Foundations of Vision Language Action Models for Generalist Robotic Manipulation

Gabriel A. Dorneles[1], Kristofer S. Kappel[1], Rodrigo S. Guerra[1], and Paulo L. J. Drews-Jr[1].

***Abstract*— Vision-language-action (VLA) models have emerged as a promising approach for executing long-horizon manipulation tasks from natural language instructions by reasoning over visual perception, linguistic context, and embodied actions. Building on vision language models, VLAs aim to generalize across tasks, objects, and environments while operating under real-world conditions. This paper presents an overview of the foundational principles underlying VLAs for generalist robotic manipulation. We review commonly used simulation frameworks and benchmarks, and categorize recent VLA architectures according to their action generation mechanisms. By providing a unified view of these topics, this work aims to clarify the current design space of VLA systems and identify trends and open challenges toward general-purpose robotic manipulation.**

I. Introduction

Vision-language-action (VLA) models extend the capabilities of vision-language-models (VLMs) by incorporating embodied actions and decision-making. While VLMs excel at passive tasks, VLA systems focus on allowing agents, often robots, to execute long-horizon, real-world tasks guided by natural language instructions. This area requires adapting models to handle domain-specific challenges, including continuous motion, real-time physical constraints, and 3D environment understanding.

Despite the successes achieved in the linguistic and visual-linguistic domains, the development of VLA foundation models remains a work in progress. While language and vision models benefit from massive datasets extracted from the web, no equivalent large-scale repository of robot interaction data exists. As a result, learning new robotic skills demands extensive, robot-specific data collection tailored to the application [1].

Analogous to how humans leverage prior experience to acquire new skills, a generalist robot policy could be adapted to new tasks with relatively limited additional data. Generalist models often outperform specialist systems precisely because their massive pretraining enables better generalization and transfer, even on the specialists' own tasks [1].

Recent work therefore emphasizes large-scale pre-training for robot learning, inspired by the success of foundation models in vision and language. By utilizing data from different tasks, embodiments, and environments, generalist VLA models can alleviate data scarcity while improving generalization, capturing behaviors that are unlikely to emerge from narrowly specialized datasets.

This survey provides a systematic overview of the current state of VLA models, including their history, architectural designs, and the metrics and benchmarks used to evaluate their performance. Throughout this paper, we aim to provide readers with a foundational understanding of how VLA models are developed and applied, with a particular emphasis on their evolution toward generalist robotic manipulation.

Although several surveys have been published on VLAs and related multimodal models, most either focus primarily on visual or linguistic aspects, don't provide an analysis of evaluation metrics [2], or predate recent advances in the field [3]. Consequently, few works assess how current benchmarks measure performance in embodied contexts, and even fewer concentrate specifically on robotic manipulation, where challenges such as continuous control and action rate play an important role.

The document is organized as follows: Section I introduces the topic, Section II reviews current evaluation practices for VLA models, including commonly used metrics and frameworks. Section III surveys the main VLA models and their architectural designs. Finally, Section IV concludes the paper.

II. Evaluation of VLA Models

Evaluating models for robotic manipulation remains an open challenge. Existing metrics for VLA evaluation are still insufficiently standardized, especially for real-world deployment. Measuring generalization on physical robots is hindered by differences in embodiment, safety constraints, environmental variability, and limited reproducibility [2].

Efforts have been made to build standardized robot setups for controlled benchmarking [4]. However, most evaluations are performed in simulation, where controlled environments, simple resets, and standardized benchmarks facilitate quantitative comparison across different models [2]. This section examines several of the primary simulation frameworks and evaluation metrics utilized for robotic manipulation.

A. Evaluation Frameworks

MuJoCo-based [5]. Many robotics simulation environments, particularly those focused on manipulation, have been built on top of MuJoCo. Among these, robosuite is a modular simulation and benchmarking framework for robot learning, its main goal being to support the research and development of data-driven robotic algorithms and techniques. Its primary

*This work was partly supported by CNPq and FAURG.

