

Enhancing Generalization in Vision–Language–Action Models by Preserving Pretrained Representations

Shresth Grover^{1*}, Akshay Gopalkrishnan^{1*}, Bo Ai¹, Henrik I. Christensen¹, Hao Su^{1,2}, and Xuanlin Li²

¹UC San Diego ²Hillbot *Equal contribution

<https://gen-vla.github.io>

Abstract— Vision–language–action (VLA) models finetuned from vision–language models (VLMs) hold the promise of leveraging rich pretrained representations to build generalist robots across diverse tasks and environments. However, direct fine-tuning on robot data often disrupts these representations and limits generalization. We present a framework that better preserves pretrained features while adapting them for robot manipulation. Our approach introduces three components: (i) a dual-encoder design with one frozen vision encoder to retain pretrained features and another trainable for task adaptation, (ii) a string-based action tokenizer that casts continuous actions into character sequences aligned with the model’s pretraining domain, and (iii) a co-training strategy that combines robot demonstrations with vision–language datasets emphasizing spatial reasoning and affordances. Evaluations in simulation and on real robot show that our method improves robustness to visual perturbations, generalization to novel instructions and environments, and overall task success compared to baselines.

I. INTRODUCTION

Building general-purpose robots that generalize across tasks and environments is a long-standing goal in robotics. While foundation models in vision and language generalize well from internet-scale data [1], the lack of large-scale, action-labeled robot data makes comparable generalization in robotic manipulation challenging [2]. Efforts on vision–language–action (VLA) models seek to improve robotic generalization by leveraging the rich representations of pretrained vision–language models (VLMs) and finetuning them on robot data (e.g., [3], [4]). This allows us to leverage advances in VLMs for robotic tasks, presenting a promising pathway toward manipulation policies that generalize across environments, tasks, and embodiments.

However, directly finetuning pretrained VLMs on robot data leads to significant representation degradation. Our preliminary experiments show that, with an existing training recipe [3], [4], background changes and small instruction paraphrases can cause large drops in performance (Figure 1), suggesting that finetuning disrupts the structures of pretrained visual and language representations. Recent work has proposed co-training on both vision–language and robotic data [5], [6] to mitigate this issue. However, we find that naive co-training on both objectives does not lead to the best performance, as the two sources of data differ substantially in structure, limiting the ability for the robot action prediction process to reuse vision–language representations. These findings point to an important question: what is the training recipe that preserves the powerful, general representations of pretrained VLMs to facilitate generalization of downstream VLAs?

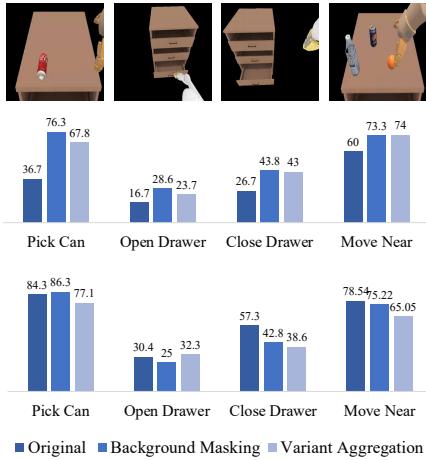
To address this challenge, we explore three design choices. First, we propose mixing frozen and finetuned visual encoders to preserve pretrained VLM representations while keeping high model flexibility to adapt well to robotic tasks. Second, we propose a language-aligned action tokenizer that casts numerical robot actions into character sequences, thereby enabling maximal reuse of pretrained language representations and allowing actions to be refined step by step during generation. Finally, leveraging this unified string-based output space, we co-train the model on both robot and vision–language datasets that emphasize spatial affordances and reasoning. The training recipe is general and can be applied to different existing VLA model architectures.

We evaluate these designs in both simulation and real-world settings and find consistent improvements over baseline VLAs. Our approach yields stronger generalization to out-of-distribution visuals, such as backgrounds, table textures, and distractor objects, as well as to varied language commands, leading to higher task success rates. These results provide practical insights into how pretrained VLMs can be effectively grounded in robotic action, bringing us closer to reliable generalist robot manipulation policies.

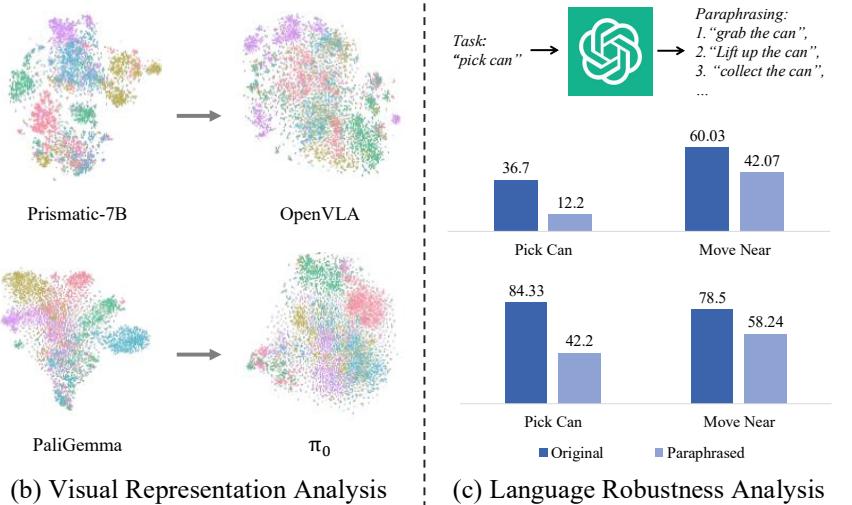
II. RELATED WORK

Vision–Language Models (VLMs). VLMs learn unified representations across vision and language modalities through large-scale pretraining on web-scale image-text data. Recent VLMs such as PaliGemma [7], Qwen2.5-VL [8], InternVL3 [9], and many others [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20] combine strong language models with pretrained visual encoders to achieve generalization across diverse tasks such as image captioning, visual-question answering, and spatial reasoning. Visual backbones like DINOv2 [21] and SigLIP [22] provide robust spatial and semantic grounding, while language models such as Gemma [23], LLaMA [24], and many others [17], [25], [26], [27], [28], [29], [30] enable instruction following and compositional reasoning. While these models show impressive zero-shot and few-shot capabilities, they are primarily optimized for passive perception and reasoning tasks. Adapting them for embodied agents introduces challenges in preserving spatial grounding capabilities, visual robustness, and cross-modal alignment when transitioning from static vision–language data to dynamic robotic control.

