AbsVLA: Learning Robust Primitive Manipulation Skills for VLA Models in Object-Centric Abstracted States

***Abstract*— We investigate the role of representation abstraction in Vision–Language–Action (VLA) policies for robotic manipulation. While recent VLA models show strong performance on multi-task benchmarks, they often exhibit limited robustness under visual and linguistic distribution shifts, especially when trained on limited demonstrations.**

We present AbsVLA, a framework that integrates vision–language grounding with VLA policies to enable manipulation learning in an object-centric abstract state space. Our approach maps language instructions to primitive skills and constructs object-centric observations that suppress appearance variations while preserving task-relevant spatial structure, improving alignment between demonstration and execution distributions.

Experiments on the LIBERO benchmark show that AbsVLA improves robustness under both visual and language perturbations compared to standard VLA baselines, and enables goal-type transfer from object-specified goals to region-specified targets. We further demonstrate zero-shot sim-to-real transfer to a real robot with a different embodiment.

I. INTRODUCTION

Vision–Language–Action (VLA) models have recently emerged as a promising paradigm for general robotic manipulation. By jointly modeling visual observations, language instructions, and action sequences, VLA policies enable robots to perform multiple tasks using a unified architecture conditioned on language prompts. Recent systems such as OpenVLA demonstrate strong performance on multi-task manipulation benchmarks and show encouraging progress toward scalable robot learning [1]–[4].

Despite these advances, we observe that VLA policies often exhibit limited robustness when deployed outside their training distribution. Small variations in visual appearance, background, or language phrasing can lead to significant performance degradation, particularly when training data provides only limited coverage of the underlying state–action space. We observe that one key reason for this limitation is that many VLA policies learn directly from *raw observations*. We further observe representation collapse in raw observation learning: spatially distinct tasks produce nearly identical hidden states (cosine similarity ≈ 0.99), indicating weak spatial grounding(Appendix A: Representation collapse analysis).

Instead of increasing training data, we investigate whether structured state abstraction can serve as an effective inductive bias for VLA policies. We present AbsVLA, a framework that integrates vision–language grounding with VLA policies to enable manipulation learning in an object-centric abstract state space. We leverage grounding models to extract object-centric identities and spatial structure, construct abstract states, and train the policy directly on this representation. This design decouples semantic grounding from action learning and reduces the distribution mismatch between demonstration and execution, leading to improved robustness under visual and language perturbations.

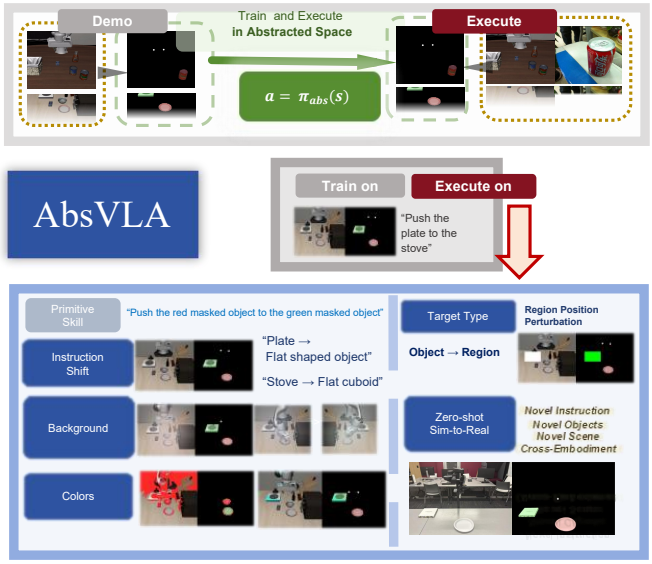


Fig. 1: Overview of AbsVLA. Instead of learning policies directly from raw observations, AbsVLA trains and executes policies in an object-centric abstract state space (top). We evaluate robustness under multiple distribution shifts (bottom-left), including instruction paraphrasing, background changes, and object color perturbations. Finally, we demonstrate additional generalization abilities (bottom-right), including target-type transfer from object goals to region targets and zero-shot sim-to-real transfer to a robot with a different embodiment.

