

Red-Teaming of Vision-Language-Action Models via Quality Diversity Environment Generation

Shihan Zhao¹, Henry Chen¹, Ya-Chuan Hsu¹, Minjune Hwang¹, Siddharth Srikanth¹², Freddie Liang¹, Akanksha Saran³, Aaquib Tabrez^{4†}, Stefanos Nikolaidis^{1†}, Daniel Seita^{1†}

¹University of Southern California ²Columbia University ³Adobe Research

⁴Cornell University [†]Equal advisement

Abstract—While Vision-Language-Action (VLA) models perform strongly on in-distribution robot control, they often fail under minor shifts in object positions, lighting, textures, and camera poses. To address this, we propose **Quality Diversity for Diverse Environment Generation (Q-DEG)**, a method for improving VLA robustness by generating adversarial environments and fine-tuning the VLA on its own successful rollouts. Q-DEG uses Quality Diversity (QD) optimization to find high-failure environments while promoting diversity with VLA policy embeddings. This yields challenging but solvable environments for fine-tuning. In simulation, experiments across three representative VLA models show that Q-DEG improves robustness to out-of-distribution (OOD) adversarial environments compared to the base VLA and VLAs fine-tuned using two red-teaming baselines. Real-world experiments further show improved robustness to object location shifts in tabletop manipulation tasks.

I. INTRODUCTION

Vision-Language-Action (VLA) models [15, 28, 76] have recently emerged as a promising mode for robot control. By leveraging Vision-Language Models (VLMs) [9, 3, 1] pretrained on internet-scale data, VLAs access rich visual and semantic knowledge and generalize across tasks and robot embodiments [63, 36, 4, 6]. However, despite strong in-distribution performance, VLAs frequently fail under peripheral test-time shifts such as object positions, colors, lighting, and camera poses.

Prior work has characterized this vulnerability through robustness benchmarks and VLA failure-mode analysis [43, 20, 13], and more recent works on robotic red-teaming have begun to probe policy weaknesses through generated visual edits [45], simulated world models [62], or adversarial instruction generation [61]. However, these approaches do not search over interpretable environment parameters that produce policy-specific failures. Another focus is fine-tuning VLAs with reinforcement learning objectives to improve downstream success [71, 44, 10]. These methods improve the VLA through action-level exploration, but do not actively target specific failure modes.

To address these limitations, we propose **Quality Diversity for Diverse Environment Generation (Q-DEG)**, which improves VLA robustness using Quality Diversity (QD) optimization [47, 16] and self-imitation [68]. Our key insight is to use QD to find environments that induce diverse pre-action-layer activations and inconsistent task success, revealing vulnerabilities while preserving successful rollouts. We then fine-tune the VLA on its own successful rollouts from these challenging environments. Like red-teaming methods,

Q-DEG stores intermediate adversarial environments in an interpretable structure for failure mode analysis. Additionally, Q-DEG requires no action-level confidence from the VLA, and applies to both autoregressive and flow-based models. Using LIBERO [41] simulation and real-world tabletop manipulation tasks, we demonstrate that Q-DEG discovers a range of such environments and improves robustness to OOD test settings.

To summarize, our contributions are: (1) Q-DEG, a method that formulates VLA environmental red-teaming as a Quality Diversity search over feasibly adversarial environments; (2) a self-imitation procedure that improves VLA robustness by fine-tuning on the VLA’s own successful rollouts in those environments; and (3) an evaluation of Q-DEG across three VLAs in simulation, showing advantage over baselines in environmental diversity, robustness improvement, and sim-to-real transfer.

II. PROBLEM FORMULATION AND METHOD

We assume a base VLA $\pi(a \mid o, c)$ with non-zero success on a task specified by language instruction c and parameterized environment $\mathcal{E}_\theta$, where $\theta \in \Theta$ denotes interpretable parameters such as object and camera poses. Q-DEG solves two problems: 1) Identify diverse perturbed parameters $\{\theta'_i\}_{i=1}^M$ that each induce a high failure rate $P^{\text{fail}}(\theta_i) = \mathbb{E}_{\zeta \sim \pi(\cdot \mid o \sim \mathcal{E}_{\theta_i}, c)}[1 - \mathbf{1}^{\text{succ}}(\zeta)]$, where $\mathbf{1}^{\text{succ}}(\zeta) \in \{0, 1\}$ is a binary task success indicator. 2) Improve π ’s success rates under perturbations in Θ . Q-DEG iterates a four-stage procedure (see Figure 1) to achieve these objectives.

A. Environment Generation

At the start of each iteration, Q-DEG generates a batch of B candidate environments from environment parameters $\{\theta_i\}_{i=1}^B$. Based on common axes of variation studied for policy robustness [7], we set object placements (xy coordinates for each movable object), the camera’s pose (xyz coordinates and orientation axis angles), the light source’s pose (xyz coordinates), table background RGB, and reflexive specular RGB as our environment parameters to perturb. Q-DEG is not tied to this specific parameterization. Different tasks or simulators can expose different controllable parameters depending on the robustness axes of interest. To ensure a minimum degree of feasibility, we define a mixed integer quadratic program (MIQP) [50] to repair object overlaps and task misalignments (e.g., if the task is to “pick up a bowl next to the plate,” there should be a bowl next to the plate) within proposed environments. This repair step ensures that high failure rates

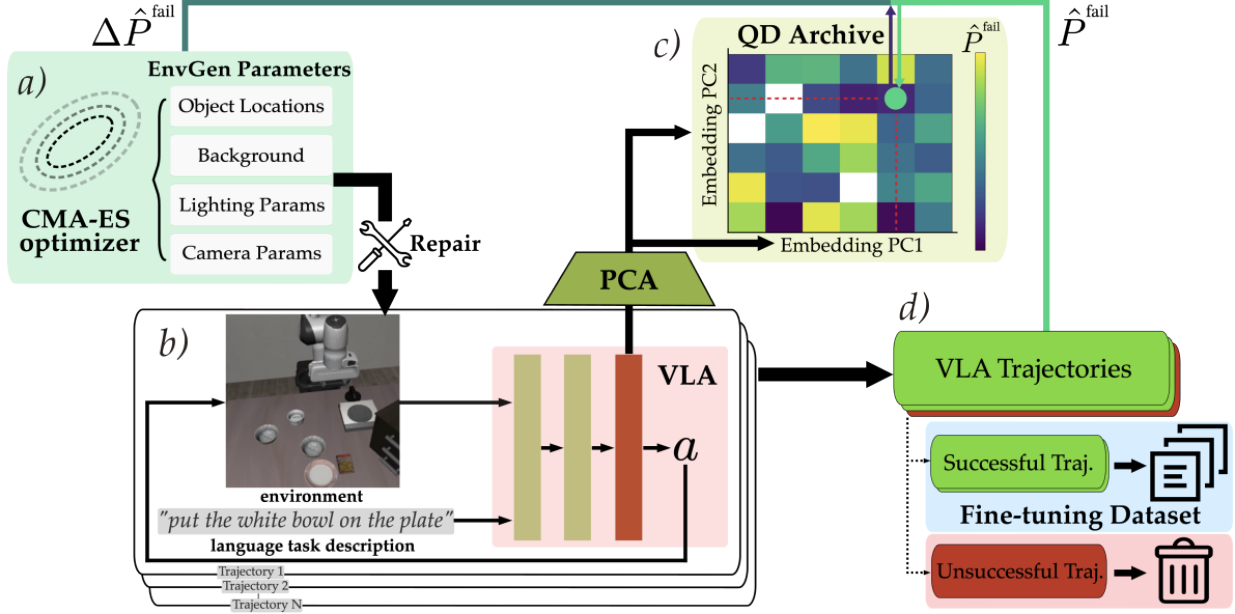


Fig. 1. **Overview of Q-DEG.** *a)* CMA-ES [25] proposes environment parameters such as object locations, background, lighting, and camera pose; invalid proposals are repaired to remove object overlaps or restore task-relevant spatial relations. *b)* The VLA is rolled out in each proposed environment to estimate $\hat{P}^{\text{fail}}$, where environments are rewarded for inducing failures while still producing at least one successful rollout. *c)* Q-DEG arranges accepted environments in a QD archive using low-dimensional PCA projections of the VLA’s policy embeddings. *d)* Successful trajectories from archived environments are collected into a fine-tuning dataset for improving the VLA’s robustness.

are more likely to reflect meaningful VLA weaknesses rather than invalid task configurations. We include more details on environment parameterization and MIQP repair in the supplementary material.

B. Feedback Metrics

On each of the B proposed environments, the base VLA π is rolled out k times to obtain two feedback metrics that inform Q-DEG’s adaptation.