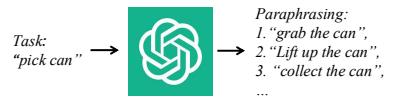
Vision–Language–Action Models (VLAs). VLAs [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41]



(a) Visual Robustness Analysis



(b) Visual Representation Analysis



(c) Language Robustness Analysis

Fig. 1: Motivating experiments. (a) Performance for both OpenVLA (top) and π_0 (down) is significantly impacted by background variations (e.g., background masking and randomization), indicating visual overfitting. (b) t-SNE plots of visual encoders show that pretrained visual representations deteriorate when VLA models are trained by directly finetuning VLMs on action data. (c) Success rates for both OpenVLA and π_0 decrease even with slightly-rephrased instructions, revealing language overfitting. While text augmentation helps, the performance still lags significantly behind our proposed approach.

adapt pretrained VLMs for robotic control by fine-tuning them on action-labeled datasets. A widely adopted approach in works like OpenVLA and Magma [4], [42] remaps rarely used language tokens to bins of discretized actions, discarding their pretrained semantics and breaking alignment with the original vision-language representations. Fine-tuning exclusively on robotic data also leads to degradation of visual representations, reducing robustness to distractors, viewpoint shifts, or rephrased instructions. Several methods attempt to mitigate this via input augmentation [43], [44], spatially enriched encoders [21], [22], [45], [46], or auxiliary cues like end-effector traces [47], [48], but their training data is still limited to robotic domains. Recent works have also explored co-training on both vision-language and action data [5], [6], [31], but it is found that robotic action prediction and vision-language reasoning often involve conflicting training objectives, and naively co-training on both data types can degrade performance in each domain [5]. In this work, we address these co-training issues with a partially frozen visual encoder designed to better preserve pretrained robust visual representations during fine-tuning, and an action tokenizer that reuses pretrained embeddings aligned better with the vision-language training objective.

III. METHOD

The robotic manipulation problem can be modeled as a partially observable Markov decision process (POMDP) [49], [50]. At the beginning of each episode, the agent receives a natural language command $c \in \mathcal{C}$, which remains fixed for the duration of the task. At each timestep t , the agent observes $o_t \in \mathcal{O}$, typically an RGB image, and predicts a short-horizon trajectory $\tau_t = (a_t, \dots, a_{t+H-1})$ consisting of

H low-level actions. Each action $a_t \in \mathcal{A}$ is parameterized as

$$a_t = (\Delta x, \Delta y, \Delta z, \phi, \theta, \psi, g),$$

where $(\Delta x, \Delta y, \Delta z)$ and (ϕ, θ, ψ) denote the end-effector's translation and rotation, and $g \in \{0, 1\}$ indicates the gripper state (open or closed). The policy π thus maps observations and commands to action trajectories, optimized to minimize a task-specific objective,

$$\hat{\tau}_t = \arg \min_{\tau_t} \mathcal{L}(\tau_t, \tau_t^*),$$

where τ_t^* is the expert trajectory.

Directly fine-tuning pretrained VLMs for this objective often results in representation collapse and limited generalization, as shown in Fig. 1. To address this, we introduce three components that can be integrated into the training process of many different VLA models.

A. Partially-Frozen Dual Encoder Architecture

We first study whether using a partially-frozen encoder for VLAs could help retain pretrained representations. Instead of completely freezing visual encoders, which prior work [4] has shown to degrade performance, we propose to use two siamese encoders: a *frozen* one that serves as an “anchor” to retain the robust, semantically rich representations from VLM pretraining, and a *finetuned* one that keeps the full flexibility to be specialized to robotic actions prediction. Mathematically,

$$z_t = [\phi_{\text{frozen}}(o_t) \parallel \phi_{\text{train}}(o_t)],$$

where ϕ_{frozen} denotes the frozen encoder, ϕ_{train} the trainable encoder, and \parallel vector concatenation. The combined latent representation z_t is then passed to a language-conditioned action tokenizer ψ to generate an action token sequence,