Our main contributions are:

(i) **Object-centric abstraction for VLA policies.** We study learning manipulation policies in an object-centric abstract state space that suppresses appearance-level variations while preserving task-relevant spatial structure. (ii) **A grounding-to-control framework for abstract skill learning.** We present a framework that integrates vision–language grounding with VLA policies, enabling primitive manipulation skills to be learned from object-centric representations. (iii) **Improved robustness and goal-type transfer.** Experiments on the LIBERO benchmark show that AbsVLA improves robustness under visual and language perturbations compared with standard VLA baselines, and enables goal–type transfer from object-specified goals to region-specified

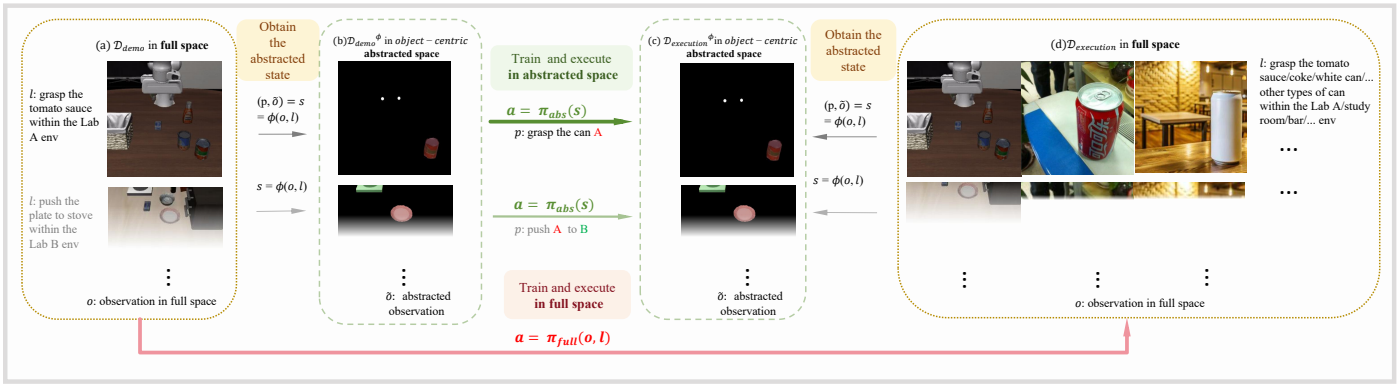


Fig. 2: Distribution Alignment via Object-Centric Abstraction. Raw observations from demonstrations are transformed into object-centric abstracted states using vision–language grounding. Policies are trained in this abstracted space, which suppresses nuisance visual variations while preserving task-relevant spatial structure. During execution, observations from diverse environments are mapped to the same abstract space, aligning the training and deployment distributions and improving robustness. *The font color indicates the mask color assigned to the corresponding object.

targets. We further demonstrate zero-shot transfer from simulation to a real robot with a different embodiment.

II. Problem Formulation and Distribution Alignment

Representation and Distribution Shift. Vision–Language–Action (VLA) policies are typically trained on a limited set of demonstrations, which only cover a small subset of the full execution space. As a result, policies trained directly on raw observations often overfit to appearance-level correlations (e.g., color, texture, background) and fail to generalize under distribution shifts.

In addition, we observe that raw observation learning can lead to weak spatial grounding. For example, tasks with different spatial instructions (e.g., "left" vs. "right") may produce highly similar internal representations(see Appendix A: Representation collapse analysis), making it difficult for the policy to distinguish between them.

Object-Centric Abstraction for Distribution Alignment. To address these issues, we introduce an object-centric abstraction that maps raw observations and language instructions (o, l) into an abstract state:

$$s = (p, \tilde{o}) = \phi(o, l),$$

where p denotes a primitive skill inferred from language and $\tilde{o}$ represents an object-centric observation that removes nuisance appearance variations while preserving task-relevant spatial structure(Fig. 2).

Human-Inspired Decomposition. Our design is inspired by how humans learn manipulation skills. First, humans acquire reusable primitive skills in an abstract, object-centric space, focusing on geometric and spatial relationships rather than memorizing raw visual appearances. Second, humans rely on semantic abstraction to generalize across object instances, grouping visually different objects into shared functional concepts.

Similarly, our formulation separates language understanding, grounding, and control through abstraction, enabling more robust and transferable policy learning.

III. Method

We formulate Vision–Language–Action (VLA) as a structured pipeline operating in an *object-centric abstract state space*. Instead of learning policies directly from raw visual observations, both training and inference are performed on a compact representation $s = (p, \tilde{o})$, where p denotes a primitive skill inferred from language(Appendix A Table II), and $\tilde{o}$ denotes an object-centric abstraction of the scene. This representation suppresses appearance-level variations while preserving task-relevant spatial structure, improving robustness under distribution shifts(Fig 3).

A. Pipeline Overview

Language-to-skill mapping. Given a natural language instruction l, we map it to a predefined primitive skill $p = \psi(l)$ using a lightweight rule-based matcher(Appendix A Table II). This step converts diverse linguistic expressions into a structured and consistent skill representation, reducing sensitivity to instruction paraphrasing.