[1]NAUTEC, Centro de Ciências Computacionais, Universidade Federal do Rio Grande - FURG, Brazil

979-8-3195-1715-9/26/$31.00 ©2026 IEEE

advantages over other simulators include a modular design that offers greater flexibility in creating new simulation environments, the availability of pre-implemented robotic controllers and learning algorithms, and a set of benchmark tasks aimed at evaluating and developing algorithms.

Version 1.0 of robosuite includes seven robot models, eight gripper models, six controller modes, many predefined objects and nine standardized tasks. Building on robosuite, robomimic [6] introduces a systematic benchmark for evaluating learning-from-demonstration methods in robotic manipulation, with eight additional tasks. RoboCasa [7] extends robosuite with photorealistic household environments, interactable furniture, AI-generated tasks and objects, and over 100 tasks spanning multiple robotic platforms.

The most widely used benchmark for evaluating VLA manipulation models is LIBERO [8], also built on top of robosuite. LIBERO is a large-scale benchmark designed to study lifelong learning in robots (the ability to acquire and reuse skills across many tasks over time, rather than being trained once on a fixed dataset). It introduces a procedural generation pipeline capable of producing an unlimited number of language-conditioned tasks based on everyday human activities, and provides 130 standardized manipulation tasks grouped into four suites that isolate different forms of knowledge transfer: spatial, object-centric, goal-oriented, and mixed.

Isaac Lab [9]. Isaac Lab uses NVIDIA PhysX for multi-sensor rendering, allowing perception-based sensing. Isaac directly accesses the GPU system to support simulations of cameras, LiDARs, radars, and ultrasonic sensors, with realistic scene rendering. Beyond sensor simulation, Isaac Lab supports robot-learning workflows by offering a high-fidelity environment that reduces the sim-to-real gap, particularly for policies that rely on vision-based perception.

More recently, Isaac Lab has been increasingly adopted as a development and evaluation platform for VLA models, due to its visual realism and multi-view rendering. Isaac Lab includes standardized manipulation environments, and has seen growing support from both industry and academia. For instance, NVIDIA introduced two industrial-scale manipulation tasks (Nut Pouring and Pipe Sorting) using a dual-arm humanoid platform in Isaac Lab, while external efforts such as the OpenArm platform [10] offer support for their hardware.

Bullet-based. The open-source Bullet physics engine is widely used for fast robot simulation. RLBench [11] is a classic PyBullet benchmark offering 100 unique manipulation tasks (picking, pushing, sliding, stacking, etc.). Another Bullet-powered simulator is Habitat [12], developed by Meta AI, primarily used for navigation. Recently, Habitat has been extended to mobile manipulation scenarios with Habitat 2.0 [13] and human avatars with realistic interactions in Habitat 3.0 [14]. More broadly, PyBullet and Bullet also power custom Gym tasks and robotics frameworks (e.g. Gazebo with Bullet solver) allowing for VLA research on CPU-friendly hardware, but their rendering pipelines typically prioritize simulation speed over photorealism, resulting in limited visual fidelity and simplified lighting, textures, and sensor noise.

Website: `https://developer.nvidia.com/physx-sdk`

Unity-based. Unity3D provides photo-realistic rendering in a massively utilized 3D engine. The AI2-THOR [15] family is a prominent example: it contains dozens of indoor scenes, the original "iTHOR" has around 120 rooms while RoboTHOR has 89 furnished apartments. Agents in AI2-THOR can perform high-level actions (open, pick, place, etc.) in these environments.

SIMPLER [16] is a cross-simulator evaluation suite designed to benchmark real-world manipulation policies in simulation. Instead of constructing full digital twins, SIMPLER focuses on achieving strong real-to-sim correlation by mitigating control and visual gaps through system identification, background blending, and object/robot texture matching. It provides standardized environments replicating common real-robot setups (e.g. RT-series Google Robot, BridgeData V2) and supports multiple physics engines. Its main contribution lies in showing that carefully constructed simulated environments can strongly correlate with real robot performance: across RT-1/RT-2/RT-1-X/Octo policies and 1500 paired rollouts, SIMPLER achieves Pearson correlations up to 0.97 and demonstrates that its evaluations accurately capture fine-grained characteristics of real-world policies beyond average performance.