$$a_t = \psi(z_t, c),$$

In practice, when the base VLA model provides only a

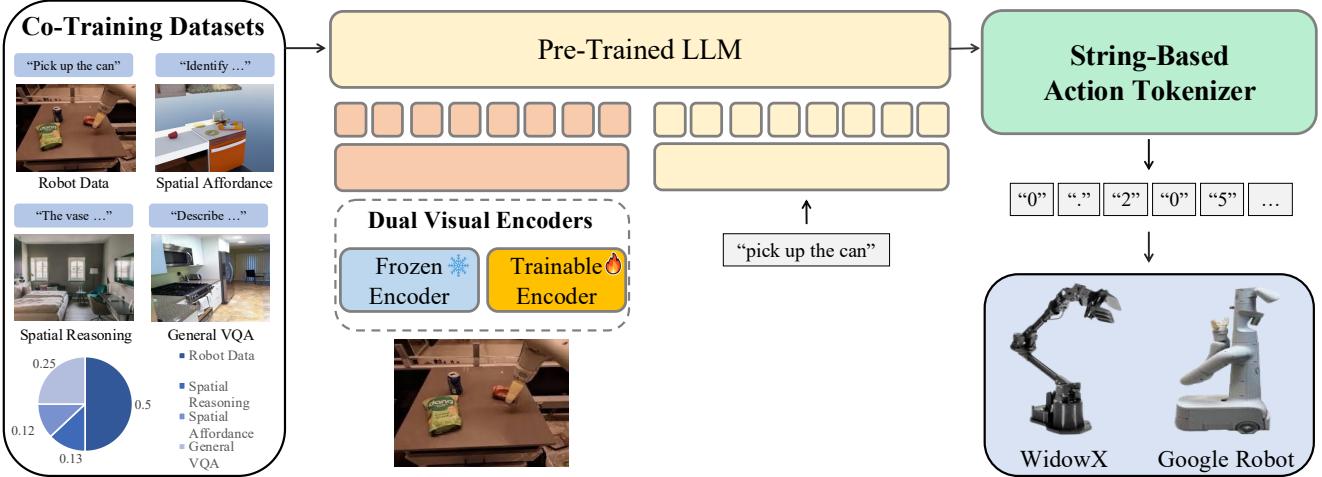


Fig. 2: **Method overview.** Our approach improves VLA generalization through three designs: (a) **Co-training:** Jointly training on robotic and vision-language datasets that emphasize spatial affordance and reasoning helps preserve pretrained representations for generalization. (b) **Partially-frozen visual encoders:** One encoder is frozen to retain robust pretrained features from VLMs, whereas the other keeps the full flexibility to specialize for the robot tasks. (c) **String tokenizer:** Robotic actions are expressed as digit-based strings to maximally reuse pretrained language representations and to unify prediction targets across non-robotic and robotic task domains for co-training.

single vision encoder, we duplicate it and freeze one copy (e.g., [3]); if it already includes two, we freeze one while adapting the other (e.g., [4]).

B. String-Based Tokenizer

To maximally reuse pretrained language representations, we render each action dimension as a character sequence and predict it with the same autoregressive objective used in language pretraining. For example, the action component $\Delta x = 0.0312$ is tokenized into:

$$\boxed{0} \quad \boxed{.} \quad \boxed{0} \quad \boxed{3} \quad \boxed{1} \quad \boxed{2}$$

Each box denotes a single character token from the language vocabulary. Representing numerical values as strings allows a single head to be co-trained on different prediction objectives (i.e., action for robot data and non-action prediction for vision-language datasets), and makes spatial information in pretrained language representations potentially more useful in adapting to robot action prediction, thereby improving generalization. Autoregressive string generation also refines actions step by step, yielding more precise predictions. We train our model with 4 decimals action.

Although string-based action tokenization along with the dual encoder together can increase model inference time by $0.5\times\text{--}1.3\times$ (depending on the model), we later show that the benefits substantially outweigh this cost.

C. Feature Regularization via Co-training

Learning solely from robot datasets often leads to overfitting, particularly in low-data regimes common in robotics. To counter this, we co-train VLAs using a mixture of robot data and vision-language datasets (Table I), enabled by our shared string-based language and action representation. We utilize a mixture of spatial reasoning, spatial affordance, and

TABLE I: Co-training datasets.

| Type | Source | Description |
|---------------------------|---------------------------------|--|
| Robot | OXE [37] | Real robot demonstrations |
| Vision-Language Reasoning | LLaVA Visual Instruct CC3M [51] | Image captioning and object grounding |
| | VQASynth-Spatial [52] | Spatial reasoning via image-language queries |
| | LLaVA OneVision [11] | OCR, chart, and multimodal question answering |
| | RoboPoint [53] | Spatial grounding of language into pixel locations |

general vision-language reasoning data. For every training batch, we sample 50% from each type to balance the gradient. We hypothesize that such a joint training strategy prevents catastrophic forgetting and enhances generalization.

IV. EXPERIMENTS

This section seeks to answer the following questions:

- Q1.** Are the proposed designs effective in learning robot manipulation policies?
- Q2.** Does our approach effectively preserve pretrained VLM representations, thereby improving generalization and robustness to novel visuals and instructions?
- Q3.** How well does the learned policy transfer to real-world environments?

We investigate **Q1-Q2** in simulation and offline vision-language benchmark datasets for scalable and controlled study, and validate **Q3** on real-world robotic platforms.

For simulation, we use SimplerEnv [54], which provides evaluation results that correlate well with real-world behaviors. To systematically study Q1–Q2, we evaluate under two SimplerEnv setups. The **Visual Matching** setup contains visuals closely matching the VLA training data, serving as

TABLE II: Ablation on SimplerEnv (Visual Matching). The Visual Matching evaluation setting contains visuals highly similar to the robot action dataset used for VLA training. D, S, and C denote dual encoder, string tokenizer, and co-training, respectively. OpenVLA⁺ and π_0^+ denote the corresponding models trained with all of D, S, and C.

| Model | Pick Can | Open Drawer | Close Drawer | Move Near | Avg |
|------------------------------------|--------------|--------------|--------------|--------------|--------------|
| OpenVLA | 36.70 | 16.70 | 26.70 | 60.03 | 35.03 |
| OpenVLA+D | 65.33 | 33.93 | 59.20 | 63.75 | 55.55 |
| OpenVLA+S | 78.00 | 1.00 | 65.00 | 57.00 | 50.25 |
| OpenVLA+C | 66.67 | 64.41 | 11.11 | 62.00 | 51.05 |
| OpenVLA+DS | 90.00 | 44.00 | 72.10 | 70.00 | 69.03 |
| OpenVLA+SC | 78.34 | 62.00 | 92.00 | 80.34 | 78.17 |
| OpenVLA ⁺ | 90.32 | 53.50 | 92.40 | 77.60 | 78.46 |
| π_0 w/ Inst. Aug. ¹ | 84.33 | 30.40 | 57.29 | 78.54 | 62.64 |
| π_0^+ | 83.40 | 44.35 | 64.37 | 84.66 | 69.19 |

an *in-distribution* evaluation. The **Visual Variant Aggregation** setup introduces backgrounds, lightings, table textures, distractor objects, and camera poses out of the distribution of the robot data (OOD), serving as a generalization test.

