

# ReVLA: Reverting Visual Domain Limitation of Robotic Foundation Models

Sombit Dey<sup>1</sup>, Jan-Nico Zaech<sup>1</sup>, Nikolay Nikolov<sup>1</sup>, Luc Van Gool<sup>1</sup>, Danda Pani Paudel<sup>1</sup>

**Abstract**—Recent progress in large language models and access to large-scale robotic datasets has sparked a paradigm shift in robotics models transforming them into generalists able to adapt to various tasks, scenes, and robot modalities. A large step for the community are open Vision Language Action models which showcase strong performance in a wide variety of tasks. In this work, we study the visual generalization capabilities of three existing robotic foundation models, and propose a corresponding evaluation framework.

Our study shows that the existing models do not exhibit robustness to visual out-of-domain scenarios. This is potentially caused by limited variations in the training data and/or catastrophic forgetting, leading to domain limitations in the vision foundation models. We further explore OpenVLA, which uses two pre-trained vision foundation models and is, therefore, expected to generalize to out-of-domain experiments. However, we showcase catastrophic forgetting by DINO-v2 in OpenVLA through its failure to fulfill the task of depth regression.

To overcome the aforementioned issue of visual catastrophic forgetting, we propose a gradual backbone reversal approach founded on model merging. This enables OpenVLA – which requires the adaptation of the visual backbones during initial training – to regain its visual generalization ability. Regaining this capability enables our ReVLA model to improve over OpenVLA by a factor of 77% and 66% for grasping and lifting in visual OOD tasks. Comprehensive evaluations, episode rollouts and model weights are available on the [ReVLA Page](#)

## I. INTRODUCTION

Generalist robot foundation models based on vision-language models (VLMs) offer a promising path toward developing dexterous policies that generalize across tasks, embodiments, and environments. This expectation is primarily based on the capabilities of the underlying large language models (LLMs) like Llama [1] that have been demonstrated in a range of real-world applications. VLMs equip LLMs with foundational vision encoders, such as DINO [2], CLIP [3] or SigLIP [4], enabling them to perform complex tasks that require perception and reasoning. Training of robot foundation models – which output the robot’s action trajectories – requires training on large amounts of robot-specific data. Doing so adequately is a delicate task, which we study in this paper while focusing on the visual generalization capabilities in the out-of-domain (OOD) scenarios.

We begin by observing that most demonstrations used for training robotic foundation models are acquired in controlled lab settings and prioritize trajectory variability rather than comprehensive coverage of all possible interaction objects and environments. Simultaneously, training protocols such

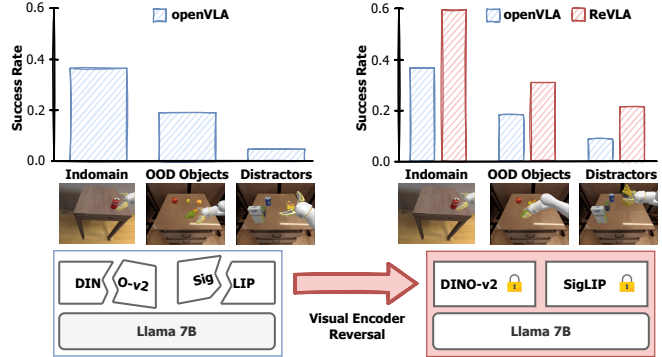


Fig. 1: The OpenVLA model tuned on fractal data struggles with out-of-domain objects and in the presence of distractors (left), due to catastrophic forgetting in its DINO-v2 and SigLIP vision encoders. Our ReVLA (DS gradual) model addresses this by reverting the vision encoders to their original pre-trained weights, leading to improved overall performance across all three settings (right).

as those in OpenVLA [5] require adapting not only the LLM-based reasoning model but also the vision backbones to achieve high performance on in-domain tasks. Although pre-trained vision models themselves are known for generalizing well across diverse settings, their adaptation during the training of robotic foundation models can lead to visual domain limitations due to catastrophic forgetting.

To quantify the impact of this training approach and assess the overall generalization capability of robotic foundation models, we develop a realistic out-of-domain evaluation framework based on the SIMPLER [6] environment. This evaluation framework includes challenging visual tasks using OOD objects and visual distractors. We use it to evaluate the visual generalization performance of three recent open-source robotic foundation models: (1) RT-1 [7], which pioneered end-to-end generalist policies and employs an EfficientNet backbone alongside a transformer model for predicting low-level robot control commands; (2) Octo [8], which is built on a diffusion policy integrated with a transformer model; and (3) OpenVLA [5], the first publicly available vision-language-action model for robot control, built on the Prismatic [9] VLM that combines Llama [1] with DINO-v2 and SigLIP. In all three cases, the vision encoders are either trained or fine-tuned on visually limited robotics data.

