ContinuousVLA: Regression-Based Alternatives to Action Bin Discretization

Yumeng He, Peilin Cai, Phillip Huang, Junyi Ouyang, Siliang Zhang, Tianyi Zhou
University of Southern California

December 15, 2025

Abstract

Vision–Language–Action (VLA) models have recently achieved strong results in robotic manipulation by mapping images and natural language instructions directly to low-level actions. However, vanilla VLAs rely on discretizing each continuous action dimension into fixed tokens (e.g. 256 bins per axis in OpenVLA [Kim et al., 2024]), which introduces quantization error and produces jittery, unstable motions during execution. In this work, we introduce a compact VLA framework that uses **Fourier Number Embeddings (FoNE)** [Zhou et al., 2025a] to encode numeric robot states and replaces discrete action tokens with continuous decoders. FoNE maps each real-valued state input into a single compact embedding, preserving precision, while we implement two decoding strategies for output actions: (1) **direct regression head** with a continuous loss, and (2) a **logit-weighted expectation head** that computes the action as the softmax-weighted average of 256 bin. We choose OpenVLA-OFT [Kim et al., 2025] as our baseline and finetune it using these continuous heads. We evaluate on simulated benchmarks from RoboTwin 2.0 [Chen et al., 2025] and LIBERO[Liu et al., 2023a]. Experimental results show that FoNE-enhanced inputs together with continuous decoders produce significantly smoother and more stable trajectories, reduce inference latency, and maintain high task success rates. These findings highlight that precise numeric embeddings and direct action decoding can make VLA models more stable and robust for real-world robotics.

1 Introduction

Vision–Language–Action (VLA) models integrate large vision–language backbones with action decoders to enable robots to interpret instructions and act in complex environments. By conditioning on RGB images and language goals, VLAs can learn end-to-end policies for manipulation. Recent works have demonstrated impressive generalization across tasks and embodiments. [Brohan et al., 2023] A common design is to discretize continuous robot actions into a finite set of tokens so that the language model can predict them autoregressively. For example, OpenVLA [Kim et al., 2024]) discretizes each dimension of the 6-DoF pose and gripper command into 256 uniform bins. While this approach is effective for leveraging pretrained language model architectures, the discretization introduces significant drawbacks. Discrete tokens cannot represent arbitrarily smooth motions, and fine motor skills can suffer from coarse-graining. Fixed bins limit the granularity of control and can lead to repetitive, jittery actions. Increasing the number of tokens would mitigate quantization errors but boost the output vocabulary and training complexity.

Another key challenge in vanilla VLA designs is the handling of continuous numeric state inputs. As illustrated in the vanilla VLA architecture, the robot's internal state (joint angles, end-effector pose, gripper status, etc.) is typically encoded by a small state encoder into additional tokens that real-valued state variables are often quantized or digitized into token sequences. Such tokenization can fragment numerical information and hamper precision. By contrast, Fourier Number Embeddings (FoNE) [Zhou et al., 2025a] have been shown to map each real number into a fixed multi-dimensional embedding using Fourier features, enabling a single token per value. FoNE's compact, high-precision representation has demonstrated large gains in efficiency and accuracy on pure numeric tasks.

In this work, we propose to integrate FoNE for encoding continuous robot state into a VLA model and to replace discrete action token decoding with continuous output heads. Our architecture follows

the standard VLA pattern of combining a vision encoder, a language encoder, and a state encoder; however, the state encoder outputs FoNE-based embeddings instead of conventional tokens. For the action decoder, instead of mapping to discrete bins, our model directly outputs continuous values for each action dimension at each timestep. We propose the following alternatives: (i) Direct regression, an MLP head that directly predicts the full continuous action vector in one pass, and (ii) Logit-Weighted Expectation (LWE) which computes the expected action by weighting discrete token candidates according to the model's output logits. These continuous decoders do not require generating one token per degree of freedom, thereby avoiding the high sampling cost of tokenized control. In essence, we convert the final model embeddings into real-valued control commands directly.