A. Analysis of Different Design Choices

Evaluation setup. We first perform SimplerEnv - Visual Matching evaluation to filter our design choices, as running the Visual Variant Aggregation evaluation is significantly more costly. We evaluate on four representative manipulation tasks, PickCan, OpenDrawer, CloseDrawer, and MoveNear, each over 300 episodes.

Model setup. We utilize two representative VLA architectures: OpenVLA and π_0 , and apply our method from Sec. III. For π_0 , which uses a flow-matching action head, we retain the original head for the baseline and replace it with string-based tokenizer for our method. All models, both baselines and ours, are initialized from their respective pretrained VLMs (DINO + SigLIP + Llama for OpenVLA; PaliGemma for π_0). We then finetune the models on the RT-1 dataset [37], [55], optionally with co-training and our method.

Results. We present the Visual Matching evaluation results in Table II. Overall, models trained with our recipe, OpenVLA⁺ and π_0^+ , outperform OpenVLA and π_0 by nearly 40% and 7% respectively, showcasing the effectiveness of our method.

Effect of partially-frozen dual vision encoder. Utilizing our partially-frozen dual encoder alone improves OpenVLA performance from 35.03% to 55.55%. When combined with our string tokenizer and co-training approaches, the performance further rises to 78.46%, though the gain over only using String Tokenizer + co-training is marginal (78.17% vs. 78.46%). However, the benefits of dual encoder will become clear when we later assess the robustness and generalization of VLAs using vision and language perturbations.

¹Throughout the paper, we trained π_0 baselines with instruction augmentation, as omitting it led to substantially poorer performance; our OpenVLA⁺ and π_0^+ models are trained without such augmentation. This setup is thus favorable to the baselines in comparison.

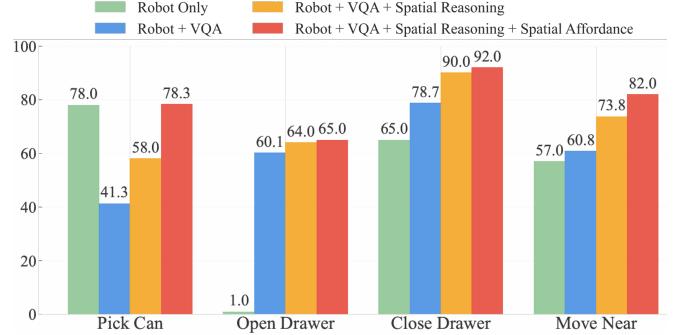


Fig. 3: Co-training data composition ablation on SimplerEnv (Visual Matching). With our unified string-based tokenizer, combining diverse datasets for co-training consistently improves robotic task performance.

Effect of string tokenizer and vision-language co-training.

Adding string tokenizer to OpenVLA alone raises the success rate from 35.03% to 50.25%. Moreover, when vision-language co-training is utilized, the model trained with string tokenizer performs significantly better than the one without (78.17% vs. 51.05%). This result indicates that unifying the action and language output space makes it easier to harness pretrained features, while not compromising the model’s ability to adapt to precise robot actions. We also analyze the effect of data composition in co-training, showing that our selected datasets are well aligned with robotic tasks and consistently yield positive performance gains (Figure 3).

B. Analysis of Representation Generalization and Robustness

Evaluation setup. We assess whether the learned visual and language representations in our models are rich, robust, and thus enable generalizable robot manipulation. We first evaluate **visual robustness** in two setups: (i) **background masking**, where background pixels are masked to test visual sensitivity, and (ii) **visual variant aggregation** in SimplerEnv [54], where task environments are visually randomized to assess generalization across environmental visual appearances. We also visualize learned representations on standard vision datasets for a qualitative analysis. For **language robustness**, we use GPT-4 to generate synonymous instructions (e.g., “grasp the can” and “get the can” for PickCan), and evaluate zero-shot generalization to these unseen language variants. In addition, we evaluate our models on standard VQA benchmarks to analyze their representation quality through their general reasoning capabilities.

Visual robustness. Both OpenVLA⁺ and π_0^+ demonstrate substantial improvements in visual robustness across both background masking and visual variant aggregation. Table III shows our methods outperform their respective baseline models on average across four robotic manipulation tasks.

Visualizing learned visual representations. We extract learned representations on the CIFAR-10 dataset before and after VLA training, and visualize the low-dimensional embeddings obtained with t-SNE (Figure 4). Embeddings of baseline VLAs typically exhibit unclear boundaries between

TABLE III: **Out-of-distribution (OOD) visual generalization evaluation in SimplerEnv.** “Masked Background” refers to Visual Matching evaluation but with background removed (illustrated in Fig. 1(a)). “Visual Variant Aggregation” follows the original evaluation protocol in SimplerEnv, randomizing backgrounds, table textures, lightings, distractors, and camera poses for testing OOD generalization. We also report OpenVLA+SC in this table to show that OpenVLA⁺ has better robustness, despite performing very similarly for in-distribution evaluation in Tab. II.

| Model | Masked Background | | | | | Visual Variant Aggregation | | | | |
|-----------------------|-------------------|-------------|--------------|-----------|-------------|----------------------------|-------------|--------------|-----------|-------------|
| | Pick Can | Open Drawer | Close Drawer | Move Near | Avg | Pick Can | Open Drawer | Close Drawer | Move Near | Avg |
| OpenVLA | 76.3 | 28.6 | 43.8 | 73.3 | 55.5 | 67.8 | 24.7 | 43.0 | 74.0 | 52.4 |
| OpenVLA+SC | 80.5 | 48.3 | 89.5 | 86.2 | 76.2 | 82.3 | 42.3 | 81.3 | 86.6 | 73.1 |
| OpenVLA ⁺ | 82.9 | 50.0 | 87.7 | 85.0 | 76.4 | 87.6 | 48.4 | 87.9 | 82.3 | 76.6 |
| π_0 w/ Inst. Aug. | 86.3 | 25.0 | 42.8 | 75.2 | 57.3 | 77.0 | 32.3 | 38.6 | 65.0 | 53.2 |
| π_0^+ | 70.3 | 36.3 | 64.7 | 73.4 | 61.2 | 60.3 | 35.5 | 60.5 | 83.1 | 59.9 |