Grounded object-centric abstraction. Given the raw observation o, we construct an abstracted observation $\tilde{o}$ using a vision–language grounding model. This process extracts task-relevant objects while suppressing irrelevant background information, producing an object-centric representation aligned with the instruction. The abstraction function is defined as

$$s = (p, \tilde{o}) = \phi(o, l).$$

Policy learning in abstract state space. We train a policy π_{abs} to predict actions conditioned on the abstract state:

$$a_t = \pi_{\text{abs}}(s_t).$$

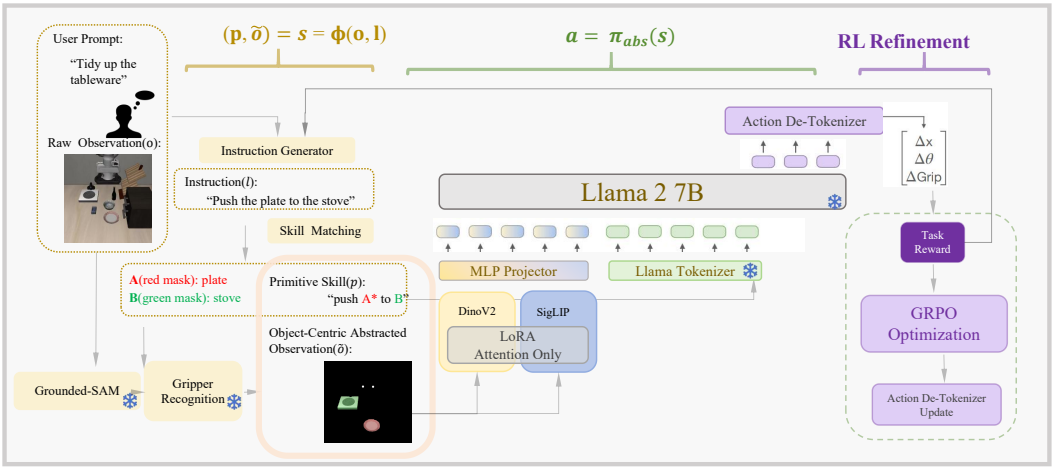


Fig. 3: Overview of the proposed framework for object-centric representation abstraction in VLA-based manipulation. Given a language instruction l and raw observation o, a vision–language grounding module extracts task-relevant objects to construct an abstract state $s = (p, \tilde{o}) = \phi(o, l)$. This representation suppresses appearance variations while preserving task-relevant geometry. A VLA policy based on a LLaMA-2 backbone predicts actions $a = \pi_{\text{abs}}(s)$ from the abstract state, followed by reinforcement learning refinement using task-level rewards.

The policy is optimized via imitation learning, and both training and inference are performed in the same abstract space. This design decouples perception and control, improving robustness to visual and language perturbations.

B. Details and Discussion

Gripper Abstraction via Endpoint Representation. Manipulation primarily depends on the spatial relationship between the gripper and the manipulated object. We therefore represent the gripper using two endpoint markers corresponding to the fingertip positions. See Appendix A (Gripper Abstraction via Endpoint Representation) for details. For different robot platforms, regardless of morphology differences, the same abstraction strategy can be applied by extracting the two fingertip endpoints, enabling consistent ***cross-robot*** representation.

RL-Based Scenario Refinement. We explore RL-based interaction learning for scenario-specific refinement [5]), without fusing into the base model to preserve its generalization. In our current setup, RL is mainly applied to tasks where the initial policy exhibits near-zero success rates. Under such sparse-success conditions, optimization becomes unstable, and we leave improving this component to future work.

Beyond 2D perception: 3D and tactile signals. The current abstraction primarily relies on monocular RGB observations, which may limit geometric reasoning in 3D manipulation scenarios. To address this, additional sensing modalities can be integrated into the abstract state. For example, wrist-mounted depth sensors or LiDAR can provide local geometric structure, while short-range LiDAR or proximity sensing can serve as a proxy for tactile feedback, capturing contact events and surface geometry that are not observable from vision alone. Incorporating such signals into $\tilde{o}$ could improve grounding accuracy and enable more reliable manipulation under occlusion or contact-rich interactions.

Method	Lang shift	Visual shift	Goal transfer
OpenVLA-OFT	0.73	0.41	0.40
OpenVLA-7B	–	0.20	–
AbsVLA	**0.74**	**0.63**	**0.58**

TABLE I: Robustness under language, visual, and goal distribution shifts.