Failure Rate: To evaluate an environment’s adversarial *quality* with respect to the VLA π , π ’s empirical failure rate $\hat{P}^{\text{fail}}$ on the environment is estimated over rollouts $\{\zeta^j\}_{j=1}^k$. Q-DEG prioritizes environments that are feasible enough to yield successful rollouts for later self-imitation, so we modify $\hat{P}^{\text{fail}}$ with a condition that hard-sets to 0 if all rollouts fail, as defined in Eq. 1.

$$\hat{P}^{\text{fail}}(\theta) = \begin{cases} 0 & \text{if } \bigwedge_{j=1}^k \neg \mathbf{1}^{\text{succ}}(\zeta^j) \\ 1 - \sum_{j=1}^k \mathbf{1}^{\text{succ}}(\zeta^j)/k & \text{otherwise.} \end{cases} \quad (1)$$

Policy Embeddings: For evaluating an environment’s *diversity*, we use π ’s pre-action layer activations at the start of each rollout, which we refer to as *policy embeddings*. Intuitively, policy embeddings represent the VLA’s “first impression” on each environment, and our goal is to diversify policy embeddings so that our adversarial environments induce diverse experience on the VLA. Inspired by prior work in multitask safety analysis of VLAs using VLA embeddings [23], we typically choose the final hidden layer before the VLA’s action decoder as the source of our policy embeddings, and average across rollouts to obtain a set of embeddings ψ for each environment. We project raw policy embeddings into a low-dimensional space using linear Principal Component Analysis (PCA) before QD search, since hidden layer activations

are high-dimensional and discretization-based QD algorithms suffer from the curse of dimensionality [53, 27, 64]. We denote the projected embedding with $A\psi + b$, with A being the principal component projection matrix and b being the centering offset.

C. Quality Diversity (QD) Optimization

QD searches for a collection of diverse and high-quality solutions within the space spanned by some measure functions. Q-DEG applies QD to environment parameters θ , using the failure rate $\hat{P}^{\text{fail}}$ as the quality metric and the VLA’s policy embeddings as the measure functions. Formally, the QD objective is to find an archive $\mathcal{A}$ of M elite environments $\{\mathcal{E}_{\theta_j}\}_{j=1}^M$, where each elite occupies a distinct region of the policy embedding space, and solve $\max_{\theta_1 \dots \theta_M} \sum_{j=1}^M \hat{P}^{\text{fail}}(\theta_j)$. Since $\hat{P}^{\text{fail}}$ is empirically estimated and thus discrete, we use CMA-MAE, a QD algorithm designed for discrete quality metrics [16, 74]. CMA-MAE uses a black-box CMA-ES optimizer to propose candidates, evaluate their qualities and measures, and bias future proposals toward high quality or novel measures. Its core components are the soft QD archive and CMA-ES feedback mechanism.

Soft QD Archive: CMA-MAE maintains an archive $\mathcal{A}$ to store adversarial environments and provide improvement scores and restart checkpoints to CMA-ES when it stagnates. To construct $\mathcal{A}$, the space spanned by policy embeddings’ top principal components is tessellated into a grid, with cell identifier $\mathcal{T}$ mapping projected embeddings to cell indices [47]. Each candidate parameter θ_i is then considered for insertion into its mapped cell $i = \mathcal{T}(A\psi_i + b)$. Each cell maintains a threshold quality value t_{cell} initialized to 0 and annealed upward as high quality metrics are observed within the cell. If $\hat{P}_i^{\text{fail}} > t_{\text{cell},i}$, then θ_i is inserted into cell i and $t_{\text{cell},i}$ is softly updated with

$t_{\text{cell}_i} \leftarrow (1 - \alpha)t_{\text{cell}_i} + \alpha\hat{P}_i^{\text{fail}}$, otherwise θ_i is discarded and no update is performed. Intuitively, by delaying each individual update with an annealing rate hyperparameter $\alpha \in [0, 1]$, t_{cell_i} captures the density of high-failure-rate environments within cell_i ’s embedding space region.

CMA-ES Adaptation: CMA-ES optimizer maintains an internal multivariate Gaussian distribution on environment parameters [25], from which it samples a batch of candidate parameters $\{\theta_i\}_{i=1}^B$ to be evaluated and inserted into $\mathcal{A}$. After insertion, CMA-ES adapts its internal distribution according to each candidate’s failure rate improvement score $\Delta\hat{P}^{\text{fail}}$. The improvement score for a candidate θ_i is defined as $\Delta\hat{P}^{\text{fail}}(\theta_i) = \hat{P}^{\text{fail}}(\theta_i) - t_{\mathcal{T}(A\psi_i+b)}$, where $t_{\mathcal{T}(A\psi_i+b)}$ is the threshold quality value in the cell assigned to θ_i at insertion time. Intuitively, a candidate’s improvement score is high when it induces high failure rate or when its policy embeddings map to a previously under-explored cell, so optimizing improvement scores simultaneously motivates more adversarial and more diverse environments. The batch of candidates is ranked by their improvement scores, and the ranking is returned as feedback to CMA-ES, which adapts to propose candidates with higher improvement scores in future iterations. When CMA-ES stagnates, we restart it by setting the mean of its internal distribution to an elite θ sampled from the archive, resuming the search from a “stepping stone” environment known to be adversarial and/or diverse.

D. Self-Imitation Fine-Tuning

In addition to generating diverse adversarial environments, Q-DEG also improves robustness of the base VLA π against similar failure modes. Q-DEG achieves this by fine-tuning π to imitate its own successful rollouts on all *feasibly* adversarial environments collected during QD optimization. To construct the fine-tuning dataset $\mathcal{D}$, we retain the shortest successful trajectory among k rollouts on each adversarial environment. We keep only the shortest successful trajectory per environment, since multiple demonstrations for the same environment can cause action divergence and undermine imitation learning [2]. Additionally, to prevent π from forgetting current skills, we add to $\mathcal{D}$ a small number $|\mathcal{D}_{\text{base}}|$ of trajectories from the task-specific dataset on which the base VLA was trained [42]. We then fine-tune π on $\mathcal{D}$ following a simple behavior cloning objective. Incidentally, the diversity among environments found by Q-DEG promotes diversity within $\mathcal{D}$ ’s trajectories, preventing behavior collapse during self-imitation [68].

III. EXPERIMENTS

Our experiments are designed to answer the following questions: 1) **Red-teaming:** Can Q-DEG generate diverse adversarial environments? 2) **Robustness:** Does Q-DEG improve the base VLA’s robustness against OOD adversarial environments? 3) **Sim-to-real:** Does the improved robustness resulting from Q-DEG facilitate VLA sim-to-real transfer?

A. Simulation Experiments

We address the first two questions in simulation using LIBERO-Spatial [41], which consists of 10 language-

conditioned pick-and-place tasks and supports our environmental parameterization. To demonstrate Q-DEG’s application toward multiple VLA architectures, we experiment using two autoregressive models, π_0 -FAST [55] and OpenVLA-OFT [36, 37], and one flow-matching model, $\pi_{0.5}$ [31].

Baselines: Due to the lack of prior work that red-team VLAs by perturbing simulation environment parameters, we implement two environmental red-teaming baselines to ablate the effect of QD optimization employed by Q-DEG: *randomization* and *CMA-ES*. Both baselines generate adversarial environments by perturbing the same environment parameters as Q-DEG; as with Q-DEG, all generated environment parameters undergo MIQP repair prior to rollout.

Randomization. This baseline adds fixed Gaussian noise to the default LIBERO-Spatial environment parameters. *Randomization* neither maintains a QD archive nor adapts its noise distribution.

CMA-ES. The second baseline, *CMA-ES*, adapts the parameter noise distribution with a CMA-ES optimizer, but the optimizer disregards policy embeddings and only optimizes for failure rates, which makes the baseline prioritize adversarial search while neglecting environmental diversity.

Evaluating red-teaming: We compare methods under the same VLA rollout budget (3200 rollouts) using two metrics: 1) the *number of feasibly adversarial environments* and 2) the *average pairwise DINOv2 embedding distance* to measure visual diversity across environments. For the average embedding distance among a set of environment parameters, we generate an environment from each parameter (see Sec. II-A), pass the first-step overview image through `dinov2_vitb14`, and compute average pairwise cosine distance among embeddings. We use DINOv2 because its embedding distances correlate with visual-semantic similarity [52].

Evaluating robustness: Following Sec. II-D, we fine-tune the base VLA with successful trajectories collected during each baseline’s rollouts. We set aside 20% of adversarial environments from each baseline as the evaluation set and exclude their trajectories from fine-tuning. On each environment in the evaluation set, we evaluate success rates of the base VLA as well as the post-fine-tune VLAs from each baseline and Q-DEG over 4 trials. We then compare the VLAs’ average success rates on the evaluation set.

B. Real-World Experiments

To investigate whether improved VLA robustness leads to better sim-to-real transfer, we construct a real-world test where the VLA controls a Gen-2 Kinova JACO arm to either *grab the bowl from table center* (Task 1), or *grab the bowl next to the plate* (Task 2). We modify LIBERO to create simulated digital twins for both tasks, and collect 40 simulated demonstrations for each. To obtain a base VLA with reasonable success rates, we fine-tune the `pi05_base` checkpoint from $\pi_{0.5}$ with LoRA (rank 16) SFT for 10000 steps on collected demonstrations. We then improve the base VLA’s robustness in simulation by running Q-DEG for 832 rollouts and fine-tuning another 10000 steps on trajectories collected by Q-

TABLE I

FINE-TUNED VLA SUCCESS RATES ON OOD ADVERSARIAL ENVIRONMENTS FOUND BY EACH METHOD. ROWS CORRESPOND TO THE SOURCE OF THE EVALUATION SET; COLUMNS CORRESPOND TO THE BASE VLA AND VLAS FINE-TUNED ON TRAJECTORIES COLLECTED BY *randomization* (RAND), *CMA-ES*, AND *Q-DEG*.

