2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

EveryDayVLA: A Vision-Language-Action Model for Affordable Robotic Manipulation

Samarth Chopra, Alex McMoil, Ben Carnovale, Evan Sokolson, Rajkumar Kubendran, and Samuel Dickerson

***Abstract*— While Vision–Language–Action (VLA) models map visual inputs and language instructions directly to robot actions, they often rely on costly hardware and struggle in novel or cluttered scenes. We introduce EverydayVLA, a 6-DOF manipulator that can be assembled for $300, capable of modest payloads and workspaces. A single unified model jointly outputs discrete and continuous actions, and our adaptive-horizon ensembler monitors motion uncertainty to trigger on-the-fly replanning for safe, reliable operation. On LIBERO, EverydayVLA matches state-of-the-art success rates, and in real-world tests it outperforms prior methods by 49% in-distribution and 34.9% out-of-distribution. By combining a state-of-the-art VLA with cost-effective hardware, EverydayVLA democratizes access to a robotic foundation model, and paves the way for economical use in homes and research labs alike. Experiment videos and more details can be found on our project page: https://everydayvla.github.io/**

I. INTRODUCTION

Vision–Language–Action (VLA) models [1] have transformed robotics by learning a direct mapping from raw images and natural-language instructions to motor commands, bypassing hand-designed perception or planning modules. Building atop large Vision–Language Models (VLMs) [2] pre-trained on massive web-scale datasets, VLAs exhibit impressive scene understanding and instruction following. Yet, even with internet-scale pretraining, they remain brittle under unfamiliar lighting [3], novel objects [4], and visual distractors [5], and often fail to generalize to out-of-distribution tasks [6].

Currently, state-of-the-art robotic manipulators typically cost thousands of dollars, driven by high-precision actuators, custom-machined components [7], and multiple control boards and motor drivers [8] [9]. This high hardware cost and system complexity—coupled with the tedious, expensive process of teleoperated data collection for fine-tuning models [10]—severely limit accessibility and slow progress toward wider adoption [11].

To address these challenges, we present a full-stack system, and present **three distinct contributions**.

- Collaborative training with adaptive horizon control (*AdaHorizon*). We jointly train continuous (L1-regression based) and discrete autoregressive action heads, use their disagreement to estimate model uncertainty, and dynamically adjust action horizons to trigger replanning under tight real-time constraints.
- A low-cost, integrated 6-DOF manipulator. Our $300 design achieves better than 10 mm repeatability while utilizing an accessible Arduino Uno breakout board with PCA9685 pulse with modulation (PWM) driver for 12-bit PWM control.
- An automated data-collection pipeline and public dataset. We streamline teleoperation to collect trajectories with language instructions, video, and end-effector poses, releasing more than 1700 task executions to enable scalable fine-tuning across diverse environments.

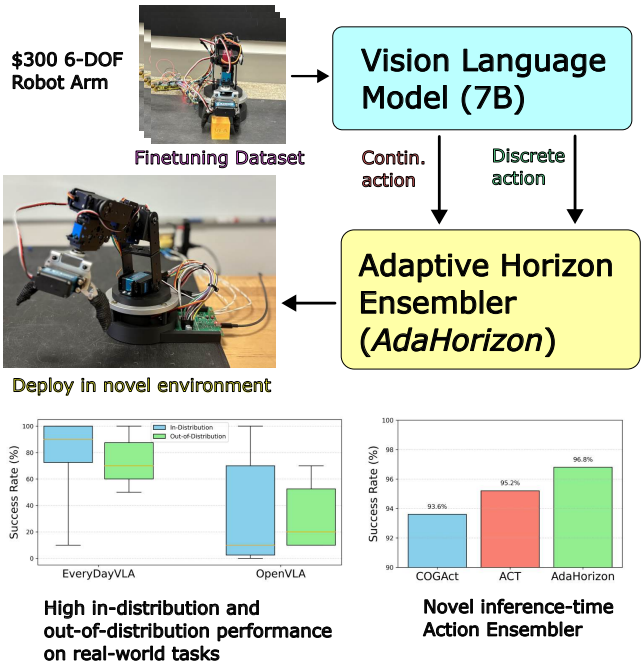

Fig. 1. *EveryDayVLA* system. **Top**: EveryDayVLA finetunes a VLA for a low-cost manipulator to generate continuous and discrete actions, which are passed to an adaptive horizon ensembler, to produce adaptive-sized action chunks, accounting for model uncertainty. **Bottom**: Our model is able to show high in-distribution and out-of-distribution performance on real world tasks, and our action ensembler (*AdaHorizon*) beats other state-of-art action ensemblers.