IV. Experiment

Setup. We evaluate AbsVLA on the LIBERO benchmark with primitive skills spanning *open*, *put*, *push*, and *turn on*. All methods are trained on the same dataset and evaluated under distribution shifts, including language paraphrasing, visual perturbations, and goal-type transfer. See Appendix B for implementation details, additional results, and ablations.

A. Main Results

Robustness under distribution shifts. Table IV summarizes performance across language, visual, and goal shifts. AbsVLA consistently outperforms baseline VLA models in all settings.

Performance under visual shifts. We further analyze robustness across visual perturbations for the push and put task. As shown in Figure 5, AbsVLA maintains stable performance across all settings, while baselines degrade significantly.

Key observation. Across all settings, AbsVLA exhibits significantly improved robustness compared to baseline VLA models, particularly under visual perturbations, suggesting that object-centric abstraction reduces reliance on appearance-level correlations.

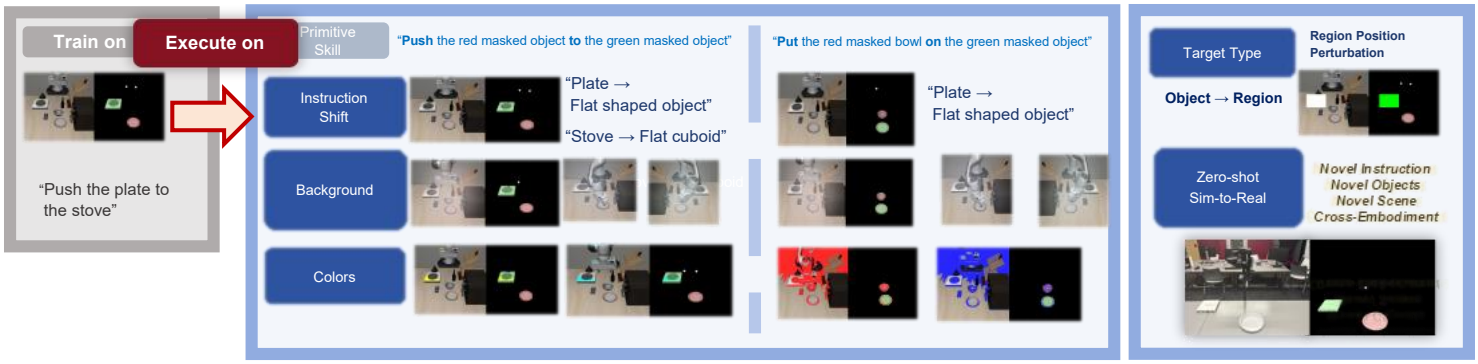


Fig. 4: Evaluation protocol for robustness under distribution shifts. Policies are trained on the original task setting (left) and evaluated under five perturbations at execution time: (i) instruction shifts, (ii) background changes , (iii) color changes affecting both objects and backgrounds while preserving geometry, (iv) target-type shifts from object-centric to region-centric goals with region position perturbations, and (v) zero-shot sim-to-real transfer under novel objects, scenes, and robot embodiments. Two primitive skills (*push* and *put*) are shown as representative examples.

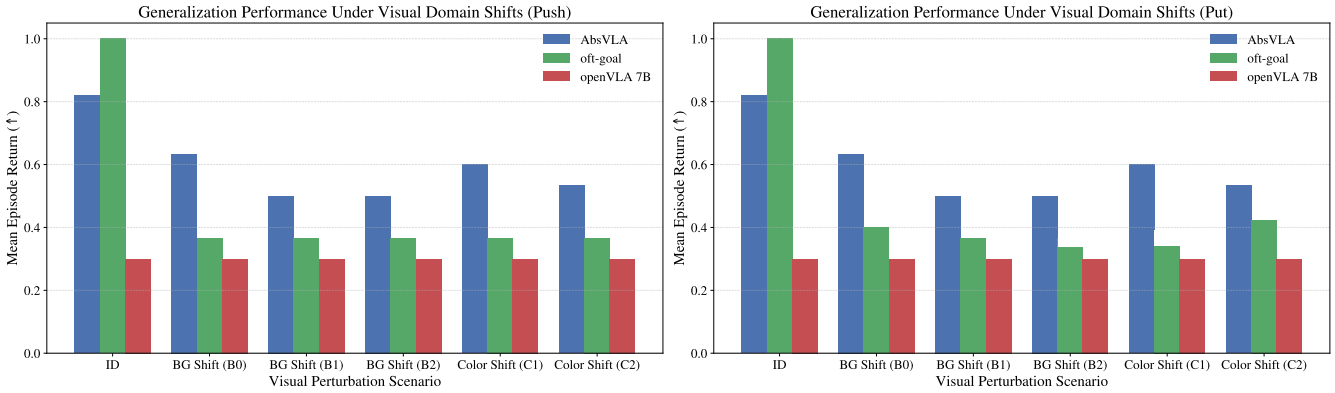


Fig. 5: Generalization Performance Under Visual Domain Shifts for Push and Put Task: AbsVLA maintains an average reward of 0.70 under all background settings, and remains stable under appearance changes.