In our evaluations, we observe that all three models exhibit limited visual generalization performance. However, OpenVLA shows strong potential due to its powerful VLM

<sup>1</sup>INSAIT, Sofia University “St. Kliment Ohridski”, Bulgaria  
Corresponding Author: jan-nico.zaech@insait.ai

foundation. Building on this, we propose a model reversal approach for the VLM’s visual encoders, motivated by the expectation that the original pre-trained vision encoders should generalize well. Our approach builds on the concept of gradual model merging that is widely utilized for pre-trained foundation models to slowly return to the pretrained weights. Overall, it enables a return to these generalizable visual embeddings, even when the training protocol for robotic foundation models requires fine-tuning the entire architecture, as is the case with OpenVLA.

In summary, we make three major contributions:

- A framework for evaluating visual OOD performance of robotic foundation models containing OOD objects and distractors.
- A model reversal approach to regain generalization performance of visual encoders, even when the adaptation is required during training.
- An in-depth analysis of our “ReVLA” approach through thorough experimental evaluations.

## II. RELATED WORKS

### A. Vision Language Models

VLMs [9], [10], [11] combine the strong generalization and few-shot learning capabilities of LLMs with the ability to understand the rich information present in image data. This is unlocked by utilizing one or more vision foundation models. In this process, the vision model serves the purpose of translating the spatially high-dimensional and strongly correlated information into a spatially sparse representation, but high-dimensional in the feature space, and thus, well suited for processing in an LLM.

Early steps were made by CLIP [3] by embedding images and text into a joint feature space that is trained using contrastive learning on a large corpus of image-text pairs. While the model is not directly suitable for visual question answering, it provides powerful features for downstream applications like image classification and retrieval. BLIP [12] improves these capabilities by allowing interactions between text and vision inputs using the Q-former [13] architecture.

By combining a pretrained Llama [1] LLM with a CLIP [3] encoder as input to the LLM, LLaVA [10] utilizes the potential present in internet-scale text data for vision tasks. This paradigm is further used by approaches such as Flamingo [14], BLIP-2 [15] or PaliGemma [11]. While architectures based on CLIP are well suited for image-level understanding, robotic problems require pixel-level semantic and spatial information. This is approached by combining image encoders in PixelLM [16] and Prismatic [9], which is the basis of OpenVLA [5] and our work.

### B. Generalist Robotics Models

Similar to the development in VLMs, progress in robotics strongly revolves around the development of generalist models trained on large-scale datasets to perform diverse tasks across various embodiments and scenarios. RT1 [7] introduced a language-conditioned model that directly maps from text and image inputs to robot control commands. By training

Model	Vision Backbone	Reasoning Model	Dataset
RT-1	EfficientNet+FiLM	Transformer	Fractal
Octo	Shallow CNN	Transformer+ Diffusion	OpenX Embodiment
OpenVLA	DINOv2+SigLIP	Llama V2	OpenX Embodiment

TABLE I: Summary of generalist robotic foundation models.

a similar architecture on the OpenX dataset, which includes over 1 million trajectories from 22 robot embodiments, RT1-X [17] demonstrated superior performance, transferring knowledge across diverse robots and tasks [17].

RT-2 [18] advanced this further by building on top of VLA models, including PaLI-X [19] and PaLM-E [20], co-fine-tuning them on both internet-scale and robotic data. This allowed RT-2 to exhibit first emergent reasoning abilities and superior generalization, outperforming RT1 in handling unseen objects and environments [18].

Octo [8] approaches robotic manipulation using diffusion-based by denoising the next actions performed by the robot. Its architecture supports both language and goal image conditioning and improves the generalization in manipulation tasks compared to RT1-X [8].

The current state-of-the-art model, OpenVLA [5] is the first open-source generalist model utilizing a LLM. It is based on the prismatic VLM [9], and trained on the OpenX embodiment dataset. It surpasses both specialist and foundational models, including RT-2, across various benchmarks. A summary of the publicly available current robotic foundation models is provided in Table I.

### C. Digital Twins and Photorealistic simulators

Simulators are important tools in robotics research, offering a cost-effective and scalable alternative to real-world data collection and experiments [21]. Real-world robotics evaluation is strongly constrained by time and cost and does not offer reproducibility, especially across different setups. To overcome these challenges, simulation environments like SAPIEN [22], Habitat [23], and Isaac Sim [24] provide nearly photorealistic proxies for real-world experiments.

In this work, we build on top of and extend standardized benchmarks for robotics, since they facilitate direct and reproducible comparisons across models. While a sim-to-real gap [25] also remains in current approaches, considerable progress has been made in matching domains and a strong correlation between simulation and real-world performance has been verified for foundation models in SIMPLER [6], which serves as the basis of our work.