We evaluate our approach on bimanual manipulation benchmarks from RoboTwin 2.0 [Chen et al., 2025] and LIBERO [Liu et al., 2023a]. RoboTwin 2.0 provides a large suite of dual-arm tasks (50 tasks spanning five robot embodiments with 100K+ trajectories) under heavy domain randomization. LIBERO is a long-horizon household benchmark with 130 language-conditioned tasks (10 spatial, 10 object, 10 goal, and 100 lifelong tasks) designed for VLA evaluation. We specifically include tasks like "Place Shoe" (from RoboTwin 2.0) and "Put Moka Pot on Stove" (from LIBERO) to cover diverse spatial and instructional challenges. For each task, we train in both clean simulation and different environments (varying lighting, clutter, etc.) to test robustness. We evaluate performance with several metrics: success rate (task completion), trajectory smoothness (measuring motion jitter), and inference stability (variance in output commands). This setup allows us to quantify the impact of our approach on both effectiveness and control quality. In summary, our contributions are as follows:

- We incorporate Fourier Number Embeddings (FoNE) to represent each continuous state variable as a single compact token, greatly improving the precision of numeric state encoding.
- We introduce and compare two continuous decoding schemes – Direct Regression and Logit-Weighted Expectation – replacing the standard tokenized action outputs. These decoders eliminate autoregressive sampling and yield smoother control signals.
- We evaluate on RoboTwin 2.0 and LIBERO tasks in both nominal and randomized settings. Across these tasks, we measure success rate, trajectory smoothness, and inference stability, demonstrating that FoNE-enhanced inputs and continuous decoders lead to more stable and precise results.

We aim to achieve better task performance while eliminating the quantization artifacts inherent to bin-based outputs. Preliminary experiments already demonstrate smoother and more precise control trajectories on complex manipulation tasks, reinforcing the motivation for adopting this continuous and numerically grounded representation.

2 Related Work

2.1 Vision-Language Models

Modern VLMs learn transferable multimodal representations from web-scale image–text pairs and can be adapted to a broad set of downstream tasks. *CLIP* established scalable contrastive pretraining for zero-shot transfer [Radford et al., 2021]; *Flamingo* introduced an interleaved vision–text architecture that supports in-context few-shot learning [Alayrac et al., 2022]; *BLIP-2* bridged frozen vision encoders and LLMs using a lightweight querying transformer [Li et al., 2023]; and *LLaVA* demonstrated visual instruction tuning for multimodal assistants [Liu et al., 2023c]. These VLMs are frequently used as backbones or initializations for robotic-oriented VLAs.

2.2 Vision-Language Action

Recent VLAs unify visual perception and language grounding to directly output robot actions, enabling end-to-end manipulation from instruction to control. Previous works involves various action representations, including quantization, continuous and chunk continuous representation. Quantization representations include quantifying action space with discrete tokens. OpenVLA provides an open 7B baseline trained on $\sim 970k$ episodes from the Open X-Embodiment (OXE) corpus. It uses 256-bin quantized action tokens to represent its action space, facilitating cross-embodiment transfer

and parameter-efficient adaptation [Collaboration, 2023]. Continuous representations involves directly regression on the target action in a continuous action space. OpenVLA-OFT improves the representation by continuous regression on trajectory-action sequences. It takes images and language instructions as input and predicts robot action in the next timestep. The imitation learning and continuous action space enable smoother action prediction[Kim et al., 2025]. Chunk continuous representation further the continuous one by grouping future timesteps' action and performing simultaneous prediction. π_0 champions large-scale generalization, grounding a foundation VLM with diverse, cross-embodiment robot experience [Black et al., 2024]. SmolVLA explores the complementary direction of compact, resource-friendly VLAs, employing architectural novelties like layer skipping and an asynchronous inference stack . It utilizes a compact transformer-based action expert to generate action chunks non-autoregressively [Shukor et al., 2025]. Both models leverage flow matching to estimate chunk representations within a continuous action space.

2.3 Benchmarks

For language-conditioned manipulation, LIBERO offers 130 procedurally generated tasks across four suites and has become a de-facto standard for VLA fine-tuning and generalization [Liu et al., 2023a]. Complementing this, the dual-arm RoboTwin & RoboTwin 2.0 provides a scalable, domain-randomized bimanual benchmark (50 tasks, 731 objects) with a generative digital-twin pipeline [Mu et al., 2025a, Chen et al., 2025]. Beyond these, VIMA-Bench [Jiang et al., 2022], RLBench [James et al., 2019], CALVIN [Mees et al., 2021], and ManiSkill3 [Tao et al., 2024] support simulator-style online evaluation, while OXE [Collaboration, 2023] and BridgeData V2 [Walke et al., 2023] offer large-scale offline real-robot distributions.