(a) Observation (b) Execution (c) Robot

Fig. 6: Zero-shot sim-to-real transfer to a different robot embodiment.

B. Zero-shot Sim-to-Real Transfer

AbsVLA successfully transfers to a real robot with a different embodiment without fine-tuning, achieving more stable behavior than baseline methods.

V. CONCLUSIONS

We study the role of representation abstraction in Vision–Language–Action policies for robotic manipulation. We show that learning directly from raw observations introduces nuisance variability that reduces robustness under distribution shifts. We propose learning policies in an object-centric abstract state space constructed via vision–language grounding. Experiments on LIBERO show improved robustness to visual and language perturbations and enable zero-shot cross-embodiment transfer.

Limitation. Grounding may become unstable under partial occlusion, which future work could address with temporal reasoning across frames.

APPENDIX

A. Additional Training Details

Primitive skill set. Meanwhile, the language instruction l is mapped to a primitive skill p from a predefined skill set (Table II). The language-to-skill mapping $\psi(l)$ is implemented using a rule-based matcher over predefined skill templates. Each instruction is mapped to the closest primitive skill pattern.

Training details. We fine-tune OpenVLA-7B using parameter-efficient adaptation with LoRA applied to attention layers of the vision backbone, while keeping the base model frozen. Training is performed on approximately 200k step-level samples with a batch size of 1. We evaluate intermediate checkpoints to avoid overfitting to appearance-level correlations and better assess robustness under distribution shifts.

Representation collapse analysis. We analyze the internal representations of a pretrained VLA policy on two spatially distinct tasks (e.g., "pick up the book in the middle" vs. "pick up the book on the right"). Despite the difference in spatial instructions, the resulting hidden states are highly similar, indicating weak spatial grounding. Table III reports similarity metrics between the two representations, showing near-identical embeddings across multiple measures.

Gripper Abstraction via Endpoint Representation. Manipulation primarily depends on the spatial relationship between the gripper and the manipulated object. We therefore represent the gripper using two endpoint markers corresponding to the fingertip positions, as illustrated in Fig. 7.

Since Grounded-SAM shows limited sensitivity to gripper detection, we additionally train a dedicated detector for the gripper. Specifically, we manually annotate approximately 120 images of the robot gripper used in the LIBERO benchmark (Franka Panda robot) and train a lightweight detector for robust gripper localization. For different robot platforms, regardless of morphology differences, the same abstraction strategy can be applied by extracting the two fingertip endpoints, enabling consistent ***cross-robot*** representation.

B. Additional Experiments and Implementation Details

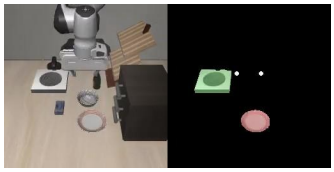


Fig. 7: Object and Gripper Abstraction in the Abstract State Space

Setup: We define a set of 7 primitive manipulation skills spanning four categories: *open*, *put*, *push*, and *turn on*. These skills are mapped to 20 corresponding tasks in the LIBERO simulation environment. In our experiments, we train and evaluate our method on a subset of 10 tasks covering 6 primitive skills (approximately 200k step-level samples), while the remaining tasks are used only for comparison with prior methods. For a fair comparison, all models (including OpenVLA-7B, OFT, and AbsVLA) were trained using the same dataset splits, action tokenization, and input configurations, with the exception of the input modality. AbsVLA and OpenVLA-7B were evaluated using only front-view inputs, while OFT was evaluated with both front and wrist observations to maintain consistency with the original configuration of OFT-GOAL.

TABLE II: Primitive skill set and instruction mapping for the 20 tasks used in this work. Multiple primitive skills are defined within the *put* class because placing different object types requires distinct manipulation actions.

Class	Primitive Skill	Instruction (#)
open	open drawer A	open center/top/bottom drawer; put in drawer (#4)
put A on B	put thin-rect A on B	pick up book (L/M/R) (#3)
	put bowl A on B	put bowl on left/right flat; plate; rightmost square (away) (#5)
	put can/bottle A on B	put can on left/right basket; bottle on square (away/close); avoid intf. (#5)
	put small-box A in B	put cream-cheese in bowl (#1)
push	push A to B	push right flat to leftmost flat object(#1)
turn on	turn on A	turn on stove (#1)

TABLE III: Representation similarity between two spatially distinct tasks on LIBERO-10. Despite different spatial instructions, the resulting hidden representations are nearly identical, indicating representation collapse.