For building evaluation scenarios, ALFRED [26], GraspNet [27], and MotionBenchMaker [28] provide comprehensive task definitions to evaluate perception and manipulation. In our work, we utilize the grasping task definition in SIMPLER [6] and extend it with the YCB object dataset [29] to model OOD objects and distractor settings.

### D. Model Merging

Efficiently reusing models trained on different datasets is crucial for applications that utilize large-scale data, since

training all models from scratch is prohibitively costly and requires large amounts of energy. Model merging [30], [31] is an approach to gain the different capabilities of pretrained models trained on different tasks and datasets, by combining their parameter weights. The approach has been shown to help save training GPU hours, mitigate catastrophic forgetting [32], and boost performance in domain generalization [33]. In VLMs, model merging [34] results in better out-of-domain generalization and better performance and mitigates the inference overhead from using multiple encoders. In this work, we utilize linear model merging to revert domain-limited encoders to well-generalizing models.

### III. BACKGROUND

This section provides the background information relevant to our model reversal approach. As the foundation of our model, we introduce Prismatic [9] and OpenVLA [5] as a corresponding VLM and VLA pair. For our benchmark, we utilize the SAPIEN [22] simulator with the SIMPLER [6] environment which is introduced in the second part.

#### A. Vision Language Action Models

OpenVLA [5] is currently the only open-source, generalist, foundational robotics model for manipulation tasks. OpenVLA builds upon the Prismatic VLM [9] architecture and weights, by tuning the model on the OpenX dataset [17] to output discrete action tokens for end-effector control similar to RT1 and RT2 [7], [18].

The model first processes the visual input using SigLIP [4] and DINO-v2 [2] to generate features respectively. The DINO-v2 [2] encoder is a self-supervised trained vision encoder that allows for spatial and relational understanding, while SigLIP [4], trained using contrastive learning between image-language pairs, is incorporated for semantic and linguistic multimodal understanding. A tokenizer projects the image features to a Llama 6.7B [1] LLM that processes both, language instructions and the tokenized visual features.

During training on the OpenX dataset end effector control is obtained by seven output action tokens ( $\Delta x, \Delta y, \Delta z, \Delta \alpha, \Delta \beta, \Delta \gamma, \Delta \text{gripper}$ ) for every image-language pair as input. For this task, the 256 least used tokens are overwritten. During training of OpenVLA on the OpenX dataset [17] tuning of the whole model, including the two vision foundation models is required to achieve sufficiently stable robot trajectories [5].

#### B. Simpler-Benchmark

Our benchmark extends the SIMPLER [6] environment, which itself is an adaptation of the Maniskill benchmark [35], based on the SAPIEN Simulator [22]. We chose SIMPLER as it addresses the two main challenges in simulated robotics evaluations, the control and visual gap. Therefore, results achieved in SIMPLER closely match real-world comparisons between algorithms and allow us to avoid hardly reproducible robot experiments.

**Control Gap:** The goal of reducing the control gap between simulated and real-world environments is to ensure

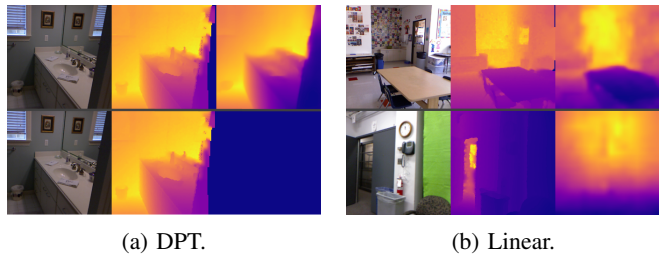


Fig. 2: Depth regression from original DINO-v2 (top-right) and OpenVLA DINO-v2 (bottom-right) using DPT head (a) and linear probing (b). Each row: input image, ground-truth, predicted depth. The original DINO-v2 estimates depths correctly, whereas OpenVLA DINO-v2 performs poorly due to catastrophic forgetting during OpenVLA training.

that policy actions performed in simulation produce similar outcomes on the robot’s end-effector when applied to the real robot, thus minimizing the real-to-sim gap. SIMPLER [6] aims to mitigate these differences by optimizing the parameters of the robot controller in the simulator such that the simulator trajectories, when rolled out in an open-loop manner for a given sequence of actions, are close to the real-world trajectories.

**Vision Gap:** Discrepancies in the visual appearance between a simulator and real-world observation cause a domain shift, which can harm the behavior of learned policies [36], [37]. To mitigate the visual gap, SIMPLER proposes a “green screening” approach. This involves overlaying the real-world environment’s background with the images rendered by the simulator, providing a realistic visual representation.