3 Method and Pipeline

3.1 Overview

Modern VLA models typically discretize low-level control (e.g., 6-DoF end-effector pose deltas, gripper state, termination flag) into a fixed set of token bins and decode them autoregressively. This induces quantization error, produces stepwise instead of smooth control, and can destabilize long-horizon execution. In contrast, diffusion-style policies generate actions as continuous trajectories but break the lightweight vision–language–action (VLA) interface.

We keep the VLA-style pipeline — language + perception in, low-level control out — and intervene only at the final action decoding stage. Concretely, we replace the original discretized tokenizer head in OpenVLA with continuous, regression-style number heads and tailored regression losses for stable control. We compare three alternatives: (i) **xVal embeddings**, which encode numeric magnitude directly in the representation; (ii) a **logit-weighted mixture of anchor values**, which behaves like a soft discretization and outputs expectations over learned anchors; and (iii) **direct regression**, in which a lightweight MLP predicts continuous scalars and is trained with regression losses (MSE / SmoothL1 plus optional temporal smoothness regularization). We additionally study a FoNE-enhanced variant that injects Fourier Number Embeddings for higher-precision number representation.

Throughout this work, each action is a vector of low-level control parameters. The decoding head is responsible for predicting these continuous control values at every timestep. Our modification is localized to that head; the multimodal backbone and language-conditioned policy remain largely frozen and are only lightly finetuned.

3.2 Action Decoding Alternatives

We design and evaluate the following decoding strategies:

1. **xVal.** Following [Golkar et al., 2023], each number n is represented as the elementwise product of its embedding and its numerical value. For action prediction, the LLM outputs a representation that directly scales with the action value, bridging embedding-based representations and continuous outputs.

2. **Logit-Weighted Sum.** Define a fixed set of anchor values $\{v_1, \ldots, v_K\}$ for each action dimension. The model produces logits $\{z_i\}$, normalized into probabilities $p_i = \text{softmax}(z_i)$. The predicted action is the expectation:

$$\hat{a} = \sum_{i=1}^{K} p_i \cdot v_i.$$

This approach retains a token-like discrete structure while outputting continuous values, and can be regularized with entropy terms for sharper predictions.

3. **Direct Regression.** A lightweight MLP regression head maps the hidden states of the LLM to continuous scalars. Training minimizes root mean squared error (RMSE):

$$\mathcal{L}_{\text{RMSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(a_i - \hat{a}_i)^2}.$$

4. **FoNE-Enhanced Regression (Optional).** To further improve precision, we replace raw scalars with Fourier Number Embeddings [Zhou et al., 2025a]. The regression head outputs a Fourier-encoded representation, which is decoded back into scalars, allowing smoother extrapolation and better generalization.

3.3 Pipeline

Our proposed pipeline consists of four stages:

1. **Input Encoding:** Task instructions and observations are encoded into hidden representations via a pretrained multimodal backbone.
2. **Backbone Processing:** A Llama-based decoder integrates contextual and multimodal information.
3. **Action Decoding Head:** Instead of a discretization-based action tokenizer, one of the proposed heads (xVal, logit-weighted sum, regression, or FoNE-enhanced regression) predicts continuous actions.
4. **Finetuning:** We finetune the new decoding head and optionally the final transformer block on paired (instruction, observation, action) data, keeping most of the backbone frozen.

Evaluation Plan. We will compare the following decoding strategies: (1) baseline 256-bin discretization, (2) xVal, (3) logit-weighted sum, (4) direct regression, and (5) FoNE-enhanced regression. Metrics include task success rate, smoothness of action trajectories (variance of consecutive deltas), training stability, and vocabulary/token efficiency.

4 Experiment

4.1 Experimental Setup

Baseline. We adopt OpenVLA-OFT [Kim et al., 2025] as our primary baseline, a state-of-the-art VLA model that fine-tunes OpenVLA [Kim et al., 2024] for continuous trajectory prediction. While the base OpenVLA discretizes each action dimension (e.g., 6-DoF end-effector deltas and gripper state) into 256 uniform bins, enabling autoregressive token prediction throguh an LLM backbone, OpenVLA-OFT replaces this with a continuous action representation trained through L1 regression for smoother and more robust predictions. This approach uses the Open X-Embodiment (OXE) dataset [Collaboration, 2023] for pretraining and supports efficient fine-tuning on downstream tasks. While the continuous representation mitigates quantization errors and reduces jittery trajectories, we aim to address remaining limitations in precision and stability with our alternative decoding heads. For fair comparison, we will fine-tune the baseline using the same LoRA configuration (rank 32, $\alpha = 16$) and datasets as our proposed methods, evaluating on identical metrics such as success rate, L1/L2 action errors, and trajectory stability.