Eval. set	π_0 -FAST				OpenVLA-OFT				$\pi_{0.5}$			
	base	rand	CMA-ES	Q-DEG	base	rand	CMA-ES	Q-DEG	base	rand	CMA-ES	Q-DEG
rand	29.15	39.40	37.79	48.73	56.25	82.29	68.75	78.13	77.38	77.98	83.33	88.09
CMA-ES	27.90	38.46	24.81	29.82	45.75	52.52	48.27	75.79	54.98	49.39	69.17	58.00
Q-DEG	54.83	51.28	36.89	56.13	47.19	50.77	55.61	69.13	38.03	61.82	70.45	81.81
average	37.29	43.05	42.51	44.89	49.73	61.86	57.54	74.35	56.80	63.06	74.32	75.97

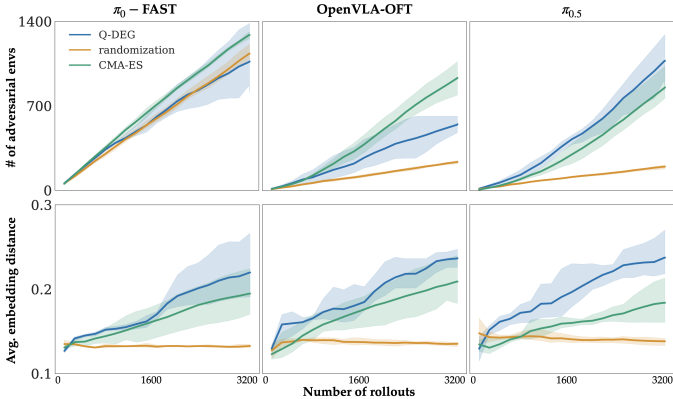


Fig. 2. Number of feasibly adversarial environments (top) found by each of *randomization*, *CMA-ES*, and *Q-DEG*, and the corresponding average pairwise DINOv2 cosine distance among environments (bottom), plotted against the number of VLA rollouts. Shaded regions denote 95% confidence intervals aggregated over three independent trials.

DEG. For a fair comparison, the base VLA is also fine-tuned for another 10000 steps on demonstrations. We then evaluate the base VLA and the robust VLA from Q-DEG on both real-world tasks for 10 trials each. A trial is successful if the robot navigates its gripper around the rim of the correct bowl within 3 minutes. We interleave trials between the two tested VLAs to control biases from external factors [38].

IV. RESULTS

Red-teaming: As shown in Figure 2, all methods find hundreds of feasibly adversarial environments where the VLAs fail, revealing their vulnerabilities to our environmental perturbations despite their near-perfect performance on vanilla LIBERO-Spatial. Since OpenVLA-OFT and $\pi_{0.5}$ have strong base performance, finding environments that induce failure is more challenging. The *randomization* baseline significantly lags behind *CMA-ES* and *Q-DEG* due to its lack of noise adaptation. Although *Q-DEG* does not find more adversarial environments than *CMA-ES*, it finds more diverse ones, as reflected by higher average DINOv2 embedding distances. By comparison, environments found by *CMA-ES* tend to cluster around a few configurations that induce the most VLA failures. The advantage of *Q-DEG* is clearest on $\pi_{0.5}$, where higher diversity yields more adversarial environments than even *CMA-ES*. This suggests that diverse environments act as “stepping stones” toward niche configurations, as outlined in Sec II-C. Example adversarial environments found by *Q-DEG* on $\pi_{0.5}$ appear in the appendix.

Robustness: Table I reports success rates on OOD adversarial environments generated by the baselines and *Q-DEG*. The rows correspond to the evaluation set source, and the columns indicate the base VLA and VLAs fine-tuned on trajectories collected by each method. Across three VLA models, *Q-DEG* consistently improves over the base VLA on all evaluation sets, achieving improvements of up to 25%. The baseline methods achieve low success rates on environments generated by other methods, suggesting overfitting to their own failure modes despite excluding evaluation-set trajectories during fine-tuning. In contrast, *Q-DEG* generalizes well across evaluation sets, often matching or exceeding success rates on its own evaluation set. This generalization can be attributed to the higher diversity of *Q-DEG*’s adversarial environments, and thereby its fine-tuning data.

A. Real-World Results

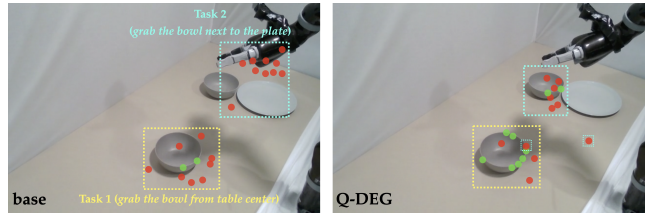


Fig. 3. JACO gripper positions at the end of each trial when evaluating the base VLA (left) and the VLA fine-tuned with *Q-DEG* (right) in the real world. Successful trials are shown in green, and failed trials are shown in red.

Sim-to-real: To visualize our real-world experiments, Figure 3 annotates the final JACO gripper position for each trial, with successful trials in green and failed trials in red. The base VLA (left) succeeds 2/10 on Task 1 and 0/10 on Task 2, whereas the VLA fine-tuned using trajectories from *Q-DEG* (right) succeeds 7/10 on Task 1 and 2/10 on Task 2, demonstrating better sim-to-real transfer with *Q-DEG*. We observe that, on Task 2, the base VLA frequently moves the gripper toward the empty space to the upper-right of the plate, corresponding to the bowl location used during the collection of training demonstrations in simulation. In contrast, *Q-DEG* improves the VLA’s robustness to object location perturbations, so its fine-tuned VLA more consistently navigates the grippers toward the bowl. We additionally observe that, on Task 1, the base VLA seems to only be capable of grasping the bowl from a single side of the rim, whereas the VLA fine-tuned with *Q-DEG* exhibits more diverse grasping poses from multiple sides, further suggesting a more robust grasping policy.

REFERENCES

- [1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025.
- [2] Suneel Belkhal, Yuchen Cui, and Dorsa Sadigh. Data quality in imitation learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.
- [3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschanen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bošnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier Henaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.
- [4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi 0$: A Vision-Language-Action Flow Model for General Robot Control. In *Robotics: Science and Systems (RSS)*, 2025.
- [5] Herbie Bradley, Andrew Dai, Hannah Teufel, Jenny Zhang, Koen Oostermeijer, Marco Bellagente, Jeff Clune, Kenneth Stanley, Grégory Schott, and Joel Lehman. Quality-Diversity Through AI Feedback. In *International Conference on Learning Representations (ICLR)*, 2024.
- [6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning (CoRL)*, 2023.
- [7] Jason Chen, I-Chun Arthur Liu, Gaurav Sukhatme, and Daniel Seita. CRAFT: Video Diffusion for Bimanual Robot Data Generation. *arXiv preprint arXiv:2604.03552*, 2026.
- [8] William Chen, Jagdeep Singh Bhatia, Catherine Glossop, Nikhil Mathihalli, Ria Doshi, Andy Tang, Danny Driess, Karl Pertsch, and Sergey Levine. Steerable Vision-Language-Action Policies for Embodied Reasoning and Hierarchical Control. *arXiv preprint arXiv:2602.13193*, 2026.
- [9] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A Jointly-Scaled Multilingual Language-Image Model. In *International Conference on Learning Representations (ICLR)*, 2023.
- [10] Yuhui Chen, Shuai Tian, Shugao Liu, Yingting Zhou, Haoran Li, and Dongbin Zhao. Conrft: A reinforced fine-tuning method for vla models via consistency policy. *arXiv preprint arXiv:2502.05450*, 2025.
- [11] Open X-Embodiment Collaboration, Abby O’Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, Albert Tung, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anchit Gupta, Andrew Wang, Andrey Kolobov, Anikait Singh, Animesh Garg, Aniruddha Kembhavi, Annie Xie, Anthony Brohan, Antonin Raffin, Archit Sharma, Arefeh Yavary, Arhan Jain, Ashwin Balakrishna, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Blake Wulfe, Brian Ichter, Cewu Lu, Charles Xu, Charlotte Le, Chelsea Finn, Chen Wang, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Christopher Agia, Chuer Pan, Chuyuan Fu, Coline Devin, Danfei Xu, Daniel Morton, Danny Driess, Daphne Chen, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dinesh Jayaraman, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Ethan Foster, Fangchen Liu, Federico Ceola, Fei Xia, Feiyu Zhao, Felipe Vieira Frujeri, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Gilbert Feng, Giulio Schiavi, Glen Berseth, Gregory Kahn, Guangwen Yang, Guanzhi Wang, Hao Su, Hao-Shu Fang, Haochen Shi, Henghui Bao, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homanga Bharadhwaj, Homer Walke, Hongjie Fang, Huy Ha, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jad Abou-Chakra, Jaehyung Kim, Jaimyn Drake, Jan Peters,