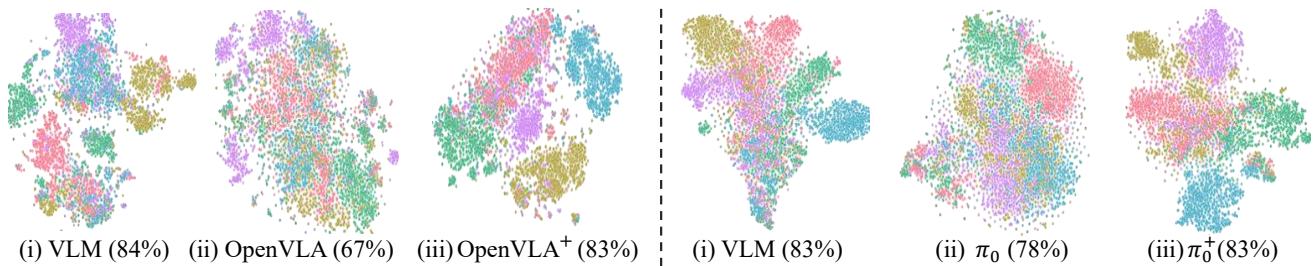


Fig. 4: **t-SNE visualizations of vision encoder features on CIFAR-10.** We compare (i) original visual backbone from VLM before VLA training, (ii) after direct VLA fine-tuning on robot data, and (iii) after our approaches in Sec. III. Numbers indicate linear-probe classification performance on CIFAR-10 using the corresponding features. For both OpenVLA and π_0 , our approach yields noticeably tighter, more well-separated class clusters that lead to better linear-probe performance, indicating better preservation of semantic structures in pretrained visual representations.

TABLE IV: **Language robustness evaluation.** Evaluating policies under original and paraphrased instructions. “w/ Inst. Aug.” indicates training with instruction augmentations. We also report OpenVLA+SC in this table to show that OpenVLA⁺ has better robustness, despite performing very similarly for in-distribution evaluation in Tab. II.

| Model | PickCan | | MoveNear | |
|-----------------------|--------------|--------------|--------------|--------------|
| | Orig. | Para. | Orig. | Para. |
| OpenVLA | 36.70 | 12.12 | 60.03 | 42.07 |
| OpenVLA w/ Inst. Aug. | 30.56 | 38.89 | 56.49 | 50.00 |
| OpenVLA+SC | 78.34 | 62.52 | 80.34 | 82.10 |
| OpenVLA ⁺ | 90.32 | 84.52 | 77.60 | 78.19 |
| π_0 w/ Inst. Aug. | 84.33 | 42.20 | 78.52 | 58.24 |
| π_0^+ | 83.40 | 55.40 | 84.66 | 63.33 |

different classes, while representations from our models are much more linearly separable, yielding much higher linear-probe classification performance.

Language robustness. Table IV compares three settings: (1) baseline VLA models, (2) baselines trained with paraphrased-instruction augmentation, and (3) our models. Unlike (2), where the paraphrased instructions used for evaluation are disjoint from those seen during training, both (1) and (3) have not seen any paraphrased instructions. The same set of paraphrased instructions is used consistently across all evaluations. Despite receiving no language augmentations, our models substantially outperform baselines trained with

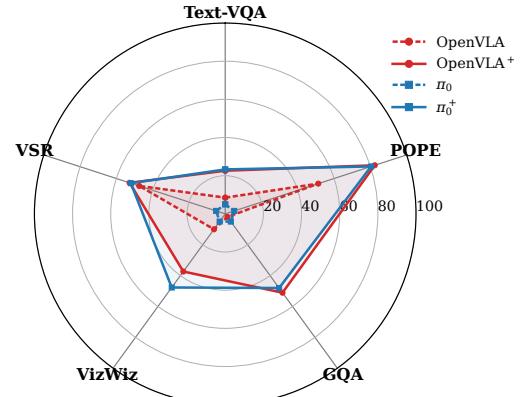


Fig. 5: **Evaluating VLAs on five VQA benchmarks.** OpenVLA⁺ and π_0^+ achieve significantly higher performance across all tasks, showing that our training recipe helps retain pretrained representations of VLM backbones.

augmentation. This shows that our method achieves stronger instruction generalization by leveraging pretrained representations through our string-based action tokenizer, effectively inheriting the language understanding and spatial reasoning capabilities of upstream VLMs.

Reasoning abilities. To further analyze the representation quality of our models, we present results on common VLM benchmarks like Text-VQA [56], POPE [57], GQA [58], VizWiz[59], and VSR[60] and report the accuracy metric in

| Task | Distractor | OpenVLA | OpenVLA ⁺ | π_0 w/ Inst. Aug. | π_0^+ |
|-------------------------|------------|---------|----------------------|-----------------------|-------------|
| PickCarrot | None | 32.0 | 60.0 | 16.0 | 40.0 |
| PickEggplant | None | 8.0 | 24.0 | 8.0 | 24.0 |
| PickKnife | None | 36.0 | 64.0 | 24.0 | 40.0 |
| PlaceKnifeNearCorn | None | 0.0 | 12.0 | 8.0 | 20.0 |
| PutKnifeOnCloth | None | 4.0 | 12.0 | 12.0 | 48.0 |
| PutCarrotOnPlate | None | 4.0 | 16.0 | 12.0 | 36.0 |
| PlaceCornNearKnife | None | 0.0 | 0.0 | 0.0 | 0.0 |
| PickCarrot | Knife | 20.0 | 48.0 | 12.0 | 36.0 |
| PickKnife | Carrot | 12.0 | 36.0 | 8.0 | 28.0 |
| PutCarrotOnPlate | Knife | 8.0 | 12.0 | 8.0 | 36.0 |
| PutCarrotOn{Color}Plate | Plate | 0.0 | 8.0 | 0.0 | 20.0 |
| PutCarrotOnPlate | Cloth | 4.0 | 8.0 | 4.0 | 32.0 |

TABLE V: **Real-world evaluation results.** Distractors are irrelevant objects that may mislead the policy, such as similar items (e.g., differently colored plates) or dissimilar ones (e.g., knife, carrot, cloth), and are used to assess the robustness of the policies. Our models consistently outperform baselines in their presence.