Metric	**Value**
Cosine Similarity	0.9902
Pearson Correlation	0.9902
Normalized L2 Distance	0.07
Token-wise Cosine (Mean)	0.9902
Token-wise Cosine (Min)	0.9659
Token-wise Cosine (Max)	0.9955

Parameter-Efficient Fine-Tuning. We fine-tune OpenVLA-7B [1] using attention-only LoRA. LoRA modules are applied only to the attention layers of the vision backbone, encouraging the model to focus on the highlighted object-centric regions in the abstracted observations, while the base model remains frozen. With LoRA rank 8, the number of trainable parameters is approximately 2.6M out of 733M total parameters (0.36%). Training is performed on a single NVIDIA A10G GPU (24GB VRAM) with batch size 1. We evaluate a checkpoint obtained after approximately 12.5k training steps, corresponding to an average training loss of about 0.07. Unlike typical OpenVLA-style training that continues until very low loss values (e.g., around 0.02), we intentionally evaluate earlier checkpoints to reduce overfitting to visual appearance in the demonstrations and to better study robustness under distribution shifts.

Robustness and Grounding of Abstracted Observations Generation. We study robustness under two types of distribu-

Method	Lang shift	Visual shift	Goal transfer
OpenVLA-OFT	0.73	0.41	0.40
OpenVLA-7B	–	0.20	–
AbsVLA	**0.74**	**0.63**	**0.58**

TABLE IV: Robustness under language shifts, visual perturbations, and goal transfer.

Setting	AbsVLA	oft-goal	openVLA 7B
origin	0.70 ± 0.00	1.00 ± 0.00	0.30 ± 0.00
bg-0	0.70 ± 0.00	0.45 ± 0.09	0.20 ± 0.14
bg-1	0.65 ± 0.09	0.43 ± 0.09	0.10 ± 0.14
bg-2	0.70 ± 0.00	0.37 ± 0.09	0.10 ± 0.14
color-1	0.70 ± 0.00	0.43 ± 0.09	0.30 ± 0.00
color-2	0.78 ± 0.13	0.37 ± 0.09	0.30 ± 0.00

TABLE V: Return (Mean ± Std) Under Visual DomaEpisode in Shifts Across Scenarios for Push Task

tion shift that commonly occur in deployment: (i) language paraphrasing and (ii) visual/background changes. As shown in Fig. 8, Grounded-SAM achieves reliable object grounding and segmentation in static, non-occluded scenes. The resulting object-centric representation suppresses appearance-level variations while preserving spatial geometry, allowing the policy to focus on learning manipulation skills rather than visual correlations.

1) Robustness Under Distribution Shifts: We first summarize the overall robustness across different distribution shifts in Table IV. Table IV reports the average performance of different methods under language shifts, visual perturbations, and goal transfer. AbsVLA achieves consistently higher returns across all settings, indicating improved robustness compared to OpenVLA-OFT and OpenVLA-7B. We then provide detailed results for individual perturbation settings in the following tables.

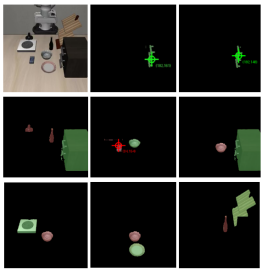


Fig. 8: Grounded-SAM provides accurate grounding for task-relevant objects. From raw observations and language instructions, it produces stable segmentation masks that enable the construction of object-centric abstracted observations.

Evaluation Under Visual and Language Shifts. Raw and object-centric abstracted observations are shown in Fig. 4. The abstraction suppresses irrelevant visual details while highlighting task-relevant objects.

We evaluate generalization under *scenario-level* visual and language distribution shifts, including (i) background

Setting	AbsVLA	oft-goal	openVLA 7B
origin	0.82 ± 0.15	1.00 ± 0.00	0.30 ± 0.00
bg-0	0.63 ± 0.09	0.37 ± 0.12	0.30 ± 0.00
bg-1	0.50 ± 0.16	0.40 ± 0.12	0.30 ± 0.00
bg-2	0.50 ± 0.16	0.35 ± 0.10	0.30 ± 0.00
color-1	0.60 ± 0.29	0.35 ± 0.10	0.30 ± 0.00
color-2	0.53 ± 0.33	0.43 ± 0.12	0.30 ± 0.00