B. Evaluation Metrics for VLAs

Modern simulators not only provide realistic physics but also enable the evaluation of vision-language-action (VLA) policies under controlled conditions. In practice, VLA models are assessed using a variety of metrics at multiple levels of abstraction.

The most common method for evaluating these models remains a high-level task success rate. For example, the ALFRED [17] benchmark defines task success as 1 if all goal conditions are satisfied at episode end and 0 otherwise, the overall rate is then the mean over trials. In other cases, the episodes will receive a score of 1.0 for a full success, and a fractional score for partial successes [18], [1], [19]

By extension, the generalization gap is often defined as the drop in success rate from in-distribution conditions to out-of-distribution or unseen conditions. Zero-shot success, in contrast, measures the absolute performance on tasks that were not fine-tuned or new object instances. For example, OpenVLA [20] and RT-2 [18] report both metrics by comparing performance on seen tasks against success rates under several generalization

In continual or lifelong VLA settings, the evaluation focuses on how knowledge transfers across tasks. Standard metrics include *Forward Transfer (FWT)* [21], which measures how prior experience on earlier tasks benefits performance on new ones. Let $c_{i,j}$ denote the success rate on task j after completing training on task i, and let $c_{k,k,e}$ denote performance on task k at training epoch e. The FWT

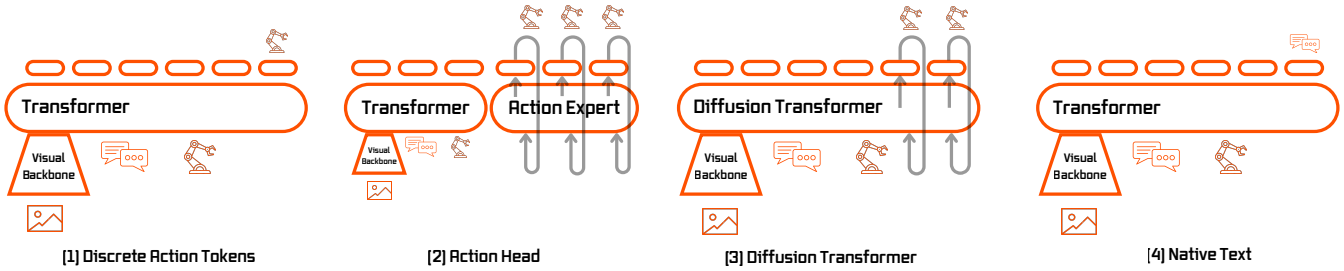

Fig. 1: This figure illustrates four of the main approaches utilized for generating actions in recent VLA research. (1) A vision-language backbone predicting discretized actions. (2) A diffusion or flow matching action head is added for generating continuous actions. (3) Diffusion integrated into the transformer architecture. (4) Native text generation utilizing an unmodified VLM.

is defined as

$$\text{FWT}_k = \frac{1}{11} \sum_{e \in \{0,\ldots,50\}} c_{k,k,e} \tag{1}$$

$$\text{FWT} = \frac{1}{K} \sum_{k \in [K]} \text{FWT}_k, \tag{2}$$

Negative Backward Transfer (NBT), in contrast, quantifies the extent of forgetting on previously learned tasks, computed as

$$\text{NBT}_k = \frac{1}{K-k} \sum_{\tau=k+1}^{K} (c_{k,k} - c_{\tau,k}), \tag{3}$$

$$\text{NBT} = \frac{1}{K} \sum_{k \in [K]} \text{NBT}_k, \tag{4}$$

where $c_{k,k}$ is the peak performance on task k immediately after it is learned, and $c_{\tau,k}$ is performance on the same task after subsequent tasks $\tau > k$ are acquired.