EveryDayVLA combined with our novel adaptive horizon algorithm (*AdaHorizon*) remains competitive on the LIBERO simulation benchmark [12] and beats state-of-the-art by 34.9% and 49% on real-world in-distribution and out-of-distribution scenarios respectively. By driving down hardware cost (see Table I), our system democratizes access to robotic foundation models, paving the way for broader research and adoption.

We gratefully acknowledge the support of the University of Pittsburgh Center for Research Computing funded by the National Institutes of Health (NIH) under NIH award number S10OD028483.

Samarth Chopra, Alex McMoil, Ben Carnovale, Evan Sokolson, Rajkumar Kubendran, and Samuel Dickerson are with the University of Pittsburgh, {sac345, alm470, bencarnovale, evs44, rajkumar.ece, dickerson}@pitt.edu

TABLE I

COMPARISON OF LOW-COST RESEARCH ROBOTIC MANIPULATORS ALONGSIDE A COMMERCIAL BASELINE.

Arm (Ref.)	Cost	DOF	Payload	Workspace	Max Speed	Repeat.
Low-cost compliant manipulator [7]	$4,135	7	2 kg	–	1.5 m/s	< 3 mm
REPLAB platform (WidowX-250) [13]	$2,000	6	0.25 kg	70×40×60 cm³	–	±1mm
Franka Emika Panda [14]	$29,900	7	3 kg	855 mm reach	2 m/s	±0.1mm
ARMADA [15]	$3,040	6	2.5kg	–	6.16 m/s	2.63mm
AhaRobot [16]	$1,000	14	2.5kg	–	6.16 m/s	2.63mm
PAMY2 [17]	$15,835	4	–	–	12 m/s	–
BLUE [18]	$< \$5,000$	7	2 kg	–	2.1 m/s	3.7mm
Ours	$311.98	6	0.2 kg	382 mm reach	0.7 m/s	$\leq$ 10mm

II. RELATED WORK

A. Vision Language Action (VLA) Models

Recent advances have transformed Vision–Language Models (VLMs) into Vision–Language–Action (VLA) systems that directly predict low-level control commands for robotic manipulators [1], [3], [6], [19], [20], [21], [22]. By harnessing internet-scale pretraining [23] and expansive cross-embodiment datasets [24], [25], [26], these models achieve both language comprehension [27], [28] and rich scene understanding. Early VLA formulations borrowed the autoregressive "next-token prediction" paradigm from language modeling [1], [6], [19], successfully encoding complex behaviors but struggling to learn high-frequency, dexterous skills from dense control data [29]. To overcome this, recent work has shifted toward generating continuous action trajectories via diffusion and flow-matching methods [3], [21], [30], [31], [32]. Although diffusion-based approaches can deliver higher inference throughput, they introduce notable computational trade-offs—namely slower training [33], [34] and multiple denoising or integration steps at inference [35], [36].

Tokenization-based methods like FAST convert continuous trajectories into compact tokens, yielding up to 5× faster training than diffusion-based VLAs [29]. However, because FAST still decodes tokens autoregressively, its inference latency remains high. OpenVLA-OFT [35] tackles this by combining parallel decoding, action chunking [37], and an L1-regression objective: the model generates entire action chunks in one forward pass, boosting throughput proportionally to chunk size. Despite these gains, OpenVLA-OFT inherits the prolonged training cycles typical of continuous-action architectures, and in real-world tests its FiLM-based language grounding [38] can be inconsistent.

Naturally, a hybrid approach, leveraging both autoregressive and continuous predictions have been formulated [39], with action ensembling to fuse predictions. While this has displayed strong performance and robustness on simulation and real-world tasks, it is bottlenecked by slow autoregression.

We take inspiration from this framework, and make a simple change by replacing the diffusion objective with an L1-regression objective. Further, instead of predicting one action at a time, via our autoregressive actions, we borrow OpenVLA-OFT's [35] design philosophy by leveraging action chunking and parallel decoding, enabling joint prediction of discrete and continuous action chunks. We then use the disagreement between the two predictions, to adaptive tune our action horizon at inference, embedding replanning capabilities into the model, while maintaining high inference rates.