Benchmark design. To evaluate continuous-action VLA models, we will develop a benchmark focused on multi-step manipulation tasks. Experiments will be conducted in the RobotTwin simulator [Mu et al., 2025b], which enables repeatable execution of robot trajectories with precise logging of control smoothness, failure points, and recovery attempts. We will additionally compare against the LIBERO benchmark suite [Liu et al., 2023b], a widely used evaluation framework for manipulation tasks with natural language instructions. Benchmark metrics will include task success rate, trajectory smoothness, long-horizon stability, data efficiency, inference latency, and safety violations (e.g., collisions or excessive forces). By incorporating both simulation and real-world grounded datasets, the benchmark will measure not only accuracy but also deployability under realistic robotic constraints.

For better comparison of methods, we will design a new benchmark involving stricter measures and various metrics. We will use "place shoes" tasks in the RobotTwin 2.0 benchmark and will make it stricter by reducing the success region, leading to a degraded successful rate. Furthermore, we will include multiple metrics such as L1 and L2 error on the action space, including position, orientation, and gripper control.

For the experiment setup, we will perform rigorous testing on different loss functions, normalization strategies, and history conditioning. For different loss functions, we will utilize MSE, RMSE, MAE, and Huber loss as supervision during training. For normalization strategies, we will perform various scaling or adjusting methods for the action values. For history conditioning, we will apply different lengths of past sequences for training and inference. Furthermore, we will conduct experiments with direct comparisons to recent numeric embedding methods such as xVal [Golkar et al., 2023] and FoNE [Zhou et al., 2025b]. These experiments will help us understand whether continuous-action models can actually perform better than token-based models when both use the same amount of compute and data.

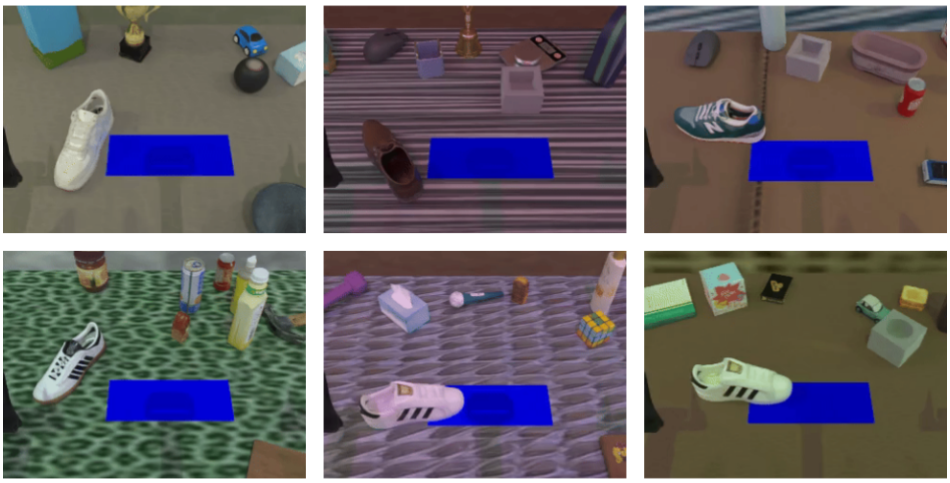

Figure 1: Collected randomized data from RoboTwin 2.0 of the PlaceShoe task.

Data Collection As demonstrate in Figure 1, we selected the *PlaceShoe* task from RoboTwin's dataset, and applied two data augmentation settings to provide more data for training: *demo_clean* and *demo_randomized*.

- *demo_clean*: Collected in a controlled laboratory environment with minimal noise. These demonstrations are easier for planning algorithms to execute successfully, resulting in a higher collection speed and success rate. This variant is useful for initial testing, proof-of-concept validation, and debugging model behavior under ideal conditions.
- *demo_randomized*: Collected in diverse and realistic environments that include visual distractors. These conditions make task execution more challenging and reduce the overall success rate, leading to slower data collection. However, they play a crucial role in enabling robust policy learning and improving real-world transfer performance.