The area under the success rate curve (AUC) aggregates overall performance across all tasks considering both NBT and FWT:

$$\text{AUC}_k = \frac{1}{K-k+1} \big(\text{FWT}_k + \sum_{\tau=k+1}^{K} c_{\tau,k}\big), \tag{5}$$

$$\text{AUC} = \frac{1}{K} \sum_{k \in [K]} \text{AUC}_k, \tag{6}$$

For example, the LIBERO benchmark uses FWT, NBT, and AUC (all based on per-task success rates) to assess lifelong adaptation [8].

III. Vision-Language-Action Models

Recent research of VLA models has taken multiple approaches, which we categorize in this section according to the mechanisms used to generate robot actions. Specifically, we distinguish between models based on discrete action tokens, VLAs with generative action heads using diffusion or flow matching, Diffusion Transformer–based VLAs, and approaches based on native text generation. Fig. 1 illustrates the architecture of these four approaches and Table I compares the different models.

A. Discrete Action Tokens

The Robotics Transformer 1 (RT-1) [22] is regarded as the first VLA that unifies a broad range of robotic tasks [2]. Its architecture processes visual observations through an ImageNet pretrained convolutional neural network [39], compresses features using a Token Learner module [40], and applies a decoder-only Transformer to produce discretized action tokens.

Following the approach of RT-1, RT-2 [18] extends the framework by building on a pre-trained VLM backbone with PaLI-X [23] and PaLM-E [24], fine-tuning these large vision-language models trained on web-scale data to directly act as generalizable and semantically aware robotic policies. By formatting the low-level action output as a string of tokens in response to a prompt, the VLM backbone, which is typically designed for open-vocabulary tasks like visual question answering, can be directly trained to serve as an instruction-following robotic policy. Importantly, RT-2 coined the term Vision-Language-Action (VLA), and this VLM-based design has since become the standard architecture for VLAs [2].

OpenVLA [20], a 7-billion-parameter open-source VLA model, closely mirrors this VLM-based architecture. To enable robot control, OpenVLA follows the same discrete tokenization method as RT-2: it represents continuous robot actions by discretizing each action dimension into 256 bins, which are then mapped to discrete text tokens, replacing the 256 least-used tokens. The policy is trained using a standard next-token prediction objective (cross-entropy loss) by fine-tuning the VLM backbone on a large collection of robot demonstrations from the Open X-Embodiment dataset. This approach yielded a new state-of-the-art for generalist robot manipulation policies, successfully outperforming the closed-source RT-2-X (55B) model by a significant margin despite being much smaller in parameter count.

While OpenVLA generally outperforms the RT-2-X model across several generalization categories, RT-2-X specifically maintains an edge in semantic generalization tasks. This is due to its VLM backbone being pretrained on larger-scale internet data and co-fine-tuned with both robot actions and web data, whereas OpenVLA is fine-tuned solely on robot data.

TABLE I: Comparison of different VLA models and their architecture.

Year	Model	Vision-Language Backbone	Action Output	Max Action Rate
2023	RT-1 [22]	CNN + Token Learner + Transformer	Discrete Action Tokens	3Hz
2023	RT-2 [18]	PaLI-X [23] & PaLM-E [24]	Discrete Action Tokens	5Hz
2024	Octo [25]	CNN + Transformer	Diffusion Action Head	15Hz
2024	OpenVLA [20]	Prismatic-7B [26]	Discrete Action Tokens	10Hz
2024	π_0 [1]	PaliGemma [27]	Flow Matching Action Head	50Hz
2024	TinyVLA [28]	Pynthia [29] + ViT	Diffusion Action Head	71Hz
2025	SmolVLA [30]	SmolVLM-2 [31]	Flow Matching Action Head	Not Specified
2025	OpenHelix [32]	LLaVA [33]	Diffusion Action Head	Not Specified
2025	$\pi_{0.5}$ [34]	PaliGemma	Flow Matching Action Head	50Hz
2025	GR00T N1 [35]	NVIDIA Eagle-2 [36]	Diffusion Transformer	120Hz
2025	VLA-0 [37]	Qwen-VL-2.5 [38]	Native Text Generation	4Hz

B. Diffusion Action Head

The first VLA to use Diffusion Policy was Octo [25], an open-source generalist robot policy specifically designed for robotic manipulation. Octo supports flexible goal specification, using natural language commands via a pre-trained T5 language encoder, or through goal images via a lightweight CNN encoder. Its architecture uses a transformer followed by a diffusion action head, denoising diffusion objectives for action decoding and predicting a chunk of several consecutive actions.