B. Low-cost Robotic Manipulators

Several prior efforts have aimed to reduce the cost barrier to robotic manipulators for broader adoption [7], [13], [16], [40], [41]. However, commercially available "low-cost" arms still retail for well over $1,000 (Table I), placing them out of reach for many home users and student researchers. More recently, LeRobot [42] released an open-source $230 robotic arm, though its workspace and payload capacity remain limited. Designing an affordable manipulator requires careful trade-offs among workspace, degrees of freedom (DOF), payload capacity, speed, and repeatability. Most sub-$1,000 solutions compromise one or more of these critical specifications, limiting their utility in research and education. Furthermore, dependence on specialized software frameworks, such as custom driver stacks [43] or ROS extensions [44], creates an additional barrier for novice users.

Guided by these priorities, we engineered an open-source, 6-DOF robotic arm for around $300 that delivers a 0.2 kg payload, 382 mm reach, up to 0.7 m/s end-effector speed, and 10 mm repeatability. To maximize accessibility, we've minimized software dependencies and ensured full OS-agnostic support, lowering the barrier for both home enthusiasts and academic users.

III. METHOD

Autoregressive VLAs suffer due to iterative generation, leading to danger of compounding errors [45], as well as inability to deal with high-frequency robot data [29]. On the other hand, diffusion-based VLAs suffer from long training times [33] [34], and multiple denoising steps. Instead, *EveryDayVLA* takes advantage of action chunking, parallel decoding, and a collaborative training recipe, predicting discrete autoregressive and continuous action chunks, using an L1-regression objective, to achieve high performance

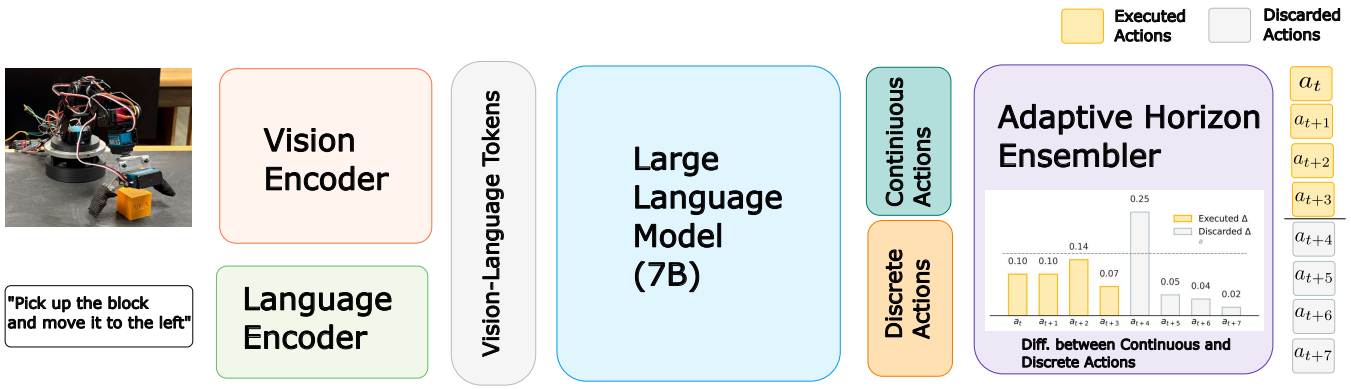

Fig. 2. *EveryDayVLA* architecture. The VLA takes as input an image and natural language instruction and these are tokenized via the vision and language encoders, and sent to the Llama 2 LLM, which produces continuous and discrete actions. These actions are then passed to the adaptive horizon ensembler, which computes the difference between the two actions, executing those only below a certain threshold.

and inference rates. To further improve performance, we introduce a novel adaptive horizon module, allowing the module to execute less number of actions, when the two modes of actions stray from each others.

A. *EveryDayVLA*

EveryDayVLA ingests as input a demonstration consisting of an image observation o_t and a language instruction l_t. The model predicts an action a_{t+1} to control the manipulator for the user-specified task.

$$\pi : (o_t,\, l_t) \;\longrightarrow\; a_{t+1}$$

The action a represents the end-effector pose and is 7-DOF. This action vector includes:

- 3-DOF for relative translation offsets: $[\Delta x,\, \Delta y,\, \Delta z] \in \mathbb{R}^3$,
- 3-DOF for rotation (Euler angles): $\in \mathbb{R}^3$,
- 1-DOF for the gripper state (open/closed): $\in \mathbb{R}^1$.