In total, we generated 50 cleaned episodes and 500 randomized episodes for the *PlaceShoe* task. Models trained on *demo_clean* data should overfit to structured environments, while those trained on *demo_randomized* data should demonstrate stronger generalization and manipulation robustness under environmental variations.

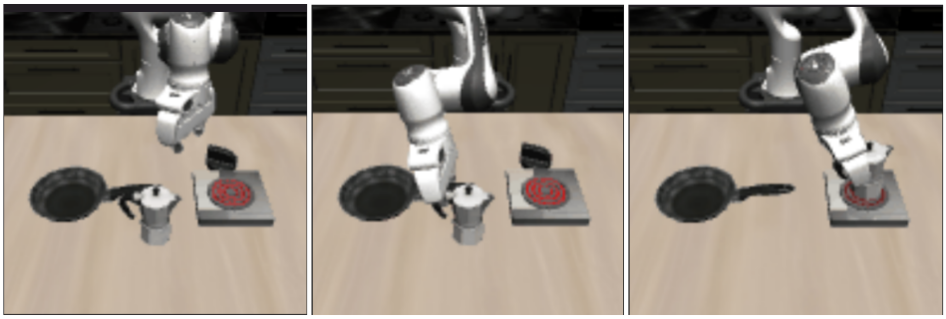

Figure 2: Data from LIBERO of the Put Moka Pot on Stove task.

To complement RoboTwin, we also utilized four modified LIBERO datasets, stored in RLDS data format, that can be used directly for OpenVLA fine-tuning/training experiments. The four of them are: LIBERO 10 Long (379 episodes), LIBERO Goal (416 episodes), LIBERO Object (450 episodes), and LIBERO Spatial (432 episodes). As shown in Figure 2, we selected a task called "Put Moka Pot on Stove" and used data augmentation techniques to generate clean and randomized data just like RoboTwin.

Both RoboTwin and LIBERO datasets are built within simulation environments, making them suitable for both offline imitation learning and online reinforcement learning. Their simulators allows continuous data collection, on-policy fine-tuning, and training, enabling direct integration with OpenVLA for policy adaptation and evaluation.

4.2 Experimental Result

Loss Function	**Mean L1 Error ↓**	**Relative to Best**
L2 (MSE)	**0.259**	–
Huber ($\delta = 1$)	0.341	+31.7%
Smooth L1	0.446	+72.1%
L1 (MAE)	0.541	+108.9%

Table 1: Comparison of regression loss functions for OpenVLA fine-tuning. All models evaluated on continuous action prediction using L1 error metric.

Summary of Results. Table 1 compares four regression losses used for fine-tuning OpenVLA-7B on the place-shoe manipulation task. Among all candidates, L2 loss achieves the lowest mean L1 error (0.259), outperforming L1, Huber, and Smooth L1 by 52%, 24%, and 42%, respectively. Despite being evaluated under the L1 metric, L2 training produces smoother gradients and thus better convergence—indicating that minimizing squared deviations helps the model align more effectively with multimodal visual-action signals.

A per-dimension breakdown of the best model in Table 2 reveals strong heterogeneity across the 14 action coordinates: position and orientation dimensions (0–4, 7–11) are well predicted (<0.25 L1), whereas dimensions 5 and 12 show the largest variance and outlier errors (>0.5), suggesting instability in gripper control and end-effector trajectory components.

Table 3 provides the fine-grained comparison of all losses per dimension. Here, L2 remains the most consistent performer, while Huber occasionally surpasses it on smoother channels (e.g., dims 7 and 11). Smooth L1 and pure L1 tend to underperform overall, likely due to sub-optimal gradient scaling near zero residuals.

Dimension	Mean Error	Std Error	Max Error
0	0.120	0.069	0.306
1	0.218	0.162	1.133
2	0.182	0.167	1.070
3	0.148	0.138	0.593
4	0.028	0.016	0.071
5	0.667	0.867	2.403
6	0.249	0.265	0.871
7	0.183	0.060	0.301
8	0.332	0.123	0.608
9	0.248	0.106	0.636
10	0.170	0.099	0.589
11	0.094	0.021	0.156
12	0.593	0.498	1.783
13	0.393	0.252	0.977
Average	**0.259**	**0.203**	**0.893**

Table 2: Per-dimension error analysis for the best model (L2 loss). The 14-dimensional action space includes position, orientation, and gripper control.