The diffusion training objective achieves improved performance compared to using simple Mean Squared Error (MSE) loss or cross-entropy loss on discretized actions, as it can model multimodal action distributions while maintaining the precision of continuous actions. Importantly, only a single forward pass through the transformer backbone is required per action prediction, after which the multi–step denoising procedure is executed entirely within the lightweight diffusion head [25].

A recent development that builds on the advantages of diffusion-based action decoding is TinyVLA [28], a family of compact Vision-Language-Action models designed to achieve fast inference and strong generalization without requiring large-scale pre-training. Unlike prior VLAs such as OpenVLA, which rely on multi billion parameter backbones and slow autoregressive action token generation, TinyVLA initializes its policy with small multimodal models (70M–1.4B parameters) and integrates a diffusion policy head for continuous action prediction. TinyVLA further adopts efficient fine-tuning via LoRA, allowing the pre-trained vision-language backbone to adapt to robotic tasks while retaining multimodal priors.

C. Flow Matching Action Head

The π_0 foundation model [1] uses flow matching [41], [42], a deterministic alternative to diffusion that learns a continuous vector field without requiring a noisy forward process, on top of a VLM backbone to model the continuous distribution of actions, making it well suited for high frequency tasks up to 50Hz. It integrates a second set of weights called the *action expert* which is used for robotics-specific inputs and outputs, while the core VLM backbone remains responsible for processing visual observations and language-based task descriptions.

Building on top of the π_0 VLA, $\pi_{0.5}$ [34] includes a range of different data sources including natural language instructions, data from non-mobile robots, and large-scale multimodal web-derived datasets, such as image captioning, question answering, and object localization. Unlike π_0, which directly trains a continuous-action head via flow matching, $\pi_{0.5}$ first learns to represent actions and reasoning as discrete tokens, after which a post-training stage aligns the model with the π_0 action expert for continuous control.

During inference, the model first outputs a high-level subtask and then, conditioned on this subtask, generates the corresponding low-level motor commands through the action expert, as shown in Fig. 2. As a result, the $\pi_{0.5}$ VLA demonstrates strong open-world generalization, successfully controlling mobile manipulators to execute multi-stage, dexterous tasks in previously unseen home environments.

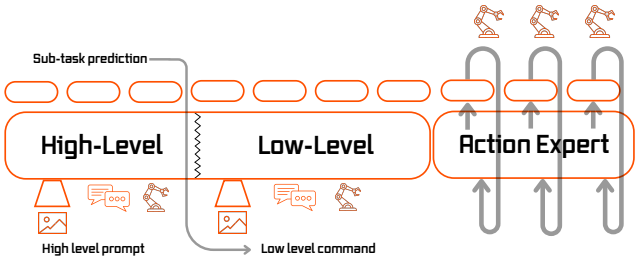

Fig. 2: Overiew of the $\pi_{0.5}$ architecture. The model adapts the action head architeecture by structuring inference into high-level subtask prediction, low-level command generation, and an action expert that executes continuous control at high frequency.

In contrast to the resource-intensive nature of many VLA models, SmolVLA [30] is presented as a lightweight, efficient open-source VLA designed for accessible robotics. It is optimized for training on a single consumer-grade GPU and deployment on consumer-grade GPUs or even CPUs. SmolVLA's architecture is built around a compact pretrained VLM and an action expert trained with flow matching. To achieve its efficiency goals, SmolVLA limits the number of visual tokens and implements a layer skipping strategy where the action expert accesses features only up to a specified layer (often half the total layers, $N = L/2$). Furthermore, SmolVLA introduces an asynchronous inference stack that decouples action execution from observation processing and

action prediction, leading to quicker task completion and better responsiveness by avoiding execution lags and increasing the frequency of observation processing.