Thus, SIMPLER is a realistic setting to evaluate robot foundation models in simulations that are reproducible, low-cost, and fast, and have demonstrated to preserve the rank correlation in model performances to real robot experiments.

### IV. REGAINING VISION ENCODER CAPABILITIES

Our method aims at reverting the vision foundation models used in OpenVLA back to their original state in Prismatic, pretrained on a large corpus of diverse vision-language tasks. Therefore, the following section first uncovers the catastrophic forgetting present in the OpenVLA backbones, based on this, two strategies for model reversal are proposed. Finally, baselines for a finetuned version of OpenVLA are established to allow us for a fair comparison between our RevLA and the base OpenVLA models.

#### A. Catastrophic Forgetting in OpenVLA

The OpenX dataset [17] is an important basis for the training of robotic foundation models, containing more than 1 million episodes of robotic demonstrations. This diversity spans a wide range of robot embodiments, tasks, and trajectories, sufficient for training generalist models that have the ability to operate across various modalities. However, the variation present in OpenX dataset is primarily focused on task complexity and embodiment rather than on visual and scene diversity.

In contrast, large-scale VLMs are trained on datasets that primarily emphasize visual diversity and thus are substantially richer in this aspect. Even more, vision foundation models like DINO-V2 [11], are trained on partly private data which further enhances diversity and generalizability.

This lack of visual diversity presents a particular challenge when training robotic foundation models. RT-2 [18] approaches this challenge by co-training on the WebLI dataset [19], however, this also considerably adds to the training requirements, which are a limiting factor for open models like OpenVLA. The risk of the model forgetting is further amplified in OpenVLA due to the need for fine-tuning both the vision encoders and the LLM simultaneously to reach maximum performance, which is in contrast to the approach taken in the underlying Prismatic architecture.

To demonstrate the catastrophic forgetting in the vision encoders we utilize DINO-v2 features for depth estimation using linear probing as well as a dense prediction (DPT) transformer head [38] as shown in Figure 2. When tasked with regressing to depth information, the trained vision encoder in OpenVLA collapses to a constant output for DPT and generates depth maps without details for linear probing, reflecting a loss of spatial understanding, while the pretrained DINO-v2 features generate high-quality depth maps, especially with DPT. This collapse is a clear indicator of catastrophic forgetting, particularly within the vision backbone, as the model fails to retain core spatial competencies learned in earlier stages of training.

This evidence highlights a key challenge in training large-scale robotics models: although fine-tuning the entire architecture, including the vision backbone, can improve stability, it may reduce the model’s ability to generalize, particularly in visually diverse environments.

### B. ReVLA: Reverting the Vision Backbones

Approaching the challenge of catastrophic forgetting in VLA vision encoders, we propose ReVLA as an approach to revert back to the pretrained vision encoders and their inherent capabilities, while maintaining the training protocol of OpenVLA ensuring high-quality robot trajectories. Towards solving this task, we evaluate two different training approaches inspired by model merging as well as the influence of the two different encoders, SigLIP[4] and DINO-v2 [2]. We base our work on the most powerful version of OpenVLA trained on the complete OpenX dataset [17], and perform the encoder reversal using the Fractal RT-1 robot action dataset [7], since it is part of the OpenX dataset and contains a wide range of robotic tasks and uses the same embodiment as our simulator.

**OpenVLA:** We utilize OpenVLA trained on the full OpenX dataset without additional finetuning as our first baseline since it has shown good generalization performance across embodiments and basic emergent behavior.

**OpenVLA-fractal:** To account for a potential bias by fine-tuning on a well-fitting subset of the whole OpenX dataset, we finetune the original OpenVLA model on the Fractal RT-1 robot action dataset which becomes our second baseline

model. We follow the original OpenVLA training protocol and tune the whole model including the visual encoders for the same number of steps and the same hyperparameters to have a fair comparison with our approach and evaluation.

**ReVLA:** ReVLA follows the OpenVLA architecture and is initialized with weights trained on OpenX. As the core component of ReVLA, visual encoder reversal is performed during training. The reversal process is inspired by continuous model merging approaches, where the weights of one model are gradually integrated into a second differently trained one. However, different from model merging, our process ends with the fully pretrained weights, rather than with an interpolation, as the pretrained model covers a considerably larger domain that is required for OOD tasks and emergent behavior.

We follow a linear merging schedule of the vision encoder weights, which slowly reverts from the OpenVLA weights  $\theta_{OpenVLA}$  to the pretrained encoder weights, according to

$$F_{theta} = (1 - \alpha)\theta_{OpenVLA} + \alpha\theta_{Pretrained}, \quad (1)$$

where  $\alpha$  is the weight of the mixing value of pre-trained dino weights. For computational efficiency, the linear transition is implemented as a step-wise curriculum of changing  $\alpha$ , every  $n$  steps by a value  $\frac{1}{k}$ . The total training steps is  $N = k \times n$ . In this work, we choose  $N = 100K$  steps and  $n = 10k$  steps.