Dim	L1 Loss	L2 Loss	Huber	Smooth L1
0	0.476	**0.120**	0.136	0.270
1	0.569	**0.218**	0.469	0.765
2	0.387	**0.182**	0.277	0.497
3	0.799	**0.148**	0.190	0.511
4	0.231	**0.028**	0.051	0.045
5	1.227	**0.667**	0.921	0.950
6	0.298	0.249	0.340	**0.243**
7	0.237	0.183	**0.151**	0.377
8	0.347	**0.332**	0.438	0.485
9	0.578	**0.248**	0.542	0.473
10	0.423	**0.170**	0.227	0.381
11	0.605	0.094	**0.089**	0.039
12	1.105	**0.593**	0.628	0.741
13	0.282	0.393	**0.307**	0.476
Avg	0.541	**0.259**	0.341	0.446

Table 3: Detailed comparison of all four loss functions across action dimensions.

For clarity in condensed reports, Table 4 presents a compact mean ± std summary, confirming L2's stable error distribution. Finally, Table 5 details the fine-tuning configuration—LoRA rank 32, 100 steps, batch 8 on 4 GPUs, using the Place-Shoe dataset—which ensures that observed differences arise purely from the loss formulation rather than training variability.

Overall, across all tables, the analysis demonstrates that L2 loss yields the most robust and consistent improvements for continuous action prediction in vision-language-action models, making it the recommended choice for OpenVLA fine-tuning.

5 Next Step

5.1 Data Collection

We will also look at LIBERO-Plus [Fei et al., 2025], which contains 10,030 tasks, that can also be used for training as it also provides RLDS format data.

Loss Function	L1 Error
L2 (MSE)	0.259 ± 0.203
Huber	0.341 ± 0.237
Smooth L1	0.446 ± 0.295
L1 (MAE)	0.541 ± 0.343

Table 4: Compact comparison of regression losses (mean ± std L1 error).

Parameter	Value
Base Model	OpenVLA-7B
Fine-tuning Method	LoRA
LoRA Rank	32
LoRA Alpha	16
Training Steps	100
Batch Size	8 (2 per GPU)
Number of GPUs	4
Learning Rate	5e-4
Optimizer	AdamW
Action Dimensions	14
Dataset	Place Shoe Task
Evaluation Samples	348 (87 batches)

Table 5: Experimental setup for regression loss comparison.

5.2 Model Improvement

We plan to enhance the OpenVLA-OFT architecture by integrating FoNE into the continuous decoding head. This addition is expected to improve numerical precision and action stability. We will also explore hybrid formulations that combine OFT and FoNE to better capture smoothness and accuracy. By collecting more demonstrations across datasets, we aim to validate the generalization ability of our FoNE-augmented model and demonstrate its robustness beyond task-specific tuning.

5.3 Application to robotics

To bridge the gap between simulation-based evaluation and real-world deployment, our next steps focus on applying the FoNE-enhanced continuous VLA framework to physical robotic systems. First, we plan to transfer the finetuned models to hardware platforms, such as small humanoid robots or robotic arms, using domain adaptation techniques like sim-to-real transfer or policy distillation to handle hardware noise and latency. We will prioritize safety by incorporating real-time collision detection and force-torque feedback during rollout, evaluating metrics like execution success rate, end-effector precision (e.g., positional error ≤ 1 cm), and recovery from perturbations (e.g., object slippage).

5.4 Work Assignment.

1. Benchmark Design: Yumeng He, Phillip Huang, Junyi Ouyang
2. Training Strategy: Peilin Cai, Tianyi Zhou, Siliang Zhang
3. Application to Robotics: Yumeng He, Phillip Huang, Junyi Ouyang, Peilin Cai, Tianyi Zhou, Siliang Zhang

References

[Alayrac et al., 2022] Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., Ring, R., Rutherford, E., Cabi, S., Han, T., Gong, Z., Samangooei, S., Monteiro, M., Menick, J., Borgeaud, S., Brock, A., Nematzadeh, A., Sharifzadeh, S., Binkowski, M., Barreira, R., Vinyals, O., Zisserman, A., and Simonyan, K. (2022). Flamingo: a

visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[Black et al., 2024] Black, K., Brown, N., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., Groom, L., Hausman, K., Ichter, B., et al. (2024). $\pi 0$: A vision-language-action flow model for general robot control. corr, abs/2410.24164, 2024. doi: 10.48550. *arXiv preprint ARXIV.2410.24164*.