- Jan Schneider, Jasmine Hsu, Jay Vakil, Jeannette Bohg, Jeffrey Bingham, Jeffrey Wu, Jensen Gao, Jiaheng Hu, Jiajun Wu, Jialin Wu, Jiankai Sun, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jimmy Wu, Jingpei Lu, Jingyun Yang, Jitendra Malik, João Silvério, Joey Hejna, Jonathan Boohar, Jonathan Tompson, Jonathan Yang, Jordi Salvador, Joseph J. Lim, Junhyek Han, Kaiyuan Wang, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Black, Kevin Lin, Kevin Zhang, Kiana Ehsani, Kiran Lekkala, Kirsty Ellis, Krishan Rana, Krishnan Srinivasan, Kuan Fang, Kunal Pratap Singh, Kuo-Hao Zeng, Kyle Hatch, Kyle Hsu, Laurent Itti, Lawrence Yunliang Chen, Lerrel Pinto, Li Fei-Fei, Liam Tan, Linxi "Jim" Fan, Lionel Ott, Lisa Lee, Luca Weihs, Magnum Chen, Marion Lepert, Marius Memmel, Masayoshi Tomizuka, Masha Itkina, Mateo Guaman Castro, Max Spero, Maximilian Du, Michael Ahn, Michael C. Yip, Mingtong Zhang, Mingyu Ding, Minhho Heo, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Muhammad Zubair Irshad, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Ning Liu, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Osbert Bastani, Pannag R Sanketi, Patrick "Tree" Miller, Patrick Yin, Paul Wohlhart, Peng Xu, Peter David Fagan, Peter Mitrano, Pierre Sermanet, Pieter Abbeel, Priya Sundareshan, Qiuyu Chen, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Mart'in-Mart'in, Rohan Bajjal, Rosario Scalise, Rose Hendrix, Roy Lin, Runjia Qian, Ruohan Zhang, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Shan Lin, Sherry Moore, Shikhar Bahl, Shivin Dass, Shubham Sonawani, Shubham Tulsiani, Shuran Song, Sichun Xu, Siddhant Halder, Siddharth Karamcheti, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Subramanian Ramamoorthy, Sudeep Dasari, Suneel Belkhale, Sungjae Park, Suraj Nair, Suvir Mirchandani, Takayuki Osa, Tanmay Gupta, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Thomas Kollar, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Trinity Chung, Vidhi Jain, Vikash Kumar, Vincent Vanhoucke, Vitor Guizilini, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiangyu Chen, Xiaolong Wang, Xinghao Zhu, Xinyang Geng, Xiyuan Liu, Xu Liangwei, Xuanlin Li, Yansong Pang, Yao Lu, Yecheng Jason Ma, Yejin Kim, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Yilin Wu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yongqiang Dou, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yue Cao, Yueh-Hua Wu, Yujin Tang, Yuke Zhu, Yunchu Zhang, Yunfan Jiang, Yunshuang Li, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zehan Ma, Zhuo Xu, Zichen Jeff Cui, Zichen Zhang, Zipeng Fu, and Zipeng Lin. Open X-Embodiment: Robotic Learning Datasets and RT-X Models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
- [12] Haoquan Fang, Jiafei Duan, Donovan Clay, Sam Wang, Shuo Liu, Weikai Huang, Xiang Fan, Wei-Chuan Tsai, Shirui Chen, Yi Ru Wang, Shanli Xing, Jaemin Cho, Jae Sung Park, Ainaz Eftekhari, Peter Sushko, Karen Farley, Angad Wadhwa, Cole Harrison, Winson Han, Ying-Chun Lee, Eli VanderBilt, Rose Hendrix, Suveen Ellawela, Lucas Ngoo, Joyce Chai, Zhongzheng Ren, Ali Farhadi, Dieter Fox, and Ranjay Krishna. MolmoAct2: Action Reasoning Models for Real-world Deployment. *arXiv preprint arXiv:2605.02881*, 2026.
- [13] Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, Shiduo Zhang, Zhaoye Fei, Jinlan Fu, Jingjing Gong, and Xipeng Qiu. LIBERO-Plus: In-depth Robustness Analysis of Vision-Language-Action Models. *arXiv preprint arXiv:2510.13626*, 2025.
- [14] Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*, 2023.
- [15] Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, Brian Ichter, Danny Driess, Jiajun Wu, Cewu Lu, and Mac Schwager. Foundation models in robotics: Applications, challenges, and the future. In *International Journal of Robotics Research (IJRR)*, volume 44, pages 701–739, 2025.
- [16] Matthew Fontaine and Stefanos Nikolaidis. Covariance matrix adaptation map-annealing. In *Conference on Genetic and Evolutionary Computation (GECCO)*, 2023.
- [17] Matthew C. Fontaine and Stefanos Nikolaidis. A Quality Diversity Approach to Automatically Generating Human-Robot Interaction Scenarios in Shared Autonomy. In *Robotics: Science and Systems (RSS)*, 2021.
- [18] Matthew C Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K Hoover. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the 2020 genetic and evolutionary computation conference*, pages 94–102, 2020.
- [19] Matthew C. Fontaine, Ya-Chuan Hsu, Yulun Zhang, Bryon Tjanaka, and Stefanos Nikolaidis. On the importance of environments in human-robot coordination. In *Robotics: Science and Systems (RSS)*, 2021.
- [20] Jensen Gao, Suneel Belkhale, Sudeep Dasari, Ashwin Balakrishna, Dhruv Shah, and Dorsa Sadigh. A taxonomy for evaluating generalist robot policies. *arXiv preprint arXiv:2503.01238*, 2025.
- [21] Divyam Goel, Yufei Wang, Tiancheng Wu, Helen Qiao, Pavel Piliptchak, David Held, and Zackory Erickson. Geometric Red-Teaming for Robotic Manipulation. In *Conference on Robot Learning (CoRL)*, 2025.
- [22] Luca Grillotti and Antoine Cully. Unsupervised behavior discovery with quality-diversity optimization. *IEEE Transactions on Evolutionary Computation*, 26(6):1539–

- 1552, 2022.
- [23] Q Gu, Y Ju, S Sun, I Gilitschenski, H Nishimura, M Itkina, and F Shkurti. SAFE: Multitask Failure Detection for Vision-Language-Action Models. In *Neural Information Processing Systems (NeurIPS)*, 2025.
 - [24] Asher J. Hancock, Allen Z. Ren, and Anirudha Majumdar. Run-time observation interventions make vision-language-action models more visually robust. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.
 - [25] Nikolaus Hansen. The CMA Evolution Strategy: A Tutorial. *arXiv preprint arXiv:1604.00772*, 2023.
 - [26] Saeed Hedayatian and Stefanos Nikolaidis. Autoqd: Automatic discovery of diverse behaviors with quality-diversity optimization. *arXiv preprint arXiv:2506.05634*, 2025.
 - [27] Saeed Hedayatian and Stefanos Nikolaidis. Soft Quality-Diversity Optimization. In *International Conference on Learning Representations (ICLR)*, 2026.
 - [28] Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, Shibo Zhao, Shayegan Omidshafiei, Dong-Ki Kim, Ali akbar Aghamohammadi, Katia Sycara, Matthew Johnson-Roberson, Dhruv Batra, Xiaolong Wang, Sebastian Scherer, Chen Wang, Zsolt Kira, Fei Xia, and Yonatan Bisk. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023.
 - [29] Yixuan Huang, Bowen Li, Vaibhav Saxena, Yichao Liang, Utkarsh Aashu Mishra, Liang Ji, Lihan Zha, Jimmy Wu, Nishanth Kumar, Sebastian Scherer, Danfei Xu, and Tom Silver. Kinder: A physical reasoning benchmark for robot learning and planning. In *Robotics: Science and Systems (RSS)*, 2026.
 - [30] Physical Intelligence, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Kevin Black, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachy Groom, Hunter Hancock, Karol Hausman, Gashon Hussein, Brian Ichter, Szymon Jakubczak, Rowan Jen, Tim Jones, Ben Katz, Liyiming Ke, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Yao Lu, Vishnu Mano, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Charvi Sharma, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Alex Swerdlow, James Tanner, Marcel Torne, Quan Vuong, Anna Walling, Haohuan Wang, Blake Williams, Sukwon Yoo, Lili Yu, Ury Zhilinsky, and Zhiyuan Zhou. $\pi_{0,6}^*$: a VLA That Learns From Experience. *arXiv preprint arXiv:2511.14759*, 2025.
 - [31] Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0,5}$: a Vision-Language-Action Model with Open-World Generalization. In *Conference on Robot Learning (CoRL)*, 2025.
 - [32] Physical Intelligence, Bo Ai, Ali Amin, Raichelle Aniceto, Ashwin Balakrishna, Greg Balke, Kevin Black, George Bokinsky, Shihao Cao, Thomas Charbonnier, Vedant Choudhary, Foster Collins, Ken Conley, Grace Connors, James Darpinian, Karan Dhabalia, Maitrayee Dhaka, Jared DiCarlo, Danny Driess, Michael Equi, Adnan Esmail, Yunhao Fang, Chelsea Finn, Catherine Glossop, Thomas Godden, Ivan Goryachev, Lachlan Groom, Haroun Habeeb, Hunter Hancock, Karol Hausman, Gashon Hussein, Victor Hwang, Brian Ichter, Connor Jacobsen, Szymon Jakubczak, Rowan Jen, Tim Jones, Gregg Kammerer, Ben Katz, Liyiming Ke, Mairbek Khadikov, Chandra Kuchi, Marinda Lamb, Devin LeBlanc, Brendon LeCount, Sergey Levine, Xinyu Li, Adrian Li-Bell, Vladislav Lialin, Zhonglin Liang, Wallace Lim, Yao Lu, Enyu Luo, Vishnu Mano, Nandan Marwaha, Aikys Mongush, Liam Murphy, Suraj Nair, Tyler Patterson, Karl Pertsch, Allen Z. Ren, Gavin Schelske, Charvi Sharma, Baifeng Shi, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, Will Stoeckle, Jiaming Tang, Jimmy Tanner, Shalom Tekeste, Marcel Torne, Kyle Vedder, Quan Vuong, Anna Walling, Haohuan Wang, Jason Wang, XuDong Wang, Chris Whalen, Samuel Whitmore, Blake Williams, Charles Xu, Sukwon Yoo, Lili Yu, Wuming Zhang, Zhuoyang Zhang, and Ury Zhilinsky. $\pi_{0,7}$: a Steerable Generalist Robotic Foundation Model with Emergent Capabilities. *arXiv preprint arXiv:2604.15483*, 2026.
 - [33] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay. In *International Conference on Machine Learning (ICML)*, 2021.
 - [34] Pushkal Katara, Zhou Xian, and Katerina Fragkiadaki. Gen2Sim: Scaling up Robot Learning in Simulation with Generative Models. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
 - [35] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pan-