OpenVLA utilizes two visual encoders, DINO-V2 for spatial and 3D reasoning tasks and SigLIP for semantic understanding. In our process, both models are reverted: DINO-v2 since OOD generalization requires strong spatial understanding beyond in-domain tasks, and SigLIP, since interactions in an open world require the capability to describe objects that are not present in the robotic training data. Furthermore, while training ReVLA we keep the vision backbone frozen. We refer to this method as ReVLA (DS gradual) in the following. To verify our design choices, we define four models:

- 1) ReVLA (DS gradual), gradually reverts both backbones.
- 2) ReVLA (DS flip), does not employ a gradual model-merging approach but restores the weights of both backbones at the start of training.
- 3) ReVLA (D gradual), where only the DINO-v2 encoder is slowly reverted and SigLIP is kept frozen.
- 4) ReVLA (D flip), which combines the two approaches and only reverts DINO-v2 using hard flipping at the start of the reversal process.

## V. EXPERIMENTS AND RESULTS

Our experiments investigated the following questions,

- How does the out-of-domain performance of ReVLA differ from OpenVLA?
- How does reverting vision backbones affect indomain performance?
- What is the best protocol to revert a vision backbone for a VLA?

Given the computational cost of training on the entire OpenX dataset, we limit our experiments to the Fractal dataset [7], which consists of approximately 73.5k robot trajectories.

This dataset is based on Table-Top and kitchen-top scenes used in the Google robot manipulation tasks and includes template-based language annotations (e.g., “Pick up the Coke Can”). All experiments were conducted using one Nvidia A100 node, training each model for 100,000 steps with identical hyperparameters.

### A. Embodied OOD Benchmark

**In Domain experiments:** To evaluate the overall impact of reverting the vision encoders of openVLA, we first assess ReVLA on the SIMPLER benchmark [6], which represents in-domain scenarios for openVLA. We chose SIMPLER as it minimizes the sim-to-real gap by using simulation environments closely aligned with the Fractal dataset.

The SIMPLER benchmark comprises two sub-evaluations: Visual Matching and the Variant Aggregation protocols, which take two different approaches for closing the domain gap; The *Visual Matching* protocol simulates photorealistic conditions as closely as possible, ensuring minimal visual discrepancy with the Fractal dataset. The *Variant Aggregation* protocol uses strong domain randomization, altering factors like lighting, table textures, backgrounds, and visual distractors to assess the model’s robustness to environmental changes. While Variant Aggregation represents diverse scenarios it visually does not closely resemble a real scenario. Thus, as ReVLA is strongly aimed at visual performance, we chose visual matching as the basis for our evaluation.

In Table II, we compare ReVLA to other foundational models, including RT-1-X, Octo, and OpenVLA. To avoid bias we choose our baseline methods as described in Section IV-B and train on the Fractal dataset for an equal number of steps. We follow the notion of SIMPLER, where for an episode to be successful, the robot has to pick up the target object and lift it up for a duration of time.

**Out of domain experiments:** To evaluate the robustness of our models to OOD tasks, we modify the simple *Pick Coke Can* task by introducing new, unseen objects into the scene. These experiments are conducted following the *Visual Matching* protocol and are designed to assess the models’ ability to generalize in more realistic, complex environments. For each setting we evaluate over 36 episodes, resulting in a total of 216 OOD evaluation scenarios.

**OOD objects:** In the first set of experiments, we introduce several OOD objects, selected from the YCB dataset [29] (*Pear*, *Mustard Bottle*, and *Tomato Can*) to test the generalization capability of models to unseen objects manipulations. Each object presents a distinct set of difficulties:

- *Tomato Can*: This object closely resembles in-domain items such as the *Coke Can* and *RedBull Can*, providing a test of the open-set recognition ability as well as the ability to differentiate between visually similar objects.
- *Pear*: In addition to not being present in the training dataset, the asymmetrical shape of the pear adds complexity to the task of grasping, requiring the model to demonstrate awareness of the 3D shape and adapt its manipulation strategy accordingly.