TABLE VI: Episode Return (Mean ± Std) Under Visual Domain Shifts Across Scenarios for Put Task

shifts, where each task is tested under background glow perturbations with warm, cool, and greenish color tints, and (ii) color shifts, which jointly change both object and background colors(shown in Fig. 4). Alongside the visual domain shifts, we apply a language shift by paraphrasing the original LIBERO instructions with more generic object references. Specifically, we evaluate two tasks: push, *push the flat-shaped object on the right to the front of the stove"*, and put, *put the bowl on the flat-shaped object on the right"*. This is obtained by replacing *plate"* in the training instruction with *flat-shaped object on the right"*. We use *origin* (or *id*) to indicate the in-distribution training condition, where both the language instruction and visual observations match those used during training (training instructions + training images). We compare ABSVLA, OFT-GOAL [6], and OPENVLA-7B [1].

We introduce a multi-stage reward to enable fine-grained analysis of generalization. For the push task, the reward is assigned by progress milestones: *move-to* $= 0.3$, *touch* $= 0.5$, *push* $= 0.7$, and *push-to-target* $= 1.0$. For the put task, we similarly define: *move-to* $= 0.3$, *pick-up* $= 0.5$, *edge* $= 0.7$, and *on-target* $= 1.0$. The final reward stage corresponds to successful task completion. The episode return is defined as the maximum stage reward achieved during the episode.

As shown in Table V, VI and Figure 5, although performance in the original setting is not the highest, ABSVLA achieves the strongest overall robustness on the push task: it maintains an average reward of 0.70 under all background settings (bg-0/bg-1/bg-2) with 0.00 standard deviation, and remains stable under appearance changes (color-1: 0.70 ± 0.00; color-2: 0.78 ± 0.13).

For the *put* task, under the first color-shift setting we observe **stronger recovery behavior**(shown as the *Object Color* setting in Fig. 4) After an initially imperfect approach or placement attempt, the policy more frequently re-corrects its motion within the same episode and succeeds in reaching the final *on-target* stage.

This suggests that color perturbations do not harm object grounding for AbsVLA and may even improve target localization by increasing visual contrast and facilitate more precise target localization, leading to improved performance. The model is able to accurately localize the bowl position in these failure cases, enabling recovery behaviors in subsequent attempts.

Overall, these results indicate that object-centric abstraction yields more consistent performance under large ap-

	S1	S2	S3
Push Task			
AbsVLA	0.70	0.65	0.70
oft	1.00	1.00	0.367
Put Task			
AbsVLA	0.82	0.76	0.82
oft	1.00	1.00	0.50

TABLE VII: Instruction shift evaluation. For the *push* task: S1: stove→flat cuboid, S2: plate→plate on the right, S3: plate→flat-shaped object on the right. For the *put* task: S1: bowl→bowl next to the wine bottle, S2: bowl→gray bowl, S3: plate→flat-shaped object on the right.

pearance and instruction variations, supporting improved generalization beyond the training distribution.

Then we mainly evaluation generalization under instruction shifts for push and put task. We isolate text-only distribution shifts by rephrasing the language instruction while keeping the underlying task unchanged, and also keeping the visual observations identical to training. As shown in Table VII, we find that OpenVLA (OFT) can be highly resilient to some text perturbations, achieving near-perfect success rates under certain paraphrases. However, its performance drops markedly under other language shifts, suggesting sensitivity to instruction phrasing and object grounding cues. AbsVLA remains more stable across the same language shifts, exhibiting smaller performance variance and avoiding severe degradation, because paraphrased instructions map to the same primitive skill representation, resulting in identical skill-conditioned inputs to the policy.

Evaluation under Goal-type transfer: (Object-goal → Region-goal). We further evaluate goal-type transfer by switching the goal specification from an *object-centric* target to a *region-centric* target. Specifically, the policy is trained with instructions of the form *"push to the object"*, but is evaluated on region-based instructions such as *"push to the white rectangular area"*, where the goal is specified as a spatial region rather than a named object(shown as Figure 4). Notably, the white square target region is *jittered* across episodes, indicating robustness to target-position variations. For the task *"push the plate to the white rectangular area"*, the episode return is 0.58 ± 0.11 (mean $\pm$ std, $n = 5$). This constitutes a task-semantics shift (object-goal → region-goal) beyond mere paraphrasing.