The ground truth (GT) and the predicted action lie in $\mathrm{SE}(3)$, and are formulated as

$$a = [\,\Delta x,\; \Delta y,\; \Delta z,\; \theta_x,\; \theta_y,\; \theta_z,\; 0/1\,].$$

Vision-Language Model (VLM). EveryDayVLA uses the Prismatic-7B VLM [46], which uses a two part visual encoder, containing pretrained SigLIP [47] and DinoV2 [48] models. For the VLM, we leverage the Llama 2 language model backbone [23]. To obtain continuous action outputs, we pass the final hidden states to a separate action head multi-layer perceptron (MLP). Discrete action tokens are obtained by independently quantizing each of the 7 degrees of freedom into 256 bins. The final hidden states are mapped to action logits, and a softmax operation yields the corresponding probability distribution over tokens.

Collaborative Training. We leverage a collaborative training recipe, inspired by [39], where we train a policy π to predict continuous and discrete action chunks $a \in \mathbb{R}^{KXDX2}$, where K and D represent the size of the action chunk and the dimensionality of the end-effector pose respectively. To supervise the discrete actions, we use cross-entropy loss:

$$\mathcal{L}_{\mathrm{CE}}(\mathbf{A}_t, \hat{\mathbf{A}}_t) = -\sum_{k=1}^{K} A_{t,k}\, \log\big(\hat{A}_{t,k}\big). \tag{1}$$

where $A_t = \{a_t^{(1)}, \ldots, a_t^{(H)}\}$ denotes the ground-truth action chunk of horizon H starting at time t, and $\hat{A}_t = \{\hat{a}_t^{(1)}, \ldots, \hat{a}_t^{(H)}\}$ represents the corresponding predicted action chunk produced by the policy. To supervise the continuous actions, we use the following L1-loss:

$$\mathcal{L}_1(\mathbf{A}_t, \hat{\mathbf{A}}_t) = \|\hat{\mathbf{A}}_t - \mathbf{A}_t\|_1 \tag{2}$$

To simultaneously train continuous and autoregressive action generation, we use the following combined loss function.

$$\mathcal{L}(\mathbf{A}_t, \hat{\mathbf{A}}_t) = \mathcal{L}_{\mathrm{CE}}(\mathbf{A}_t, \hat{\mathbf{A}}_t) \;+\; \lambda\, \mathcal{L}_1(\mathbf{A}_t, \hat{\mathbf{A}}_t) \tag{3}$$

In practice, we set the weight $\lambda = 1$, to balance the optimization of discrete and continuous action outputs.

Adaptive Horizon Ensembler. Inspired by [39], we generate both discrete and continuous action chunks and observe that discrete actions excel at tasks requiring high-level semantic reasoning, while continuous actions provide superior precision for fine manipulation. To harness both strengths, we introduce a more robust uncertainty metric: the mean absolute difference mad_t between the continuous and discrete action predictions, which outperforms the autoregressive model's mean confidence score. Moreover, our collaborative training reveals that discrete actions outperform continuous outputs on average (Table II). Accordingly, rather than fusing actions based on confidence thresholds [39], we adaptively adjust the planning horizon using our difference threshold—executing only those discrete action chunks whose mean absolute difference falls below this cutoff (see Algorithm 1). Using this metric leads to improved performance compared to confidence-based gating methods such as [39] (see Table V). We term our method *AdaHorizon*, and find that it enables discrete actions to match or exceed the precision of continuous controls, even on dexterous tasks

where discrete policies alone struggle. The method incorporates an early-bail mechanism: if the mean absolute difference exceeds a predefined `replan_threshold` for more than `max_replan_count` times, the ensembler defaults to executing the full eight-step action chunk. To satisfy real-time constraints, we enforce a minimum execution length of four actions per eight-step chunk.

Algorithm 1 Adaptive Horizon Ensembler

Require: Continuous actions $\{\mathbf{a}_t^c\}_{t=1}^T$, Discrete actions $\{\mathbf{a}_t^d\}_{t=1}^T$
Require: Parameters: min_actions, replan_threshold, threshold, max_replan_count, next_task_thresh
state: replan_ctr, max_replan_ctr
$T \leftarrow \text{length}(\mathbf{a}^c)$
for $t = 1 \rightarrow T$ **do**
  $\text{mad}_t \leftarrow \frac{1}{D}\sum_{d=1}^{D}\left|\mathbf{a}_{t,d}^c - \mathbf{a}_{t,d}^d\right|$
end for
if $\exists t \in [1, \ldots, \text{min\_actions}] : \text{mad}_t > \text{replan\_threshold}$ **and** $\text{min\_actions} > 1$ **then**
  $\text{replan\_ctr} \mathrel{+}= 1$
end if
$\text{max\_replan\_ctr} \leftarrow \max(\text{max\_replan\_ctr}, \text{replan\_ctr})$
if $\text{max\_replan\_ctr} \geq \text{max\_replan\_count}$ **and** $\text{replan\_ctr} \geq \text{next\_task\_thresh}$ **then**
  return $\{\mathbf{a}_t^d\}_{t=1}^T$, mad
end if
$\text{mad\_mask}_t \leftarrow (\text{mad}_t < \text{threshold}) \quad \forall t$
$\text{horizon} \leftarrow \text{min\_actions}$
for $t = \text{min\_actions} + 1 \rightarrow T$ **do**
  if not mad_mask_t **then**
    break
  end if
  $\text{horizon} \leftarrow t$
end for
return $\{\mathbf{a}_t^d\}_{t=1}^{\text{horizon}}$