Policy		Pick Coke Can			
		Horizontal	Vertical	Standing	Average
Var Aggregation	RT-1	0.969	0.760	0.964	0.897
	RT-1-X	0.569	0.204	0.698	0.490
	Octo-Base	0.005	0.000	0.013	0.006
	OpenVLA	<b>0.644</b>	0.218	<b>0.729</b>	<b>0.530</b>
	OpenVLA-fractal	0.483	0.231	0.368	0.361
	ReVLA (flip)	0.360	0.422	0.469	0.417
	ReVLA (gradual)	0.483	<b>0.469</b>	0.610	0.521
Visual Matching	RT-1	0.969	0.900	0.710	0.857
	RT-1-X	0.860	0.790	0.480	0.710
	Octo-Base	0.210	0.210	0.090	0.170
	OpenVLA	0.310	0.030	0.190	0.177
	OpenVLA-fractal	<b>0.655</b>	0.135	0.295	0.361
	ReVLA (flip)	0.5475	0.357	0.726	0.544
	ReVLA (gradual)	0.600	<b>0.413</b>	<b>0.792</b>	<b>0.600</b>

TABLE II: In domain evaluation with SIMPLER [6].

- *Mustard Bottle*: As a significantly larger object compared to the typical items in the dataset, the Mustard Bottle evaluates the model’s ability to adapt its policy for size variations and to execute robust grasping behaviors in such cases.

**OOD Objects with Distractor:** To further challenge the models, we introduce additional distractors around the target objects. Picking a single object in isolation does not fully assess visual robustness. By surrounding the target object with other items, we aim to evaluate the models’ visual grounding capabilities, as the correct object must be selected based on the language prompts and visual clues.

These distractors are chosen from the YCB dataset [29], as well as from the objects available in SIMPLER. Thus, this experiment can further evaluate a bias of the model to grasp objects that were present during training. Finally, distractors often occlude the target object, making the task even more demanding by forcing the model to work under partial visibility.

By combining OOD objects with distractors, these experiments highlight the model’s ability to handle complex scenes, demonstrating both visual robustness and the capacity for multimodal reasoning under challenging conditions.

### B. Results

We evaluate ReVLA using both SIMPLER protocols in Table II, variant aggregation as well as visual matching for their in-domain performance. Using visual matching, our models outperform the OpenVLA baselines by 42.3% and 23.9% for the OpenX and Fractal tuned models respectively. This demonstrates that ReVLA retains and improves upon OpenVLA’s capabilities for in-domain tasks.

Using variant aggregation, the performance for a wider range of tasks parameters is evaluated. In this scenario, our best model outperforms OpenVLA trained on Fractal by 14.9% and is comparable with the OpenX-trained model with a drop of 0.8% in performances. This may be explained by the smaller variety of tasks present in Fractal.

Our OOD evaluation is presented in Table III. Compared to the in-domain task of visual matching, all models lose

Policy	Single			Distractor			Overall		
	Pear	Mustard Bottle	Tomato Can	Pear	Mustard Bottle	Tomato Can	Single	Distractors	Total
RT1-X	0.222	0.000	0.118	0.167	0.000	0.059	0.113	0.075	0.094
Octo	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
OpenVLA	0.194	0.083	0.389	0.056	0.028	0.222	0.222	0.102	0.162
OpenVLA-fractal	0.139	0.028	0.389	0.056	0.028	0.167	0.185	0.084	0.135
ReVLA (D flip)	0.306	0.056	0.306	0.222	0.000	0.389	0.223	0.204	0.213
ReVLA (DS flip)	<b>0.440</b>	<b>0.194</b>	0.361	<b>0.500</b>	0.083	0.139	0.330	<b>0.241</b>	<b>0.287</b>
ReVLA (D gradual)	0.222	0.139	<b>0.556</b>	0.167	0.000	<b>0.472</b>	0.306	0.213	0.259
ReVLA (DS gradual)	0.389	0.110	0.528	0.306	<b>0.110</b>	0.220	<b>0.340</b>	0.213	0.278

TABLE III: Success rate of robotics foundation models in OOD tasks.

Model	Success Lift	Success Grasp
openVLA	0.162	0.348
openVLA-fractal	0.135	0.317
ReVLA (Flip)	<b>0.287</b>	0.495
ReVLA (Gradual)	0.278	<b>0.579</b>

TABLE IV: Grasping and lifting success in OOD tasks.

performance, which is the expected behavior due to the domain gap. Nevertheless, all ReVLA models improve the OOD performance compared to OpenVLA, with the best OOD model ReVLA (DS flip) improving by 12.5%, which corresponds to a factor of 77%. The strongest loss of performance is encountered for RT1, which is the best model in the in-domain visual matching task, and Octo, which is not also to solve any OOD task.

### C. Ablation

We conducted an ablation study to assess the design choices in our ReVLA architecture, specifically focusing on two key aspects: (1) the impact of a gradual transition during the reversal process, and (2) whether the reversal should be applied only to the DINO-v2 encoder or to both visual encoders (DINO-v2 and SigLIP).