- nag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, Vitor Guizilini, David Antonio Herrera, Minh Heo, Kyle Hsu, Jiaheng Hu, Muhammad Zubair Irshad, Donovan Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeanette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset. In *Robotics: Science and Systems (RSS)*, 2024.
- [36] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.
- [37] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-Tuning Vision-Language-Action Models: Optimizing Speed and Success. In *Robotics: Science and Systems (RSS)*, 2025.
- [38] Hadas Kress-Gazit, Kunimatsu Hashimoto, Naveen Kuppuswamy, Paarth Shah, Phoebe Horgan, Gordon Richardson, Siyuan Feng, and Benjamin Burchfiel. Robot Learning as an Empirical Science: Best Practices for Policy Evaluation. *arXiv preprint arXiv:2409.09491*, 2024.
- [39] Jason Lee, Jiafei Duan, Haoquan Fang, Yuquan Deng, Shuo Liu, Boyang Li, Bohan Fang, Jieyu Zhang, Yi Ru Wang, Sangho Lee, Winson Han, Wilbert Pumacay, Angelica Wu, Rose Hendrix, Karen Farley, Eli VanderBilt, Ali Farhadi, Dieter Fox, and Ranjay Krishna. Molmoact: Action reasoning models that can reason in space. *arXiv preprint arXiv:2508.07917*, 2025.
- [40] Bryan Lim, Manon Flageat, and Antoine Cully. Large language models as in-context ai generators for quality-diversity. *arXiv preprint arXiv:2404.15794*, 2024.
- [41] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Neural Information Processing Systems (NeurIPS)*, 2023.
- [42] Huihan Liu, Changyeon Kim, Bo Liu, Minghuan Liu, and Yuke Zhu. Pretrained vision-language-action models are surprisingly resistant to forgetting in continual learning. *arXiv preprint arXiv:2603.03818*, 2026.
- [43] Jijia Liu, Feng Gao, Bingwen Wei, Xinlei Chen, Qingmin Liao, Yi Wu, Chao Yu, and Yu Wang. What can rl bring to vla generalization? an empirical study. *arXiv preprint arXiv:2505.19789*, 2025.
- [44] Guanxing Lu, Wenkai Guo, Chubin Zhang, Yuheng Zhou, Haonan Jiang, Zifeng Gao, Yansong Tang, and Ziwei Wang. Vla-rl: Towards masterful and general robotic manipulation with scalable reinforcement learning. *arXiv preprint arXiv:2505.18719*, 2025.
- [45] Anirudha Majumdar, Mohit Sharma, Dmitry Kalashnikov, Sumeet Singh, Pierre Sermanet, and Vikas Sindhwani. Predictive red teaming: Breaking policies without breaking robots. *arXiv preprint arXiv:2502.06575*, 2025.
- [46] Yifei Ming, Ziyang Cai, Jiuxiang Gu, Yiyun Sun, Wei Li, and Yixuan Li. Delving into out-of-distribution detection with vision-language representations. *Advances in neural information processing systems*, 35:35087–35102, 2022.
- [47] Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.
- [48] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024.
- [49] Soroush Nasiriany, Sepehr Nasiriany, Abhiram Maddukuri, and Yuke Zhu. Robocasa365: A large-scale simulation framework for training and benchmarking generalist robots. In *International Conference on Learning Representations (ICLR)*, 2026.
- [50] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.
- [51] NVIDIA, Johan Bjorck, Nikita Cherniadev, Fernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llonet, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.
- [52] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervi-

- sion. *arXiv preprint arXiv:2304.07193*, 2023.
- [53] Giuseppe Paolo, Alban Laflaquiere, Alexandre Coninx, and Stephane Doncieux. Unsupervised learning and exploration of reachable outcome space. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.
 - [54] Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In *International Conference on Machine Learning (ICML)*, 2022.
 - [55] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient Action Tokenization for Vision-Language-Action Models. In *Robotics: Science and Systems (RSS)*, 2025.
 - [56] Justin K Pugh, Lisa B Soros, Paul A Szerlip, and Kenneth O Stanley. Confronting the challenge of quality diversity. In *Conference on Genetic and Evolutionary Computation (GECCO)*, 2015.
 - [57] Mikayel Samvelyan, Sharath Chandra Raparthy, Andrei Lupu, Eric Hambro, Aram H. Markosyan, Manish Bhatt, Yuning Mao, Minqi Jiang, Jack Parker-Holder, Jakob Foerster, Tim Rocktäschel, and Roberta Raileanu. Rainbow Teaming: Open-Ended Generation of Diverse Adversarial Prompts. In *Neural Information Processing Systems (NeurIPS)*, 2024.
 - [58] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmoVLA: A Vision-Language-Action Model for Affordable and Efficient Robotics. *arXiv preprint arXiv:2506.01844*, 2025.
 - [59] Rohan Sinha, Amine Elhafsi, Christopher Agia, Matthew Foutter, Edward Schmerling, and Marco Pavone. Real-time anomaly detection and reactive planning with large language models. *arXiv preprint arXiv:2407.08735*, 2024.
 - [60] Siddharth Srikanth, Varun Bhatt, Boshen Zhang, Werner Hager, Charles Michael Lewis, Katia P Sycara, Aaqib Tabrez, and Stefanos Nikolaidis. Algorithmic prompt generation for diverse human-like teaming and communication with large language models. *arXiv preprint arXiv:2504.03991*, 2025.
 - [61] Siddharth Srikanth, Freddie Liang, Ya-Chuan Hsu, Varun Bhatt, Shihan Zhao, Henry Chen, Bryon Tjanaka, Min-june Hwang, Akanksha Saran, Daniel Seita, Aaqib Tabrez, and Stefanos Nikolaidis. Red-Teaming Vision-Language-Action Models via Quality Diversity Prompt Generation for Robust Robot Policies. *arXiv preprint arXiv:2603.12510*, 2026.
 - [62] Gemini Robotics Team, Krzysztof Choromanski, Coline Devin, Yilun Du, Debidatta Dwibedi, Ruiqi Gao, Abhishek Jindal, Thomas Kipf, Sean Kirmani, Isabel Leal, Fangchen Liu, Anirudha Majumdar, Andrew Marmon, Carolina Parada, Yulia Rubanova, Dhruv Shah, Vikas Sindhwani, Jie Tan, Fei Xia, Ted Xiao, Sherry Yang, Wenhao Yu, and Allan Zhou. Evaluating Gemini Robotics Policies in a Veo World Simulator. *arXiv preprint arXiv:2512.10675*, 2026.
 - [63] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sankeeti, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024.
 - [64] Bryon Tjanaka, Henry Chen, Matthew C Fontaine, and Stefanos Nikolaidis. Discount Model Search for Quality Diversity Optimization in High-Dimensional Measure Spaces. In *International Conference on Learning Representations (ICLR)*, 2026.
 - [65] J. Wang, M. Leonard, K. Daniilidis, D. Jayaraman, and E. S. Hu. Evaluating pi0 in the Wild: Strengths, Problems, and the Future of Generalist Robot Policies, 2025.
 - [66] Lirui Wang, Yiyang Ling, Zhecheng Yuan, Mohit Shridhar, Chen Bao, Yuzhe Qin, Bailin Wang, Huazhe Xu, and Xiaolong Wang. GenSim: Generating Robotic Simulation Tasks via Large Language Models. In *International Conference on Learning Representations (ICLR)*, 2024.
 - [67] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. RoboGen: Towards Unleashing Infinite Data for Automated Robot Learning via Generative Simulation. In *International Conference on Machine Learning (ICML)*, 2024.
 - [68] Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.
 - [69] Michał Zawalski, William Chen, Karl Pertsch, Oier Mees, Chelsea Finn, and Sergey Levine. Robotic control via embodied chain-of-thought reasoning. In *Conference on Robot Learning (CoRL)*, 2024.
 - [70] Shiduo Zhang, Zhe Xu, Peiju Liu, Xiaopeng Yu, Yuan Li, Qinghui Gao, Zhaoye Fei, Zhangyue Yin, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. VLABench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
 - [71] Zijian Zhang, Kaiyuan Zheng, Zhaorun Chen, Joel Jang, Yi Li, Siwei Han, Chaoqi Wang, Mingyu Ding, Dieter Fox, and Huaxiu Yao. Grape: Generalizing robot policy via preference alignment. *arXiv preprint arXiv:2411.19309*, 2024.
 - [72] Zongzheng Zhang, Jingrui Pang, Zhuo Yang, Kun Li, Minwen Liao, Saining Zhang, Guoxuan Chi, Jinbang