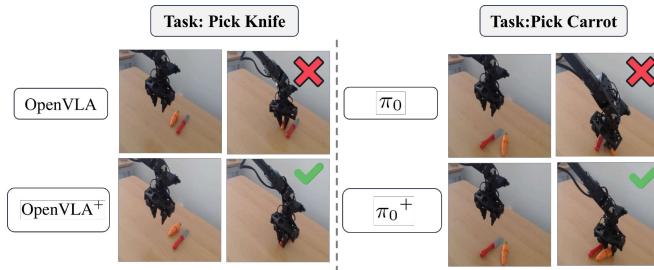


Fig. 6: **Qualitative results on the PickKnife and PickCarrot tasks.** Our models have stronger robustness to distractors.

Fig. 5. These benchmarks include many challenging reasoning questions that place high demands on the underlying representations. We find that baseline OpenVLA and π_0 perform poorly, whereas OpenVLA⁺ and π_0^+ achieve substantially higher accuracy, demonstrating that our training recipe more effectively preserves the reasoning capabilities of pretrained VLM representations within VLAs.

C. Real-World Evaluation

Evaluation setup. We initialized VLAs from their respective pretrained VLMs and finetune the policies on the Bridge dataset. We deploy both baselines and our method on the ViperX 300s² robot to evaluate real-world performance. Our experimental configuration utilizes a Logitech C920 Webcam for visual input [61] and dual GTX 1080 Ti GPUs³. We evaluate models on both in-distribution and out-of-distribution (OOD) instructions, along with potentially unseen distractors, to comprehensively analyze model capabilities. We conduct 25 trials per task and per model. For a 7-DoF action, the inference time is about 1.53s for baseline OpenVLA, 2.30s for OpenVLA⁺, 0.73s for baseline π_0 , and 1.70s for π_0^+ .

Results. Figure 6 and Table V illustrate the results. We find that our models consistently outperform baseline VLAs on all manipulation tasks. In particular, we observe that base

²Although a WidowX setup was not available, we adapted the ViperX control stack to closely match that of the WidowX.

³More advanced GPUs were not available in the room where we conducted the experiments. However, this should not affect our findings.

models such as OpenVLA and π_0 tend to misinterpret the task under distractors, often reaching for the closest object in the scene rather than the instructed target, e.g., picking a carrot instead of a knife in PickKnife, or failing to pick the correct carrot among other similar items. In contrast, our approach demonstrates a more robust understanding of the task, successfully completing goal-conditioned actions even in the presence of semantically or spatially similar distractors, and achieving substantially better performance.

V. CONCLUSION

In this paper, we proposed a set of approaches that improve the robustness and generalization of vision-language-action (VLA) models by better preserving the robust representation structures from pretrained vision-language models (VLMs). These include a dual-visual-encoder design that mixes pretrained and finetuned features, aligning robotic action with language output via a string-based tokenizer to better transfer pretrained knowledge, and a balanced co-training approach on both robot and vision-language reasoning data. Experiments in both simulation and the real world demonstrate that our designs enable better learning of robot manipulation policies, better generalization and robustness to novel visuals and instructions, and better performance in the real-world.

REFERENCES

- [1] R. Bommasani et al., “On the opportunities and risks of foundation models,” *ArXiv*, 2021. [Online]. Available: <https://crfm.stanford.edu/assets/report.pdf>
- [2] K. Goldberg, “Good old-fashioned engineering can close the 100,000-year “data gap” in robotics,” *Science Robotics*, vol. 10, no. 105, eaea7390, 2025. DOI: 10.1126/scirobotics.aea7390 eprint: <https://www.science.org/doi/pdf/10.1126/sciroboticsaea7390> [Online]. Available: <https://www.science.org/doi/abs/10.1126/sciroboticsaea7390>
- [3] K. Black et al., “ π_0 : A vision-language-action flow model for general robot control,” *CoRR*, vol. abs/2410.24164, 2024. DOI: 10.48550/ARXIV.2410.24164 arXiv: 2410.24164. [Online]. Available: <https://doi.org/10.48550/arXiv.2410.24164>