The performance results, shown in Table III, indicate that ReVLA (Flip) marginally outperforms ReVLA (Gradual) by 0.9%. However, upon further analysis of robot trajectories in Table IV, which evaluates partial success (i.e., how often the robot successfully grasps the target object), ReVLA (Gradual) demonstrates a clear advantage. It is considerably more effective in locating and grasping the correct target object compared to ReVLA (Flip).

Notably, in many of the partially successful episodes, although the robot grasps the object, it either fails to lift it properly or the object slips shortly after being lifted. This suggests that the drop in overall performance, despite higher grasping success, may be attributed to a lack of precise control during the lifting phase.

Overall, we find the following answers to above questions,

- Reverting both vision encoders, helps ReVLA to regain the zero-shot multi-modal capability of SigLIP and the representative power of DINO-v2, resulting in better performance in OOD scenarios.
- Reverting the vision encoders further helps the in-domain performance compared to openVLA, likely owing to a better and more precise spatial understanding

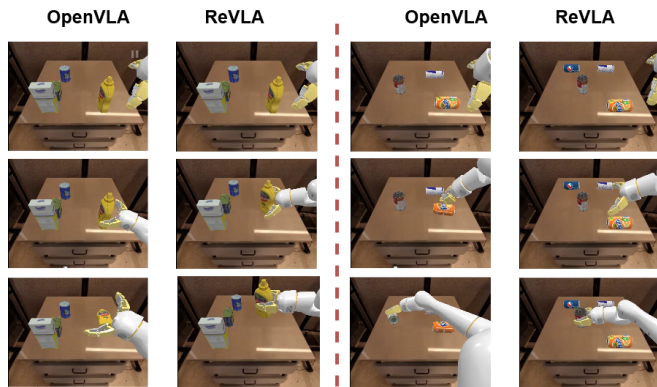


Fig. 3: Left: Openvla and ReVLA are able to identify the target object. Openvla fails to grasp it. Right: OpenVLA tries to grasp the *Fanta* can, which is nearest to the arm. ReVLA initially hesitates but eventually grasps the tomato can.

of the Dino-V2 vision encoders.

- Our ablation study on OOD Task shows that both hard and gradual flipping of the vision encoder outperform OpenVLA, with gradual flipping leading to a more robust model.

## VI. CONCLUSION AND FUTURE WORK

In this work, we evaluated and addressed the visual domain limitation in robotic generalist foundation models, specifically focusing on catastrophic forgetting in vision encoders when fine-tuning them on robotic datasets. We proposed ReVLA, a novel approach that reverts visual backbones to their pretrained states for models that require training of the whole architecture. This reversal approach successfully mitigated the catastrophic forgetting present in OpenVLA, restoring the visual generalization capabilities.

Despite these improvements, limitations remain. Our training was conducted on the Fractal dataset, which may not fully represent the diverse real-world scenarios in robotics, especially across embodiments. Additionally, an exploration of model merging techniques has the potential to yield even greater performance gains by integrating additional data from robotic datasets.

## ACKNOWLEDGMENT

This research was partially funded by the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure).