- Guo, Huan ang Gao, Modi Shi, Dongyun Ge, Yao Mu, Jiayuan Gu, Rui Chen, Hao Dong, Huazhe Xu, Li Yi, Yixin Zhu, Hang Zhao, Pengwei Wang, Shanghang Zhang, Guocai Yao, Jianyu Chen, Hongyang Li, and Hao Zhao. Dexora: Open-source vla for high-dof bimanual dexterity. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.
- [73] Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, Ankur Handa, Ming-Yu Liu, Donglai Xiang, Gordon Wetzstein, and Tsung-Yi Lin. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.
- [74] Shihan Zhao, Bryon Tjanaka, Matthew C Fontaine, and Stefanos Nikolaidis. Covariance matrix adaptation map-annealing: Theory and experiments. *ACM Transactions on Evolutionary Learning*, 5(1):1–53, 2025.
- [75] Zhenyu Zhao, Hongyi Jing, Xiawei Liu, Jiageng Mao, Abha Jha, Hanwen Yang, Rong Xue, Sergey Zakharov, Vitor Guizilini, and Yue Wang. Humanoid everyday: A comprehensive robotic dataset for open-world humanoid manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.
- [76] Yifan Zhong, Fengshuo Bai, Shaofei Cai, Xuchuan Huang, Zhang Chen, Xiaowei Zhang, Yuanfei Wang, Shaoyang Guo, Tianrui Guan, Ka Nam Lui, Zhiquan Qi, Yitao Liang, Yuanpei Chen, and Yaodong Yang. A survey on vision-language-action models: An action tokenization perspective. *arXiv preprint arXiv:2507.01925*, 2025.

A. Related Works

a) *Vision-Language-Action (VLA) models.*: VLAs aim to provide general purpose robot control by mapping visual observations and language instructions directly to robot actions [76]. Examples of such VLAs include OpenVLA [36], the π family [4, 31, 30, 32, 55], SmolVLA [58], GR00T [51], MolmoAct models [39, 12], and Dexora [72]. These models are trained on broad robot datasets for language-conditioned manipulation. Techniques such as fine-tuning [37], reasoning [69, 73], and steering [8] further improve their success rate and adaptability on downstream tasks. However, making VLAs robust remains challenging because even large robot datasets [35, 11, 75] cover only a limited subset of possible object poses, lighting conditions, camera viewpoints, textures, and scene layouts. Consequently, VLAs remain brittle [65, 24] under minor environment shifts that do not alter the semantic task. To address this, Q-DEG searches for *physical environment variations* that expose failures, then uses the VLA’s successful rollouts in those environments to improve robustness.

b) *Quality Diversity (QD) optimization.*: Quality Diversity (QD) methods search for solutions that are both high-performing and diverse with respect to user-specified measures of diversity. While early QD methods rely on non-adaptive evolution for proposing solutions [47, 56], subsequent works improve search efficiency through covariance matrix adaptation [16, 18]. Key application areas of QD include generating human-robot interaction scenarios [19, 17, 60] and optimizing text with LLMs [5, 14, 40]. Our work is most relevant to methods that apply QD to red-teaming and probing robustness of learned systems. For example, Rainbow Teaming [57] uses QD to generate adversarial prompts for general LLMs, and Q-DIG [61] uses QD to red-team and improve VLAs on adversarial instructions. Unlike these methods, Q-DEG moves the QD search from instructions to *environment parameters*, allowing red-teaming to probe failures induced by visual and physical changes.

c) *Environment Design.*: A complementary line of work develops simulated environments and benchmarks for evaluating language-conditioned robot policies. Benchmarks such as LIBERO [41], LIBERO-Plus [13], RoboCasa [48, 49], KinDER [29], and VLABench [70] provide valuable testbeds for measuring generalization across tasks, scenes, and physical reasoning challenges. However, these benchmarks are mainly designed for evaluation, with environment variations specified by benchmark designers rather than searched to target a VLA’s failure modes, and the resulting failures are not directly converted into fine-tuning data. Other work on unsupervised environment design automatically generates challenging tasks or curricula for reinforcement learning agents [33, 54] but does not target modern VLA policies. Complementary works study geometric red-teaming for robot manipulation [21], generative simulation for robot learning [67, 34, 66], and world simulators for policy evaluation [62]. Unlike these works, Q-DEG uses

QD to search over interpretable environment parameters and generate feasible adversarial environments targeting a given VLA.

As described in Section II, we sample environment parameters, spanning object location, tabletop background color, lighting parameters, and camera pose, from a multivariate Gaussian distribution. To start the search from relatively familiar environments on which the base VLA has non-zero success rates, we initialize the mean of the environment parameter distribution to match the corresponding parameter values in vanilla LIBERO-Spatial. For parameters with well-defined validity constraints, such as object locations that must remain within the table boundaries, we also clip sampled parameters to their valid ranges. We list the 25 environment parameters we perturb during our experiments along with their corresponding mean vector entries and clipping ranges in Table II.

B. Environment Repair

Before generating an environment from sampled environment parameters, we repair the parameters with a Mixed Integer Quadratic Program (MIQP) to ensure that the resulting environment satisfies validity constraints and remains consistent with the task instruction. The repair focuses on object location parameters due to their well-defined validity and alignment conditions. To obtain more accurate feedback metrics, we encourage the repaired environment to remain close to the proposed environment by setting MIQP to minimize the squared L_2 distance between object locations before and after repair. In other words, the MIQP objective is

$$\min_{\{x_o, y_o\}_{o \in \mathcal{O}}} \sum_o [(x_o - x'_o)^2 + (y_o - y'_o)^2] \quad (2)$$

where (x'_o, y'_o) and (x_o, y_o) are the coordinates of object o before and after repair, respectively. For the MIQP constraints, we describe validity constraints and task-specific alignment constraints below. Since MIQP requires linear constraints, the distance between two objects i, j is expressed using the Chebyshev distance, written as

$$L_\infty(i, j) = \max(|x_i - x_j|, |y_i - y_j|).$$

We begin by defining three *validity constraints* that apply across all tasks:

All objects are within table bounds: $-0.6 \leq x_o \leq 0.45$ and $-0.61 \leq y_o \leq 0.59$, $\forall o \in \mathcal{O}$.

No two objects overlap: We enforce non-overlap by requiring every pair of objects to be separated by at least the sum of their radii. Note that we only resolve overlaps in the horizontal plane, allowing one object to be stacked on top of another if they are sufficiently separated along the z axis. This gives us the constraint

$$|z_i - z_j| \leq h_i + h_j \implies L_\infty(i, j) \geq r_i + r_j, \quad \forall i, j \in \mathcal{O}, i < j$$

where z_i , h_i , and r_i are the z coordinate, half-height, and radius padding of object i , respectively.

TABLE II

ENVIRONMENT PARAMETERS PERTURBED DURING OUR SIMULATION EXPERIMENTS. CENTER DENOTES THE ENTRY OF THE MULTIVARIATE GAUSSIAN MEAN VECTOR CORRESPONDING TO EACH ENVIRONMENT PARAMETER. RANGE DENOTES THE CLIPPING BOUNDS APPLIED TO THE SAMPLED VALUES.

Parameter	Center	Range	Parameter	Center	Range
Object location x			Light source location		
• Target bowl	See Table III		• x	1.0	
• Confounding bowl			• y	1.0	Free
• Plate	0.06	[-0.6,0.45]	• z	4.0	
• Cookie box	0.07		Light source specular		
• Ramekin	-0.20		• Red multiplier		
Object location y			• Green multiplier	0.3	[0,1]
• Target bowl	See Table III		• Blue multiplier		
• Confounding bowl			Camera location		
• Plate	0.20	[-0.61,0.59]	• x	0.66	
• Cookie box	0.03		• y	0	Free
• Ramekin	0.20		• z	1.61	
Tabletop material color tint			Camera rotation vector		
• Red multiplier			• x component		
• Green multiplier	0.5	[0,1]	• y component	0	Free
• Blue multiplier			• z component		

TABLE III

TASK INSTRUCTION AND CORRESPONDING CENTER LOCATIONS OF THE TARGET (TARG.) AND CONFOUNDING (CONF.) BOWLS FOR EACH LIBERO-SPATIAL TASK.