B. Hardware

We built a $300 6-DOF robotic manipulator (see Figure 3), using brackets on online marketplaces due to cost constraints. The robot arm is 38.2 cm from the base to the wrist and 46 cm from the base to the tip of the gripper. The robot is actuated by several types of off-the-shelf servo motors (MG996R, DS3225, DS3245) depending on the torque necessary for each joint. The 6 servomotors in the arm are configured in a roll-pitch-pitch-roll-pitch-roll configuration. The 3 servomotors comprising the wrist approximate a spherical joint. This configuration was chosen as the minimum number of servos to achieve any position or orientation.

The bottom-most servo motor was held in a custom, 3-D printed base. The base was secured to 1/2 inch plywood board cut to 0.5m (length) x 1.0m (width) to be centered along the width and at the back of the board lengthwise, to maximize the manipulator's task space. Atop the base, we mounted a 120 mm swivel ball bearing. This ball bearing

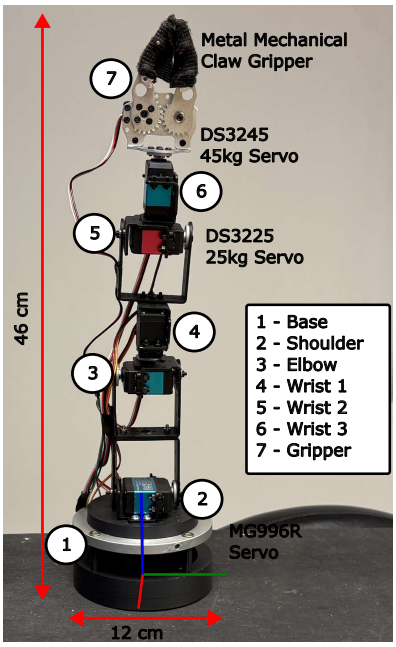

Fig. 3. *EveryDayVLA* hardware. The robot consists of 7 joints, including a base and claw gripper as the end-effector. In sum, the hardware costs $311.98, affording 6 DOF, a payload of 0.2 kg, 382 mm reach, a max speed of 0.7 m/s and a repeatability of withing 10 mm.

was secured to a custom 3-D printed plate where the bracket structure was then mounted.

To precisely control the movements of the robot, servo control, we use an off-the-shelf PCA9685 16-channel, 12-bit servo motor driver breakout board that communicates over I2C with an Arduino Uno.

IV. Experiments

A. Dataset

We fine-tune the OpenVLA-7B model [19] on our custom dataset of 1773 demonstrations—each pairing a natural-language instruction with an RGB observation sequence and corresponding end-effector poses—collected using our $300 manipulator. Our dataset was captured across multiple tabletop environments and spans a diverse set of manipulation tasks. These include object repositioning under random initial configurations (1133 samples), structured placements from uniform positions (425 samples), cup-related tasks such as dropping or pouring objects into a cup (147 samples), block stacking and unstacking tasks (52 samples), and drawer interaction tasks including opening, closing, and placing objects inside (16 samples). To streamline data collection, we abstracted core skills into parameterized trajectory primitives and paired them with generic language templates such as "pick up the ball and place it away from the robot," enabling rapid generation of varied, yet semantically consistent, training samples.

B. Implementation Details

We extended the [35] codebase to jointly train autoregressive and regression heads and deploy with our AdaHorizon ensembler. For finetuning, we use LoRA [50] and 100 k

TABLE II

LIBERO TASK PERFORMANCE RESULTS. THE BEST SCORES ARE HIGHLIGHTED IN BOLD, AND THE SECOND-BEST SCORES ARE UNDERLINED.

