyang Li, Zuxuan Wu, Junchi Yan, and Yu-Gang Jiang. Guidedvla: Specifying task-relevant factors via plug-and-play action attention specialization. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.084.

Y. Jiang, A. Gupta, Z. Zhang, et al. VIMA: Robot manipulation with multimodal prompts. In *International Conference on Machine Learning (ICML)*, 2023.

M. J. Kim, K. Pertsch, S. Karamcheti, et al. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024. Proceedings published in PMLR 270.

Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. In *Robotics: Science and Systems (RSS)*, 2025. URL https://arxiv.org/abs/2502.19645.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 44776–44791. Curran Associates, Inc., 2023. doi: 10.52202/075280-1939. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3c666820ea055a77726d66fc7d447f-Paper-Datasets_and_Benchmarks.pdf.

A. Radford, J. W. Kim, C. Hallacy, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pp. 8748–8763, 2021.

M. Shridhar, L. Manuelli, and D. Fox. CLIPort: What and where pathways for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

StarVLA Community. StarVLA: A lego-like codebase for vision-language-action model developing. *arXiv preprint arXiv:2604.05014*, 2026. URL https://arxiv.org/abs/2604.05014.

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 *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 3397–3417. PMLR, 2023.

Priya Sundaresan, Suneel Belkhale, Dorsa Sadigh, and Jeannette Bohg. Kite: Keypoint-conditioned policies for semantic manipulation. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1006–1021. PMLR, 2023.

Theodor Wulff, Federico Tavella, Rahul Singh Maharjan, Manith Adikari, and Angelo Cangelosi. Grounding hierarchical vision-language-action models through explicit language-action alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9269–9281, 2026.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Tsung-Yi Lin, Gordon Wetzstein, Ming-Yu Liu, and Donglai Xiang. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713, 2025.

Y. Zheng, X. Li, S. Gu, et al. GIFT: Guided intermediate feature training via action-oriented structural supervision for robotic manipulation. arXiv:2609.04193, 2026.

B. Zitkovich, T. Yu, S. Xu, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.