Task instructions	Targ. x	Targ. y	Conf. x	Conf. y
<i>pick up the black bowl between the plate and the ramekin and place it on the plate</i>	-0.05	0.20	-0.18	0.32
<i>pick up the black bowl next to the ramekin and place it on the plate</i>	-0.18	0.32	0.13	-0.07
<i>pick up the black bowl from table center and place it on the plate</i>	-0.075	0.0	0.01	0.31
<i>pick up the black bowl on the cookie box and place it on the plate</i>	0.07	0.03	0.03	-0.27
<i>pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate</i>	0.08	-0.15	0.03	-0.27
<i>pick up the black bowl on the ramekin and place it on the plate</i>	-0.20	0.20	0.07	0.03
<i>pick up the black bowl next to the cookie box and place it on the plate</i>	0.13	-0.07	-0.41	-0.14
<i>pick up the black bowl on the stove and place it on the plate</i>	-0.41	-0.14	0.03	-0.27
<i>pick up the black bowl next to the plate and place it on the plate</i>	0.01	0.31	-0.18	0.32
<i>pick up the black bowl on the wooden cabinet and place it on the plate</i>	0.03	-0.27	-0.41	-0.14

No object is initialized on top of the plate: Since all LIBERO-Spatial tasks require placing the target bowl on top of the plate, the plate needs to be empty at the start of the rollout. We reserve the plate with the constraint

$$L_{\infty}(o, \text{plate}) \geq r_o + r_{\text{plate}}, \quad \forall o \in \mathcal{O}.$$

In addition to basic validity constraints, we add task-specific alignment constraints to ensure that the repaired environment admits a unique ground-truth behavior consistent with the task instruction. Because all LIBERO-Spatial tasks require picking up a target bowl distinguished by a predicate condition while avoiding a confounding bowl, these constraints are designed to disambiguate the two bowls by enforcing that the target bowl satisfies the predicate condition and the confounding bowl does not.

pick up the black bowl between the plate and the ramekin and place it on the plate: To ensure the target bowl₁ is between the plate and the ramekin, we constrain its position to a tightened region centered between the two objects. We first define

$$c_x = \frac{x_{\text{plate}} + x_{\text{ramekin}}}{2}, \quad d_x = |x_{\text{plate}} - x_{\text{ramekin}}|$$

$$c_y = \frac{y_{\text{plate}} + y_{\text{ramekin}}}{2}, \quad d_y = |y_{\text{plate}} - y_{\text{ramekin}}|$$

where (c_x, c_y) denotes the midpoint between the plate and the ramekin, and (d_x, d_y) denotes the width and height of the axis-aligned bounding box. We then add constraints

$$c_x - 0.3d_x \leq x_{\text{bowl}_1} \leq c_x + 0.3d_x$$

$$c_y - 0.3d_y \leq y_{\text{bowl}_1} \leq c_y + 0.3d_y.$$

The factor 0.3 shrinks the admissible region relative to the full bounding box, which makes the target bowl more clearly satisfy the predicate. Conversely, we also add a disjunctive constraint to ensure that the confounding bowl₂ does not satisfy the predicate

$$x_{\text{bowl}_2} \leq c_x - 0.5d_x - \epsilon \quad \vee$$

$$x_{\text{bowl}_2} \geq c_x + 0.5d_x + \epsilon \quad \vee$$

$$y_{\text{bowl}_2} \leq c_y - 0.5d_y - \epsilon \quad \vee$$

$$y_{\text{bowl}_2} \geq c_y + 0.5d_y + \epsilon$$

where $\epsilon = 10^{-6}$ is a small margin introduced to enforce strict separation from the threshold. We implement this disjunction using four binary indicator variables and require that at least one of the inequalities hold. Since the factor 0.5 corresponds to the full bounding box, satisfying any of these inequalities guarantees that bowl₂ is not between the plate and the ramekin. **pick up the black bowl next to the {object} and place it on the plate:** This template is shared by three tasks in

LIBERO-Spatial, where $\langle \text{object} \rangle$ is instantiated as the *ramekin*, *cookie box*, or *plate*. To ensure that the target bowl₁ is close to $\langle \text{object} \rangle$, we constrain

$$L_\infty(\text{bowl}_1, \langle \text{object} \rangle) \leq r_{\text{bowl}_1} + r_{\langle \text{object} \rangle} + 0.1.$$

Conversely, we enforce that the confounding bowl₂ is sufficiently far from $\langle \text{object} \rangle$ through the constraint

$$L_\infty(\text{bowl}_2, \langle \text{object} \rangle) \geq r_{\text{bowl}_2} + r_{\langle \text{object} \rangle} + 0.2.$$

The larger threshold in the latter constraint introduces a separation margin between the target and confounding bowls, making the predicate more easily distinguishable.

pick up the black bowl from table center and place it on the plate: We enforce that the target bowl₁ lies inside the table center bounding box $x \in [-0.1, -0.05]$, $y \in [-0.01, 0.01]$ by defining constraints

$$\begin{aligned} -0.1 &\leq x_{\text{bowl}_1} \leq -0.05 \\ -0.01 &\leq y_{\text{bowl}_1} \leq 0.01. \end{aligned}$$

Conversely, we ensure that the confounding bowl₂ lies outside the bounding box through a disjunctive constraint

$$\begin{aligned} x_{\text{bowl}_2} &\leq -0.1 - \epsilon \quad \vee \\ x_{\text{bowl}_2} &\geq -0.05 + \epsilon \quad \vee \\ y_{\text{bowl}_2} &\leq -0.01 - \epsilon \quad \vee \\ y_{\text{bowl}_2} &\geq 0.01 + \epsilon. \end{aligned}$$

pick up the black bowl on the $\langle \text{object} \rangle$ and place it on the plate: This template is shared by four tasks in LIBERO-Spatial, where $\langle \text{object} \rangle$ is instantiated as the *cookie box*, *ramekin*, *stove*, or *wooden cabinet*. LIBERO-Spatial already initializes the target bowl₁ with a higher z coordinate, so to ensure that bowl₁ is stacked on top of $\langle \text{object} \rangle$, we simply set $(x_{\text{bowl}_1}, y_{\text{bowl}_1}) \leftarrow (x_{\langle \text{object} \rangle}, y_{\langle \text{object} \rangle})$. This task template does not require a converse constraint for the confounding bowl. Because all candidate instances of $\langle \text{object} \rangle$ are only large enough to fit a single bowl, the non-overlap constraints naturally force the confounding bowl away from $\langle \text{object} \rangle$.

pick up the black bowl in the top drawer of the wooden cabinet and place it on the plate: Similar to the previous task, we set $(x_{\text{bowl}_1}, y_{\text{bowl}_1}) \leftarrow (x_{\text{drawer}}, y_{\text{drawer}})$ without defining a converse constraint for the confounding bowl. Because the task requires that the target bowl is positioned in the drawer within the wooden cabinet, the overlap between the target bowl and the cabinet is necessary by design. Therefore, we exempt this object pair from the non-overlap constraints during repair.

C. Embedding Extraction

Our choice to use pre-action-layer activations as policy embeddings is motivated by prior work on anomaly detection in foundation models [23, 46, 59], which often relies on distances in latent representation spaces to identify inputs that are novel relative to a model’s prior experience. The success of these methods suggests that intermediate activations in a foundation model often encode rich information about the model’s internal processing and exhibit meaningful local

geometry. These properties make intermediate activations a natural choice for QD measure space, since the diversity in activation space corresponds to diversity in the model’s internal experience, while the locally meaningful geometry facilitates exploration of neighboring regions once a region of the measure space has been discovered. Among these works, SAFE [23] is particularly relevant to our setting, as it demonstrates that pre-action-layer activations in a VLA contain not only rich information for failure detection, but also meaningful local structure, enabling the identification of clustered “failure zones” within the activation space. Motivated by these findings, we adopt SAFE’s design principle, and extract embeddings from π_0 -FAST, OpenVLA-OFT, and $\pi_{0.5}$ as follows:

π_0 -FAST: During autoregressive action decoding, the final transformer block preceding the action head in π_0 -FAST produces a 2048-dimensional hidden state at each FAST-token decoding step. We average the hidden states across all decoding steps prior to the EOS token, yielding a 2048-dimensional policy embedding.

OpenVLA-OFT: The final transformer block preceding the action head outputs hidden states of shape $\mathbb{R}^{d_a \times d_c \times 4096}$, where d_a and d_c denote the action dimension and action chunk size. We average over the first two dimensions to obtain a 4096-dimensional policy embedding.

$\pi_{0.5}$: Unlike the π_0 -FAST and OpenVLA-OFT, $\pi_{0.5}$ incorporates robot state information as part of its action expert during conditional flow-matching. To ensure that the policy embeddings capture all information relevant to action generation, we extract activations from the final layer before velocity prediction at the first Euler integration step, yielding a 1024-dimensional policy embedding. Because flow-matching is initialized from a standard Gaussian source distribution, the resulting activations can exhibit stochastic variation unrelated to the observation. To obtain more consistent embeddings, we perform a separate flow evaluation initialized from the zero vector rather than a Gaussian sample. This auxiliary evaluation is used solely for embedding extraction and does not affect the actions generated by the policy.

D. Embedding Dimensionality Reduction

The high dimensionality of raw policy embeddings poses a challenge for discretization-based QD algorithms such as CMA-MAE. These methods partition the measure space into discrete cells and require sufficient resolution to effectively cover the space. Because the volume of the measure space grows exponentially with its dimensionality, the number of cells needed to maintain a fixed resolution likewise grows exponentially. As a result, using raw policy embeddings directly as measures would make the computational and memory costs of maintaining the QD archive prohibitively large [53, 27, 64].