Policy inputs: third-person image, language instruction					
	Spatial SR (%)	**Object SR (%)**	**Goal SR (%)**	**Long SR (%)**	**Average SR (%)**
Diffusion Policy (scratch) [3]	78.3	92.5	68.3	50.5	72.4
Octo (fine-tuned) [5]	78.9	85.7	84.6	51.1	75.1
DiT Policy (fine-tuned) [49]	84.2	96.3	85.4	63.8	82.4
OpenVLA [19]	84.7	88.4	79.2	53.7	76.5
OpenVLA-OFT [35]	96.2	**98.3**	**96.2**	**90.7**	**95.3**
Ours (Cont-L1 + AR, Use Cont-L1)	95.0	92.4	90.0	79.4	89.2
Ours (Cont-L1 + AR, Use AR)	96.4	95.0	90.0	80.8	90.6
Ours (Cont-L1 + AR, Use *AdaHorizon*)	**96.8**	95.6	91.0	82.0	91.4

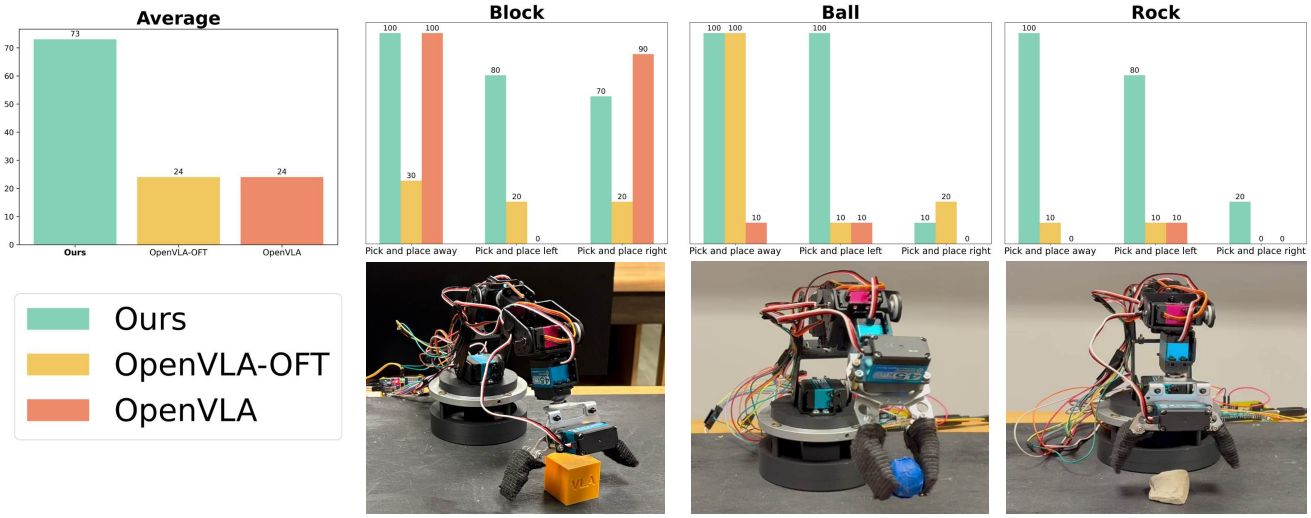

Fig. 4. Real-world evaluation results on in-distribution tasks, including picking a block, ball and rock. Our model is able to beat state-of-the art models on tasks and environments present in the training set by 49% on average. We evaluate on three different objects, with three instructions each, "pick and place {away, left, right}". Our experiments show better success rates on almost every single task.

iterations for simulation (on two A100s) and 50 k for real-world (on one A100). We use a LoRA rank of 32, batch size 8, and 4-step gradient accumulation.

We use IKPy [51] for inverse kinematics to get robot joint angles. For the camera, we use an iPhone 12 mini and the DroidCam app [52] to stream visual inputs to the model. For real-world deployment, inference is performed remotely on an A100 accessed over a Wi-Fi connection.

C. Baselines

On the LIBERO simulation benchmark, we report success rates across all four task suites, comparing against Diffusion Policy [3], Octo [5], DiT Policy [49], OpenVLA [19], and OpenVLA-OFT [35]. In real-world trials, we evaluate both in-distribution and out-of-distribution scenarios against OpenVLA and OpenVLA-OFT. We also measure inference throughput in simulation. Finally, we benchmark our adaptive-horizon ensemble, AdaHorizon, against ACT [37], HybridVLA [39], and COGAct [53].

For all experiments, we use the discrete actions, with the adaptive horizon ensembler (*AdaHorizon*), which yields higher success rates and improved grasp accuracy compared to continuous-action baselines. For each task in the real-world evaluation, we perform 10 independent trials and report the corresponding success rate.