D. Diffusion Transformer

The GR00T N1 [35] model, an open foundation model developed by NVIDIA for generalist humanoid robots, includes a Diffusion Transformer (DiT) — an architectural design that integrates the Transformer network with the diffusion process, executing this process directly within the transformer to generate actions [2] — as its low-level action execution component within a dual-system architecture.

The DiT module consists of alternating cross-attention and self-attention blocks. The cross-attention blocks permit conditioning on the vision-language features output by the VLM, while the self-attention blocks operate on the noised action token embeddings along with the robot's state. Both the DiT and VLM modules are tightly coupled and jointly optimized during training. This hierarchical approach is similar to $\pi_{0.5}$, which also utilizes a hierarchical policy to bridge high-level language understanding with low-level motor execution.

GR00T N1 is trained on a heterogeneous mix of data, including real-robot trajectories, synthetically generated datasets, and human videos, utilizing techniques like latent action learning from videos (LAPA) [43] to unify many data sources, thereby improving generalization across various robot embodiments, from tabletop arms to humanoid robots.

E. Native Text Generation

VLA-0 [37] represents a class of VLA models that generate robot actions through native text generation. Instead of introducing discrete action tokens, auxiliary action heads, or architectural modifications to a base VLM, VLA-0 prompts the model to output actions directly as textual sequences. This design preserves the original VLM architecture and vocabulary, preventing a decline in the language understanding and grounding capabilities of the underlying VLM.

While VLA-0 is not the first to explore text-based action representations, earlier approaches largely treated text generation as an auxiliary or intermediate step rather than a competitive end-to-end control strategy [44], [45]. In contrast, recent results show that, with an appropriate training and inference recipe, VLA-0 can outperform state-of-the-art VLA models on several benchmarks, including LIBERO (see Section II-A), even surpassing discrete token and generative action-head methods trained on the same data and, in some cases, models with large-scale action pretraining.

Despite its strong performance, VLA-0 exhibits an important practical limitation in its lower control frequency compared to that of action head based methods. Because actions are generated autoregressively as text, inference incurs higher latency, resulting in control rates on the order of a few hertz, compared to tens of hertz commonly achieved by diffusion or flow-matching action heads.

IV. Conclusion

This survey provided a review of the state of VLAs for generalist robotic manipulation. While they hold great promise for enabling long-horizon generalist robotic manipulation, their development remains constrained by challenges such as data scarcity, evaluation limitations, and the inherent complexity of embodied interaction.

Although a true foundation model for robotics has not yet emerged, recent advances suggest steady movement toward this goal. Architecturally, models with explicit action heads remain the most prevalent and consistently achieve the strongest performance, particularly for reactive control. Finally, recent systems such as $\pi_{0.5}$ show a trend toward incorporating multimodal inputs beyond simple vision and language, and data collection efforts are also showing improvement [46]. Together, these developments indicate continued progress toward more general and capable robotic foundation models.

References

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," 2024. [Online]. Available: https://arxiv.org/abs/2410.24164

[2] K. Kawaharazuka, J. Oh, J. Yamada, I. Posner, and Y. Zhu, "Vision-language-action models for robotics: A review towards real-world applications," *IEEE Access*, vol. 13, pp. 162 467–162 504, 2025.

[3] Y. Ma, Z. Song, Y. Zhuang, J. Hao, and I. King, "A survey on vision-language-action models for embodied ai," 2025. [Online]. Available: https://arxiv.org/abs/2405.14093

[4] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, and S. K. et al., "Droid: A large-scale in-the-wild robot manipulation dataset," 2024.

[5] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

[6] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Mart'in-Mart'in, "What matters in learning from offline human demonstrations for robot manipulation," *ArXiv*, vol. abs/2108.03298, 2021.