- [4] M. J. Kim et al., "Openvla: An open-source vision-language-action model," in *Conference on Robot Learning*, 6-9 November 2024, Munich, Germany, P. Agrawal, O. Kroemer, and W. Burgard, Eds., ser. Proceedings of Machine Learning Research, vol. 270, PMLR, 2024, pp. 2679–2713. [Online]. Available: <https://proceedings.mlr.press/v270/kim25c.html>
- [5] J. Yang et al., "Magma: A foundation model for multimodal ai agents," *arXiv preprint arXiv:2502.13130*, 2025.
- [6] P. Intelligence et al., $\pi_{0.5}$: A vision-language-action model with open-world generalization, 2025. arXiv: 2504.16054 [cs.LG]. [Online]. Available: <https://arxiv.org/abs/2504.16054>
- [7] L. Beyer et al., "Paligemma: A versatile 3b vlm for transfer," *arXiv preprint arXiv:2407.07726*, 2024.
- [8] S. Bai et al., "Qwen2.5-vl technical report," *CoRR*, vol. abs/2502.13923, 2025. DOI: 10.48550/ARXIV.2502.13923 arXiv: 2502.13923. [Online]. Available: <https://doi.org/10.48550/arXiv.2502.13923>
- [9] J. Zhu et al., *Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models*, 2025. arXiv: 2504.10479 [cs.CV]. [Online]. Available: <https://arxiv.org/abs/2504.10479>
- [10] P. Wang et al., "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," *arXiv preprint arXiv:2409.12191*, 2024.
- [11] B. Li et al., "Llava-onevision: Easy visual task transfer," *arXiv preprint arXiv:2408.03326*, 2024.
- [12] J. Li et al., "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*, PMLR, 2023, pp. 19730–19742.
- [13] D. Zhu et al., "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.
- [14] S. Karamchetti et al., "Prismatic vlms: Investigating the design space of visually-conditioned language models," in *Forty-first International Conference on Machine Learning*, 2024.
- [15] M. Deitke et al., "Molmo and pixmo: Open weights and open data for state-of-the-art multimodal models," *CoRR*, vol. abs/2409.17146, 2024. DOI: 10.48550/ARXIV.2409.17146 arXiv: 2409.17146. [Online]. Available: <https://doi.org/10.48550/arXiv.2409.17146>
- [16] J. Lin et al., "VILA: on pre-training for visual language models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, IEEE, 2024, pp. 26679–26689. DOI: 10.1109/CVPR52733.2024.02520 [Online]. Available: <https://doi.org/10.1109/CVPR52733.2024.02520>
- [17] M. I. Abdin et al., "Phi-3 technical report: A highly capable language model locally on your phone," *CoRR*, vol. abs/2404.14219, 2024. DOI: 10.48550/ARXIV.2404.14219 arXiv: 2404.14219. [Online]. Available: <https://doi.org/10.48550/arXiv.2404.14219>
- [18] B. Li et al., "Llava-onevision: Easy visual task transfer," *CoRR*, vol. abs/2408.03326, 2024. DOI: 10.48550/ARXIV.2408.03326 arXiv: 2408.03326. [Online]. Available: <https://doi.org/10.48550/arXiv.2408.03326>
- [19] W. Hong et al., "Cogvlm2: Visual language models for image and video understanding," *CoRR*, vol. abs/2408.16500, 2024. DOI: 10.48550/ARXIV.2408.16500 arXiv: 2408.16500. [Online]. Available: <https://doi.org/10.48550/arXiv.2408.16500>
- [20] Y. Yao et al., "Minicpm-v: A GPT-4V level MLLM on your phone," *CoRR*, vol. abs/2408.01800, 2024. DOI: 10.48550/ARXIV.2408.01800 arXiv: 2408.01800. [Online]. Available: <https://doi.org/10.48550/arXiv.2408.01800>
- [21] M. Oquab et al., "Dinov2: Learning robust visual features without supervision," *Trans. Mach. Learn. Res.*, vol. 2024, 2024. [Online]. Available: <https://openreview.net/forum?id=a68SUt6zFt>
- [22] X. Zhai et al., "Sigmoid loss for language image pre-training," in *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, IEEE, 2023, pp. 11941–11952. DOI: 10.1109/ICCV51070.2023.01100 [Online]. Available: <https://doi.org/10.1109/ICCV51070.2023.01100>
- [23] T. Mesnard et al., "Gemma: Open models based on gemini research and technology," *CoRR*, vol. abs/2403.08295, 2024. DOI: 10.48550/ARXIV.2403.08295 arXiv: 2403.08295. [Online]. Available: <https://doi.org/10.48550/arXiv.2403.08295>
- [24] H. Touvron et al., "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.
- [25] A. Chowdhery et al., "Palm: Scaling language modeling with pathways," *J. Mach. Learn. Res.*, vol. 24, 240:1–240:113, 2023. [Online]. Available: <https://jmlr.org/papers/v24/22-1144.html>
- [26] DeepSeek-AI et al., "Deepseek-v3 technical report," *CoRR*, vol. abs/2412.19437, 2024. DOI: 10.48550/ARXIV.2412.19437 arXiv: 2412.19437. [Online]. Available: <https://doi.org/10.48550/arXiv.2412.19437>
- [27] DeepSeek-AI et al., "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *CoRR*, vol. abs/2501.12948, 2025. DOI: 10.48550/ARXIV.2501.12948 arXiv: 2501.12948. [Online]. Available: <https://doi.org/10.48550/arXiv.2501.12948>
- [28] S. Gunasekar et al., "Textbooks are all you need," *CoRR*, vol. abs/2306.11644, 2023. DOI: 10.48550/ARXIV.2306.11644 arXiv: 2306.11644. [Online]. Available: <https://doi.org/10.48550/arXiv.2306.11644>
- [29] OpenAI, "GPT-4 technical report," *CoRR*, vol. abs/2303.08774, 2023. DOI: 10.48550/ARXIV.2303.08774 arXiv: 2303.08774. [Online]. Available: <https://doi.org/10.48550/arXiv.2303.08774>
- [30] A. Q. Jiang et al., "Mixtral of experts," *CoRR*, vol. abs/2401.04088, 2024. DOI: 10.48550/ARXIV.2401.04088 arXiv: 2401.04088. [Online]. Available: <https://doi.org/10.48550/arXiv.2401.04088>
- [31] Y. Li et al., "Hamster: Hierarchical action models for open-world robot manipulation," *arXiv preprint arXiv:2502.05485*, 2025.
- [32] J. Wen et al., "Dexvla: Vision-language model with plug-in diffusion expert for general robot control," *arXiv preprint arXiv:2502.05855*, 2025.
- [33] C. Cui et al., "Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation," *arXiv preprint arXiv:2505.03912*, 2025.
- [34] C.-L. Cheang et al., "Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation," *arXiv preprint arXiv:2410.06158*, 2024.
- [35] D. Ghosh et al., "Octo: An open-source generalist robot policy," in *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, D. Kulic et al., Eds., 2024. DOI: 10.15607/RSS.2024.XX.090 [Online]. Available: <https://doi.org/10.15607/RSS.2024.XX.090>
- [36] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conference on Robot Learning, CoRL 2022, 14-18 December 2022, Auckland, New Zealand*, K. Liu, D. Kulic, and J. Ichnowski, Eds., ser. Proceedings of Machine Learning Research, vol. 205, PMLR, 2022, pp. 785–799. [Online]. Available: <https://doi.org/10.48550/arXiv.2408.01800>