V. RESULTS

TABLE III

INFERENCE RATES ON LIBERO.

Method	**Inference Rate (Hz) ↑**	**Latency (sec) ↓**
OpenVLA [19]	4.2	0.2396
OpenVLA-OFT [35]	**109.7**	**0.0729**
Ours	54.2–108.4	0.0738

TABLE IV

GENERALIZATION AND ROBUSTNESS TO UNSEEN TASKS, ENVIRONMENTS AND CONDITIONS

	Ours	OpenVLA-OFT [35]	OpenVLA [19]
Original	**100**	30	**100**
OOD Tasks	**90** (-10%)	43 (+43%)	43 (-57%)
OOD Environments	**67.5** (-32.5%)	20 (-33%)	15 (-85%)
Static distractors	**80** (-20%)	30 (+0%)	70 (-30%)
Dynamic distractors	**90** (-10%)	20 (-33%)	60 (-40%)

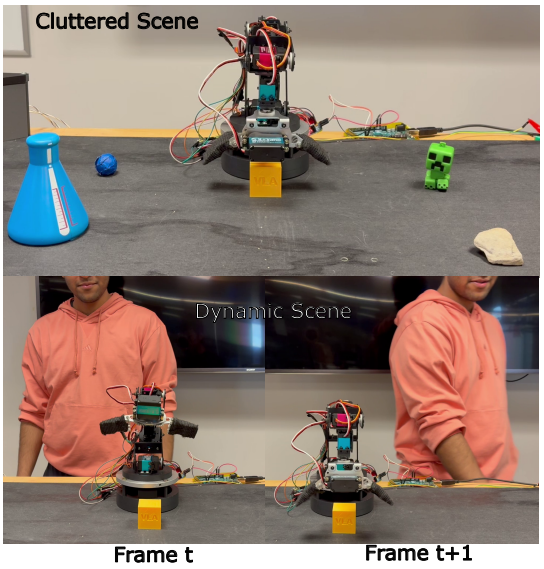

Fig. 5. Static and dynamic distractors. **Top:** We benchmark our model with static distractors, and a cluttered scene where we add different objects, and vary the arrangement after every single trial. **Bottom:** We benchmark with dynamic distrators, where a human walks in the scene and moves to distract the model.

TABLE V

COMPARISON OF ACTION ENSEMBLERS VS FULL DISCRETE AND CONTINUOUS ACTIONS CHUNKS AS A BASELINE ON LIBERO SPATIAL

Method	Success rate (%)
ACT (Temporal Ensembling) [37]	95.2
HybridVLA ($\theta = 0.8$) [39]	94.2
COGAct (Use Cont-L1) [53]	93.6
Ours (Cont-L1 + AR, Use Cont-L1)	95.0
Ours (Cont-L1 + AR, Use AR)	96.4
Ours (Cont-L1 + AR, Use AdaHorizon)	**96.8**

A. Results on LIBERO simulation benchmark

Across the four LIBERO suites, our method trails the top-performing baseline by an average of 3.9 % (Table II). Notably, we outperform the best method on the Spatial suite by 0.6 %, while incurring drop-offs of 2.7 %, 5.2 %, and 8.7 % on the Object, Goal, and Long suites, respectively.

In an ablation against using only continuous (Cont-L1) or only discrete (AR) actions, *AdaHorizon* consistently improves performance in every suite—on average by 0.8 % over the better of the two single-modality policies, demonstrating the value of adaptively combining both action outputs.

We also achieve an inference rate of up to 108.4 Hz (Table III), adding only 0.9 ms of latency relative to OpenVLA-OFT [35], which corresponds to the overhead of our adaptive-horizon module.

B. Results on Real-World Tests

In real-world, in-distribution pick-and-place experiments, EverydayVLA outperforms other methods by an average of 49% in success rate across blocks, balls, and rocks (Figure 4). Although our model performs similarly to [19] on picking and placing blocks (most common task in the collected dataset), the latter struggles when placing to the left, and often fails to release the block (once grasped) in a timely manner. On ball pick-and-place, *EverydayVLA* exceeds both baselines in every variant except "pick-and-place right." For rock manipulation, our model achieves comparable success, underscoring its robustness across diverse object types.

The primary failure mode for *EverydayVLA* is delayed object release and not finishing the task in a timely manner. OpenVLA-OFT [35] most often fails due to misaligned grasps and excessive current draw (which forces a safety shutdown), while OpenVLA [19] suffers from intermittent stuttering and pauses that prevent task completion within the allotted time.