[7] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," *ArXiv*, vol. abs/2406.02523, 2024.

[8] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *arXiv preprint arXiv:2306.03310*, 2023.

[9] M. Mittal, C. K.-C. Yu, Q. Yu, J. J. Liu, N. Rudin, D. Hoeller, J. Yuan, P. P. Tehrani, R. Singh, Y. Guo *et al.*, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, pp. 3740–3747, 2023.

[10] enatic, "Openarm platform," https://github.com/enactic/openarm, 2024, accessed: 2025-01.

[11] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, pp. 3019–3026, 2019.

[12] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9338–9346.

[13] A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. S. Chaplot, O. Maksymets *et al.*, "Habitat 2.0: Training home assistants to rearrange their habitat," *ArXiv*, vol. abs/2106.14405, 2021.

[14] X. Puig, E. Undersander, A. Szot, M. D. Cote, T.-Y. Yang, R. Partsey, R. Desai, A. W. Clegg, M. Hlavac, S. Y. Min *et al.*, "Habitat 3.0: A co-habitat for humans, avatars and robots," 2023. [Online]. Available: https://arxiv.org/abs/2310.13724

[15] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu *et al.*, "Ai2-thor: An interactive 3d environment for visual ai," *ArXiv*, vol. abs/1712.05474, 2017.

[16] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani *et al.*, "Evaluating real-world robot manipulation policies in simulation," *arXiv preprint arXiv:2405.05941*, 2024.

[17] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 737–10 746.

[18] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 2165–2183. [Online]. Available: https://proceedings.mlr.press/v229/zitkovich23a.html

[19] S. Zhang, Z. Xu, P. Liu, X. Yu, Y. Li, Q. Gao, Z. Fei, Z. Yin, Z. Wu, Y.-G. Jiang *et al.*, "Vlabench: A large-scale benchmark for language-conditioned robotics manipulation with long-horizon reasoning tasks," 2024. [Online]. Available: https://arxiv.org/abs/2412.18194

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

[21] N. D. Rodríguez, V. Lomonaco, D. Filliat, and D. Maltoni, "Don't forget, there is more than forgetting: new metrics for continual learning," *ArXiv*, vol. abs/1810.13166, 2018.

[22] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," in *Proceedings of Robotics: Science and Systems (RSS)*, Daegu, Republic of Korea, Jul. 2023.

[23] X. Chen, J. Djolonga, P. Padlewski, B. Mustafa, S. Changpinyo, J. Wu, C. R. Ruiz, S. Goodman, X. Wang, Y. Tay *et al.*, "Pali-x: On scaling up a multilingual vision and language model," 2023. [Online]. Available: https://arxiv.org/abs/2305.18565

[24] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," 2023. [Online]. Available: https://arxiv.org/abs/2303.03378

[25] Octo Model Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, C. Xu, J. Luo *et al.*, "Octo: An open-source generalist robot policy," in *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024.

[26] S. Karamcheti, S. Nair, A. Balakrishna, P. Liang, T. Kollar, and D. Sadigh, "Prismatic VLMs: Investigating the design space of visually-conditioned language models," in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR, 21–27 Jul 2024, pp. 23 123–23 144. [Online]. Available: https://proceedings.mlr.press/v235/karamcheti24a.html

[27] L. Beyer, A. Steiner, A. S. Pinto, A. Kolesnikov, X. Wang, D. M. Salz, M. Neumann, I. M. Alabdulmohsin, M. Tschannen, E. Bugliarello *et al.*, "Paligemma: A versatile 3b vlm for transfer," *ArXiv*, vol. abs/2407.07726, 2024.

[28] J. Wen, Y. Zhu, J. Li, M. Zhu, Z. Tang, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen *et al.*, "Tinyvla: Toward fast, data-efficient vision-language-action models for robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 10, pp. 3988–3995, 2024.