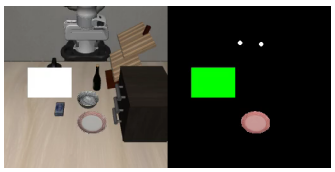
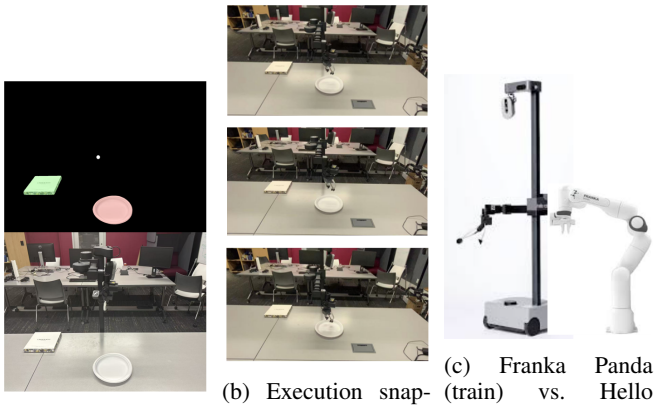


Fig. 9: Task: *push the plate to the white rectangular area*. We consider a goal-specification shift by switching from an *object-centric* target to a *region-centric* target.



(a) Real observation and abstract state. (b) Execution snapshots in the real environment. (c) Franka Panda (train) vs. Hello Robot Stretch (deploy).

Fig. 10: Zero-shot sim-to-real deployment in novel settings.

TABLE VIII: Zero-shot cross-embodiment execution results over five trials.

Method	Episode Return (mean $\pm$ std)
OpenVLA-OFT	0.18 ± 0.16
AbsVLA	0.42 ± 0.11

2) Zero-shot Sim-to-Real Execution: We further explore sim-to-real generalization by deploying the learned policy in a real-world setting with unseen visual inputs and a different robot morphology. The policy is trained in LIBERO using the Franka Panda arm and deployed on a Hello Robot Stretch platform. This transfer is enabled by our gripper abstraction, where the end-effector is represented as two fingertip points in the abstract state space.

Setup: We randomly select a tabletop location and construct a novel scene with a white plate and a chocolate box—objects that are unseen during training (which uses a red-rimmed plate and a stove) but share similar geometric shapes with the training instances. The policy receives RGB observations from the real environment (i.e., all raw observations are out-of-distribution relative to training). We issue a previously unseen instruction, *"push the plate to the chocolate box"*, which does not appear in the training set, although the underlying primitive skill template (*push A to B*) has been seen during training. To interface with Stretch, we apply only a coordinate frame transformation; no additional fine-tuning is performed. Due to hardware interface constraints, only the first action chunk of the 8-action sequence was executed.

Execution Results. Across five trials, the mean episode return is 0.42. During execution, the gripper always maintains an approximately constant clearance above the tabletop, reflecting imperfect height control under embodiment differences. Nevertheless, the overall motion pattern remains consistent: the end-effector approaches the target, adjusts orientation, maintains contact, and completes the pushing trajectory. These observations suggest that the abstract repre-

Task	Setting	AbsVLA	w/o gripper	Δ
push	color-1	0.70 ± 0.00	0.567 ± 0.115	-0.133
push	color-2	0.78 ± 0.13	0.567 ± 0.115	-0.213
put	color-1	0.60 ± 0.29	0.800 ± 0.173	$+0.200$
put	color-2	0.53 ± 0.33	0.700 ± 0.000	$+0.170$

TABLE IX: Gripper ablation results. Δ denotes the performance change after removing the gripper representation at inference time.

sentation captures transferable manipulation structure across embodiments.

C. Gripper Abstraction Ablation

We investigate the importance of gripper abstraction by training the model without it for 1700 steps. During training, we observe that the model repeatedly samples identical action chunks on certain tasks, suggesting that gripper abstraction is important for representing relative spatial relationships between the gripper and manipulated objects.

Inference-Time Gripper Removal. We further conduct a second ablation where the model is trained with gripper input but the gripper representation is removed at inference time. This causes a clear performance drop on the *push* task, In contrast, the *put* task remains relatively stable. We hypothesize that *push* relies on explicit gripper localization to determine the next motion, whereas in *put* the bowl pose implicitly cues the gripper configuration for placement.

REFERENCES

[1] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. P. Foster, P. R. Sanketi, Q. Vuong *et al.*, "Openvla: An open-source vision-language-action model," in *8th Annual Conference on Robot Learning*.

[2] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023, pp. 2165–2183.

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[4] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[5] H. Li, Y. Zuo, J. Yu, Y. Zhang, Z. Yang, K. Zhang, X. Zhu, Y. Zhang, T. Chen, G. Cui *et al.*, "Simplevla-rl: Scaling vla training via reinforcement learning," *arXiv preprint arXiv:2509.09674*, 2025.

[6] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," 2025. [Online]. Available: https://arxiv.org/abs/2502.19645