## REFERENCES

- [1] H. Touvron, *et al.*, “Llama: Open and efficient foundation language models,” *arXiv preprint arXiv:2302.13971*.
- [2] M. Oquab, *et al.*, “Dinov2: Learning robust visual features without supervision,” *arXiv preprint arXiv:2304.07193*, 2024.
- [3] A. Radford, *et al.*, “Learning transferable visual models from natural language supervision,” in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.
- [4] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, “Sigmoid loss for language image pre-training,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 11 975–11 986.
- [5] M. J. Kim, *et al.*, “Openvla: An open-source vision-language-action model,” *arXiv preprint arXiv:2406.09246*, 2024.
- [6] X. Li, *et al.*, “Evaluating real-world robot manipulation policies in simulation,” *arXiv preprint arXiv:2405.05941*, 2024.
- [7] A. Brohan, *et al.*, “Rt-1: Robotics transformer for real-world control at scale,” *arXiv preprint arXiv:2212.06817*, 2022.
- [8] Octo Model Team, *et al.*, “Octo: An open-source generalist robot policy,” in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.
- [9] S. Karamcheti, S. Nair, A. Balakrishna, P. Liang, T. Kollar, and D. Sadigh, “Prismatic vlms: Investigating the design space of visually-conditioned language models,” *arXiv preprint arXiv:2402.07865*, 2024.
- [10] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual instruction tuning,” *Advances in neural information processing systems*, vol. 36, 2024.
- [11] L. Beyer, *et al.*, “PaliGemma: A versatile 3B VLM for transfer,” *arXiv preprint arXiv:2407.07726*, 2024.
- [12] J. Li, D. Li, C. Xiong, and S. Hoi, “Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation,” in *ICML*, 2022.
- [13] Q. Zhang, J. Zhang, Y. Xu, and D. Tao, “Vision transformer with quadrangle attention,” *arXiv preprint arXiv:2303.15105*, 2023.
- [14] J.-B. Alayrac, *et al.*, “Flamingo: a visual language model for few-shot learning,” *arXiv preprint arXiv:2204.14198*, 2022.
- [15] J. Li, D. Li, S. Savarese, and S. Hoi, “Blip-2: bootstrapping language-image pre-training with frozen image encoders and large language models,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML’23, 2023.
- [16] Z. Ren, Z. Huang, Y. Wei, Y. Zhao, D. Fu, J. Feng, and X. Jin, “Pixellm: Pixel reasoning with large multimodal model,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 26 374–26 383.
- [17] A. Padalkar, *et al.*, “Open x-embodiment: Robotic learning datasets and rt-x models,” *arXiv preprint arXiv:2310.08864*, 2023.
- [18] B. Zitkovich, *et al.*, “Rt-2: Vision-language-action models transfer web knowledge to robotic control,” in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 11 2023, pp. 2165–2183.
- [19] X. Chen, *et al.*, “Pali-x: On scaling up a multilingual vision and language model,” *arXiv preprint arXiv:2305.18565*, 2023.
- [20] D. Driess, *et al.*, “Palm-e: An embodied multimodal language model,” *arXiv preprint arXiv:2303.03378*, 2023.
- [21] F. Muratore, F. Ramos, G. Turk, W. Yu, M. Gienger, and J. Peters, “Robot learning from randomized simulations: A review,” *Frontiers in Robotics and AI*, vol. 9, 2022.
- [22] F. Xiang, *et al.*, “SAPIEN: A simulated part-based interactive environment,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6 2020.
- [23] A. Szot, *et al.*, “Habitat 2.0: Training home assistants to rearrange their habitat,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
- [24] V. Makoviychuk, *et al.*, “Isaac gym: High performance gpu-based physics simulation for robot learning,” *arXiv preprint arXiv:2108.10470*, 2021.
- [25] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-real transfer in deep reinforcement learning for robotics: a survey,” in *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2020, pp. 737–744.
- [26] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, “ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [27] H.-S. Fang, C. Wang, M. Gou, and C. Lu, “Graspnet-1billion: A large-scale benchmark for general object grasping,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 444–11 453.
- [28] C. Chamzas, C. Quintero-Pena, Z. Kingston, A. Orthey, D. Rakita, M. Gleicher, M. Toussaint, and L. E. Kavraki, “Motionbenchmarker: A tool to generate and benchmark motion planning datasets,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 882–889, 2021.
- [29] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, “Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set,” *IEEE Robotics & Automation Magazine*, vol. 22, no. 3, pp. 36–52, 2015.
- [30] S. K. Ainsworth, J. Hayase, and S. Srinivasa, “Git re-basin: Merging models modulo permutation symmetries,” *arXiv preprint arXiv:2209.04836*, 2023.
- [31] E. Yang, L. Shen, G. Guo, X. Wang, X. Cao, J. Zhang, and D. Tao, “Model merging in llms, mllms, and beyond: Methods, theories, applications and opportunities,” *arXiv preprint arXiv:2408.07666*, 2024.
- [32] A. Alexandrov, V. Raychev, M. N. Müller, C. Zhang, M. Vechev, and K. Toutanova, “Mitigating catastrophic forgetting in language transfer via model merging,” *arXiv preprint arXiv:2407.08699*, 2024.
- [33] D. Arpit, H. Wang, Y. Zhou, and C. Xiong, “Ensemble of averages: Improving model selection and boosting performance in domain generalization,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 8265–8277, 2022.
- [34] Y.-L. Sung, L. Li, K. Lin, Z. Gan, M. Bansal, and L. Wang, “An empirical study of multimodal model merging,” *arXiv preprint arXiv:2304.14933*, 2023.
- [35] T. Mu, Z. Ling, F. Xiang, D. Yang, X. Li, S. Tao, Z. Huang, Z. Jia, and H. Su, “Maniskill: Generalizable manipulation skill benchmark with large-scale demonstrations,” *arXiv preprint arXiv:2107.14483*, 2021.
- [36] W. Zhao, J. P. Queralta, and T. Westerlund, “Sim-to-real transfer in deep reinforcement learning for robotics: a survey,” in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 737–744.
- [37] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, “Closing the sim-to-real loop: Adapting simulation randomization with real world experience,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8973–8979.
- [38] R. Ranftl, K. Lasinger, D. Hafner, K. Schindler, and V. Koltun, “Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer,” *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.