[29] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff *et al.*, "Pythia: A suite for analyzing large language models across training and scaling," 2023. [Online]. Available: https://arxiv.org/abs/2304.01373

[30] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti *et al.*, "Smolvla: A vision-language-action model for affordable and efficient robotics," 2025. [Online]. Available: https://arxiv.org/abs/2506.01844

[31] A. Marafioti, O. Zohar, M. Farré, M. Noyan, E. Bakouch, P. Cuenca, C. Zakka, L. B. Allal, A. Lozhkov, N. Tazi *et al.*, "Smolvlm: Redefining small and efficient multimodal models," 2025. [Online]. Available: https://arxiv.org/abs/2504.05299

[32] C. Cui, P. Ding, W. Song, S. Bai, X. Tong, Z. Ge, R. Suo, W. Zhou, Y. Liu, B. Jia *et al.*, "Openhelix: A short survey, empirical analysis, and open-source dual-system vla model for robotic manipulation," *arXiv preprint arXiv:2505.03912*, 2025.

[33] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *ArXiv*, vol. abs/2304.08485, 2023.

[34] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, M. Y. Galliker *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Lim, S. Song, and H.-W. Park, Eds., vol. 305. PMLR, 27–30 Sep 2025, pp. 17–40. [Online]. Available: https://proceedings.mlr.press/v305/black25a.html

[35] NVIDIA, N. C. Johan Bjorck andFernando Castañeda, X. Da, R. Ding, L. J. Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang *et al.*, "GR00T N1: An open foundation model for generalist humanoid robots," in *ArXiv Preprint*, March 2025.

[36] Z. Li, G. Chen, S. Liu, S. Wang, V. VS, Y. Ji, S. Lan, H. Zhang, Y. Zhao, S. Radhakrishnan *et al.*, "Eagle 2: Building post-training data strategies from scratch for frontier vision-language models," 2025. [Online]. Available: https://arxiv.org/abs/2501.14818

[37] A. Goyal, H. Hadfield, X. Yang, V. Blukis, and F. Ramos, "Vla-0: Building state-of-the-art vlas with zero modification," *arXiv preprint arXiv:2510.13054*, 2025.

[38] Qwen, :, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu *et al.*, "Qwen2.5 technical report," 2025. [Online]. Available: https://arxiv.org/abs/2412.15115

[39] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[40] M. S. Ryoo, A. J. Piergiovanni, A. Arnab, M. Dehghani, A. Angelova, and G. Research, "Tokenlearner: Adaptive space-time tokenization for videos," in *Neural Information Processing Systems*, 2021.

[41] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," *ArXiv*, vol. abs/2210.02747, 2022.

[42] Q. Liu, "Rectified flow: A marginal preserving approach to optimal transport," *ArXiv*, vol. abs/2209.14577, 2022.

[43] S. Ye, J. Jang, B. Jeon, S. J. Joo, J. Yang, B. Peng, A. Mandlekar, R. Tan, Y.-W. Chao, B. Y. Lin *et al.*, "Latent action pretraining from videos," *ArXiv*, vol. abs/2410.11758, 2024.

[44] Y. Li, Y. Deng, J. Zhang, J. Jang, M. Memmel, R. Yu, C. R. Garrett, F. Ramos, D. Fox, A. Li *et al.*, "Hamster: Hierarchical action models for open-world robot manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2502.05485

[45] D. Niu, Y. Sharma, G. Biamby, J. Quenum, Y. Bai, B. Shi, T. Darrell, and R. Herzig, "Llarva: Vision-action instruction tuning enhances robot learning," 2024. [Online]. Available: https://arxiv.org/abs/2406.11815

[46] G. A. Dorneles, K. S. Kappel, S. L. Briao, J. F. de Souza Santos Lemos, R. da Silva Guerra, and P. L. J. D. Junior, "A review on dataset collection strategies for learning methods in robotic manipulation," in *RoboCup 2025: Robot World Cup XXVIII*, ser. Lecture Notes in Computer Science, A. P. F. M. aes Mascarenhas, A. Ferrein, and R. Villing, Eds., vol. 16460. Springer, Cham, 2026, to appear, due May 2026.