- [37] A. O'Neill et al., "Open x-embodiment: Robotic learning datasets and RT-X models : Open x-embodiment collaboration," in *IEEE International Conference on Robotics and Automation, ICRA 2024, Yokohama, Japan, May 13-17, 2024*, IEEE, 2024, pp. 6892–6903. DOI: 10.1109/ICRA57147.2024.10611477 [Online]. Available: <https://doi.org/10.1109/ICRA57147.2024.10611477>
- [38] A. Goyal et al., "RVT-2: learning precise manipulation from few demonstrations," in *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, D. Kulic et al., Eds., 2024. DOI: 10.15607/RSS.2024.XX.055 [Online]. Available: <https://doi.org/10.15607/RSS.2024.XX.055>
- [39] Y. Jiang et al., "VIMA: general robot manipulation with multimodal prompts," *Corr*, vol. abs/2210.03094, 2022. DOI: 10.48550/ARXIV.2210.03094 arXiv: 2210.03094 [Online]. Available: <https://doi.org/10.48550/arXiv.2210.03094>
- [40] J. Liu et al., "Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model," *arXiv preprint arXiv:2503.10631*, 2025.
- [41] J. Bjorck et al., "Gr0ot n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.
- [42] B. Zitkovich et al., "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*, PMLR, 2023, pp. 2165–2183.
- [43] T. Xiao et al., "Robotic skill acquisition via instruction augmentation with vision-language models," in *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, K. E. Bekris et al., Eds., 2023. DOI: 10.15607/RSS.2023.XIX.029 [Online]. Available: <https://doi.org/10.15607/RSS.2023.XIX.029>
- [44] T. Yu et al., "Scaling robot learning with semantically imagined experience," in *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, K. E. Bekris et al., Eds., 2023. DOI: 10.15607/RSS.2023.XIX.027 [Online]. Available: <https://doi.org/10.15607/RSS.2023.XIX.027>
- [45] A. Radford et al., "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, M. Meila and T. Zhang, Eds., ser. Proceedings of Machine Learning Research, vol. 139, PMLR, 2021, pp. 8748–8763. [Online]. Available: <https://proceedings.mlr.press/v139/radford21a.html>
- [46] J. Shang et al., "Theia: Distilling diverse vision foundation models for robot learning," in *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, P. Agrawal, O. Kroemer, and W. Burgard, Eds., ser. Proceedings of Machine Learning Research, vol. 270, PMLR, 2024, pp. 724–748. [Online]. Available: <https://proceedings.mlr.press/v270/shang25a.html>
- [47] J. Gu et al., "Rt-trajectory: Robotic task generalization via hindsight trajectory sketches," in *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, OpenReview.net, 2024. [Online]. Available: <https://openreview.net/forum?id=F1TKzG8LJO>
- [48] D. Niu et al., "LLARVA: vision-action instruction tuning enhances robot learning," in *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, P. Agrawal, O. Kroemer, and W. Burgard, Eds., ser. Proceedings of Machine Learning Research, vol. 270, PMLR, 2024, pp. 3333–3355. [Online]. Available: <https://doi.org/10.1109/ICRA57147.2024.10611477>
- [49] B. Ai et al., "Deep visual navigation under partial observability," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.
- [50] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, no. 1-2, pp. 99–134, 1998.
- [51] P. Sharma et al., "Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2556–2565.
- [52] B. Chen et al., "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14455–14465.
- [53] W. Yuan et al., "Robopoint: A vision-language model for spatial affordance prediction in robotics," in *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, P. Agrawal, O. Kroemer, and W. Burgard, Eds., ser. Proceedings of Machine Learning Research, vol. 270, PMLR, 2024, pp. 4005–4020. [Online]. Available: <https://proceedings.mlr.press/v270/yuan25c.html>
- [54] X. Li et al., "Evaluating real-world robot manipulation policies in simulation," in *Conference on Robot Learning, 6-9 November 2024, Munich, Germany*, P. Agrawal, O. Kroemer, and W. Burgard, Eds., ser. Proceedings of Machine Learning Research, vol. 270, PMLR, 2024, pp. 3705–3728. [Online]. Available: <https://proceedings.mlr.press/v270/li25c.html>
- [55] A. Brohan et al., "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.
- [56] A. Singh et al., "Towards vqa models that can read," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8317–8326.
- [57] Y. Li et al., "Evaluating object hallucination in large vision-language models," *arXiv preprint arXiv:2305.10355*, 2023.
- [58] D. A. Hudson and C. D. Manning, "Gqa: A new dataset for real-world visual reasoning and compositional question answering," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 6700–6709.
- [59] J. P. Bigham et al., "Vizwiz: Nearly real-time answers to visual questions," in *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, 2010, pp. 333–342.
- [60] H. Liu et al., "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 34 892–34 916, 2023.
- [61] H. R. Walke et al., "Bridgedata V2: A dataset for robot learning at scale," in *Conference on Robot Learning, CoRL 2023, 6-9 November 2023, Atlanta, GA, USA*, J. Tan, M. Toussaint, and K. Darvish, Eds., ser. Proceedings of Machine Learning Research, vol. 229, PMLR, 2023, pp. 1723–1736. [Online]. Available: <https://proceedings.mlr.press/v229/walke23a.html>