To obtain a low-dimensional measure space suitable for discretization-based QD algorithms, a common approach is to fit a PCA model to a collection of feature vectors and project them onto a low-dimensional subspace defined by the leading principal components [22, 26]. Following this

approach, Q-DEG first samples a small set of environments by perturbing the default environment (see Appendix A0c) with fixed Gaussian noise. For each sampled environment, we obtain a policy embedding by rolling out the VLA for a single timestep. We then fit a PCA model to the resulting embeddings and use the first two PCA coordinates as measures during QD search. Although reducing the embeddings to two dimensions discards a substantial amount of information, we find that the first two principal components consistently capture a large fraction of the variation in the sampled policy embeddings. Across π_0 -FAST, OpenVLA-OFT, and $\pi_{0.5}$, the first two components explain approximately 50% of the total variance, while subsequent components contribute substantially less, as illustrated in Figure 4.

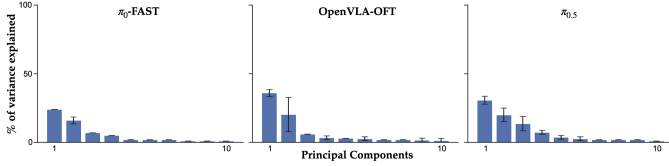


Fig. 4. Variance explained by the first 10 principal components of raw policy embeddings extracted from π_0 -FAST, OpenVLA-OFT, and $\pi_{0.5}$. Results are aggregated over three independent trials, each sampling 320 environments. Error bars denote 95% confidence intervals.

E. Overview of Adversarial Environments on $\pi_{0.5}$

Through manual inspection of the adversarial environments discovered by Q-DEG and the *randomization* and *CMA-ES* baselines, we find that failures identified on $\pi_{0.5}$ can be broadly categorized into six failure modes: (1) placing the target bowl near the edge of the plate rather than at its center, (2) placing the target bowl on an object other than the plate, (3) picking up an object other than the target bowl, (4) dropping the target bowl during transport, (5) failing to pick up any object, and (6) failing to navigate to the plate after picking up the target bowl.

The *randomization* baseline remains close to the default environment configuration and discovers only failure modes (1) and (3) (Table IV). While *CMA-ES* explores environments farther from the default configuration, it concentrates disproportionately on a small subset of environment configurations, resulting primarily in failure modes (1), (2), and (3) (Table V). In contrast, the diversity objective introduced by QD enables Q-DEG to discover adversarial environments spanning all six failure modes (Table VI).

Additional visualizations are available at our anonymous project website: <https://qd-env.github.io/qdeg/>.

F. Method Pseudocode

See Algorithm 1 for an overview of the procedures in Q-DEG.

G. Hyperparameters

We list the hyperparameters we used to evaluate each method in Table VII.

Algorithm 1: Q-DEG

```

1 Q-DEG ( $N, P, M, B, \pi, k, \theta_0, \sigma_0, \alpha$ )
   input: QD iterations  $N$ , PCA calibration iterations
          $P$ , measure space dimension  $M$ , CMA-ES
         batch size  $B$ , base VLA  $\pi$ , rollouts per
         environment  $k$ , initial environment
         parameters  $\theta_0$ , CMA-ES initial step size
          $\sigma_0$ , CMA-MAE learning rate  $\alpha$ .
   result: Archive  $\mathcal{A}$  containing diverse and feasibly
         adversarial environments; fine-tuning
         dataset  $\mathcal{D}$  containing the VLA's successful
         rollouts on these environments.

2 Initialize CMA-ES parameters  $\mu \leftarrow \theta_0$  and
    $\Sigma \leftarrow \sigma_0 I$ .
3 Initialize acceptance thresholds  $\{t_{\text{cell}} \leftarrow 0\}_{\text{cell} \in \mathcal{A}}$ 
   for all archive cells.
4 /* Sample some embeddings  $\Psi$  for
   PCA calibration */
5 for  $iter \leftarrow 1$  to  $P$  do
6    $\theta_i \sim \mathcal{N}(\theta_0, \sigma_0 I)$ 
7    $\psi_i \leftarrow \text{rollout1step}(\text{repair}(\theta_i), \pi, 1)$  // Only
   need first timestep
   embeddings
8    $\Psi \leftarrow \Psi \cup \{\psi_i\}$ 
9    $\mathbf{A}, \mathbf{b} = \text{PCA}(\Psi, M)$ 
10  for  $iter \leftarrow 1$  to  $N$  do
11    for  $i \leftarrow 1$  to  $B$  do
12       $\theta_i \sim \mathcal{N}(\mu, \Sigma)$ 
13       $\theta_i \leftarrow \text{repair}(\theta_i)$ 
14       $\hat{P}_i^{\text{fail}}, \psi_i, \zeta_i^{1:k} \leftarrow \text{rollout}(\theta_i, \pi, k)$ 
15      /* Update assigned archive
      cell's solution and
      threshold */
16       $\text{cell}_i = \mathcal{T}(\mathbf{A}\psi_i + \mathbf{b})$ 
17       $\Delta \hat{P}_i^{\text{fail}} = \hat{P}_i^{\text{fail}} - t_{\text{cell}_i}$ 
18      if  $\Delta \hat{P}_i^{\text{fail}} > 0$  then
19         $t_{\text{cell}_i} \leftarrow (1 - \alpha)t_{\text{cell}_i} + \alpha \hat{P}_i^{\text{fail}}$ 
20         $\mathcal{A}[\text{cell}_i] \leftarrow \theta_i$ 
21       $\mathcal{D} \leftarrow \mathcal{D} \cup \text{shortest}(\{\zeta_i^j \mid \mathbf{1}^{\text{succ}}(\zeta_i^j)\})$ 
        // Keep shortest successful
         $\zeta$ 
22      rank  $\theta_i$  by  $\Delta \hat{P}_i^{\text{fail}}$ 
23      adapt CMA-ES parameters  $\mu, \Sigma$  according to
        rank
24      if CMA-ES stagnates then
25        /* Restart CMA-ES from sampled
        stepping stone */
26        Restart CMA-ES with  $\mu \sim \mathcal{A}$  and
         $\Sigma \leftarrow \sigma_0 I$ .

```

TABLE IV
Randomization FINDS FEW ADVERSARIAL ENVIRONMENTS WITH POOR DIVERSITY.

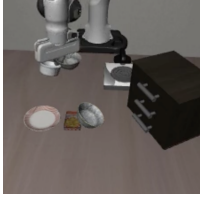
<i>placing the target bowl near the edge of the plate rather than at its center</i>	<i>placing the target bowl on an object other than the plate</i>	<i>picking up an object other than the target bowl</i>	<i>dropping the target bowl during transport</i>	<i>failing to pick up any object</i>	<i>failing to navigate to the plate after picking up the target bowl</i>
	N/A		N/A	N/A	N/A
					

TABLE V
CMA-ES FINDS A LARGE NUMBER OF ADVERSARIAL ENVIRONMENTS BUT IS CONFINED TO A FEW RECURRING CONFIGURATIONS.

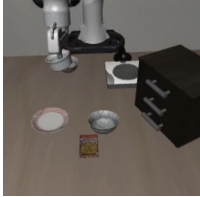
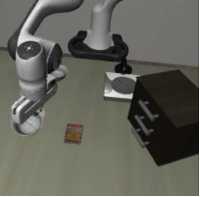
<i>placing the target bowl near the edge of the plate rather than at its center</i>	<i>placing the target bowl on an object other than the plate</i>	<i>picking up an object other than the target bowl</i>	<i>dropping the target bowl during transport</i>	<i>failing to pick up any object</i>	<i>failing to navigate to the plate after picking up the target bowl</i>
			N/A	N/A	N/A
					
					
					

TABLE VI
Q-DEG FINDS DIVERSE ADVERSARIAL ENVIRONMENTS SPANNING SIX FAILURE MODES.

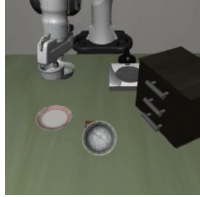
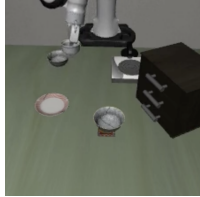
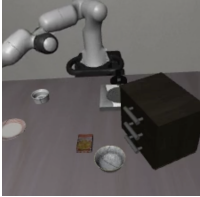
<i>placing the target bowl near the edge of the plate rather than at its center</i>	<i>placing the target bowl on an object other than the plate</i>	<i>picking up an object other than the target bowl</i>	<i>dropping the target bowl during transport</i>	<i>failing to pick up any object</i>	<i>failing to navigate to the plate after picking up the target bowl</i>
					
					

TABLE VII
EVALUATION HYPERPARAMETERS FOR EACH METHOD.

Hyperparameter	Notation	Value
Number of rollouts per environment	k	4
Number of task-specific trajectories mixed into the self-imitation dataset	$ \mathcal{D}_{\text{base}} $	40
Q-DEG		
• QD iterations		
• For simulation experiments	N	96
• For real-world experiments		24
• PCA calibration iterations		
• For simulation experiments	M	4
• For real-world experiments		2
• CMA-ES batch size	B	8
• CMA-ES initial step size	σ_0	0.02
• CMA-MAE learning rate	α	0.1
• QD archive resolution		100×100
randomization		
• Gaussian noise standard deviation	σ	0.02
CMA-ES		
• Search iterations	N	100
• CMA-ES batch size	B	8
• CMA-ES initial step size	σ_0	0.02