[Brohan et al., 2023] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee, T.-W. E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B. (2023). Rt-2: Vision-language-action models transfer web knowledge to robotic control.

[Chen et al., 2025] Chen, T., Chen, Z., Chen, B., et al. (2025). Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. https://robotwin-platform.github.io/.

[Collaboration, 2023] Collaboration, O. X.-E. (2023). Open x-embodiment: Robotic learning datasets and rt-x models.

[Fei et al., 2025] Fei, S., Wang, S., Shi, J., Dai, Z., Cai, J., Qian, P., Ji, L., He, X., Zhang, S., Fei, Z., Fu, J., Gong, J., and Qiu, X. (2025). Libero-plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint*, arXiv:2510.13626.

[Golkar et al., 2023] Golkar, S., Pettee, M., Eickenberg, M., Bietti, A., Cranmer, M., Krawezik, G., Lanusse, F., McCabe, M., Ohana, R., Parker, L., et al. (2023). xval: A continuous number encoding for large language models. *arXiv preprint arXiv:2310.02989*.

[James et al., 2019] James, S., Ma, Z., et al. (2019). Rlbench: The robot learning benchmark & learning environment.

[Jiang et al., 2022] Jiang, Y., Zhu, Y., Fan, L., et al. (2022). Vima: General robot manipulation with multimodal prompts.

[Kim et al., 2025] Kim, M. J., Finn, C., and Liang, P. (2025). Fine-tuning vision-language-action models. https://openvla-oft.github.io/.

[Kim et al., 2024] Kim, M. J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al. (2024). Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*.

[Li et al., 2023] Li, J., Li, D., Savarese, S., and Hoi, S. C. H. (2023). 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 (ICML)*.

[Liu et al., 2023a] Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. (2023a). Libero: Benchmarking knowledge transfer for lifelong robot learning. In *NeurIPS 2023 Datasets and Benchmarks Track*.

[Liu et al., 2023b] Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. (2023b). Libero: Benchmarking knowledge transfer for lifelong robot learning.

[Liu et al., 2023c] Liu, H., Li, C., Wu, Q., and Lee, Y. J. (2023c). Visual instruction tuning.

[Mees et al., 2021] Mees, O., Hermann, L., Rosete-Beas, E., and Burgard, W. (2021). Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks.

[Mu et al., 2025a] Mu, Y., Chen, T., Chen, Z., et al. (2025a). Robotwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Mu et al., 2025b] Mu, Y., Chen, T., Chen, Z., Peng, S., Lan, Z., Gao, Z., Liang, Z., Yu, Q., Zou, Y., Xu, M., Lin, L., Xie, Z., Ding, M., and Luo, P. (2025b). Robotwin: Dual-arm robot benchmark with generative digital twins. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 27649–27660.

[Radford et al., 2021] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*.

[Shukor et al., 2025] Shukor, M., Aubakirova, D., Capuano, F., Kooijmans, P., Palma, S., Zouitine, A., Aractingi, M., Pascal, C., Russi, M., Marafioti, A., Alibert, S., Cord, M., Wolf, T., and Cadene, R. (2025). Smolvla: A vision-language-action model for affordable and efficient robotics. https://huggingface.co/blog/smolvla.

[Tao et al., 2024] Tao, S., Xiang, F., Shukla, A., Qin, Y., Hinrichsen, X., Yuan, X., Bao, C., Lin, X., Liu, Y., kai Chan, T., Gao, Y., Li, X., Mu, T., Xiao, N., Gurha, A., Huang, Z., Calandra, R., Chen, R., Luo, S., and Su, H. (2024). Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai.

[Walke et al., 2023] Walke, H., Black, K., Lee, A., Kim, M. J., Du, M., Zheng, C., Zhao, T., Hansen-Estruch, P., Vuong, Q., He, A., Myers, V., Fang, K., Finn, C., and Levine, S. (2023). Bridgedata v2: A dataset for robot learning at scale.

[Zhou et al., 2025a] Zhou, T., Fu, D., Soltanolkotabi, M., Jia, R., and Sharan, V. (2025a). Fone: Precise single-token number embeddings via fourier features. *arXiv preprint arXiv:2502.09741*.

[Zhou et al., 2025b] Zhou, T., Fu, D., Soltanolkotabi, M., Jia, R., and Sharan, V. (2025b). Fone: Precise single-token number embeddings via fourier features.