C. Generalization results

On generalization and robustness evaluation of unseen tasks, environments and conditions (Table IV), our model does the best. We outperform the OpenVLA-OFT [19] and OpenVLA [35], on generalization to unseen tasks, environments, and robustness to static and dynamic distractors. We notice minimal dropoffs in presence of distractors.

Compared to static distractors, such as a more cluttered environment, our model only experiences a 20% decline in performance compared to the training environment. Further, in presence of dynamic, moving distractors, such as humans, we see only a 10% performance dropoff.

D. Results on Comparison to Action Ensemblers

We evaluate our ensembler against competing methods on the LIBERO Spatial suite (Table V). *AdaHorizon* outperforms all baselines by dynamically adjusting the executed action-chunk length: for complex manipulation tasks, shorter chunks enable timely replanning and higher success rates, whereas for simpler tasks full-chunk execution is optimal. Almost all existing ensemblers, except temporal ensembling [37], fall below even the continuous and discrete single-modality baselines. In particular, temporal ensembling, HybridVLA [39], and COGAct [53] oversmooth the action stream via sliding-window aggregation, sacrificing the fine-grained control needed for dexterous tasks. Our ensembler shows a 1.6% improvement in success rate compared to the next best method.

VI. CONCLUSIONS

In this work we present *EveryDayVLA*, a framework with a low-cost manipulator, a VLA leveraging a novel collaborative training and an adaptive horizon (*AdaHorizon*) that computes uncertainty from the predicted discrete and continuous actions, adaptively modifying the action horizon.

Our work demonstrates substantial improvement on in-distribution and out-of-distribution scenarios. Specifically, on in-distribution scenarios our model showcases 49% improvement on average compared to the second-best method. On out-of-distribution scenarios, including unseen tasks, environments, static and dynamic distractors, *EveryDayVLA* beats the next best method by 34.9% on average. Further, we show competitive results to the LIBERO benchmark,

placing second-best on average success rate. We also show that our model is fast and that its highest inference rate nearly matches that of [35]. Finally, we beat other existing action ensemblers with our Adaptive Horizon action ensembler, beating the second-best method by 1.6% on the LIBERO Spatial task suite.

Our work has a few limitations. On the hardware front, we have not ensured long-term robustness. In the future, we look to improve the arm's mechanical durability. We also experience limitations in executing fine-grained manipulation, which is due to the limited servo precision as well as relatively low number of expert demonstrations in our fine-tuning dataset (compared to simulation). An extension of our work is to use higher precision servos, and collect more expert trajectories to consolidate our dataset for improved fine-grained control. For our action ensembler, *AdaHorizon* currently requires task-specific tuning to select appropriate mean absolute difference and replanning thresholds. In future work, we aim to make these parameters learnable, eliminating manual tuning and improving adaptability across tasks.

ACKNOWLEDGMENT

We would also like to thank the Senior Design course professors and teaching assistants for their input during project discussions. We would like to thank the staff at Student Electronics Resource Center for assistance with providing the Arduino micro-controller.

REFERENCES

[1] 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," *arXiv preprint arXiv:2212.06817*, 2022.

[2] X. Chen, X. Wang, S. Changpinyo, A. Piergiovanni, P. Padlewski, D. Salz, S. Goodman, A. Grycner, B. Mustafa, L. Beyer *et al.*, "Pali: A jointly-scaled multilingual language-image model," *arXiv preprint arXiv:2209.06794*, 2022.

[3] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, p. 02783649241273668, 2023.

[4] A. Xie, L. Lee, T. Xiao, and C. Finn, "Decomposing the generalization gap in imitation learning for visual robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 3153–3160.

[5] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[7] M. Quigley, A. Asbeck, and A. Ng, "A low-cost compliant 7-dof robotic manipulator," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 6051–6058.

[8] S. A. Arduino, "Arduino," *Arduino LLC*, vol. 372, 2015.

[9] A. Industries, "Adafruit 16-channel 12-bit pwm/servo driver - i2c interface." [Online]. Available: https://www.adafruit.com/product/815

[10] S. Dass, K. Pertsch, H. Zhang, Y. Lee, J. J. Lim, and S. Nikolaidis, "Pato: Policy assisted teleoperation for scalable robot data collection," *arXiv preprint arXiv:2212.04708*, 2022.

[11] C. M. Christensen, *The innovator's dilemma: when new technologies cause great firms to fail*. Harvard Business Review Press, 2015.

[12] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.

[13] B. Yang, J. Zhang, V. Pong, S. Levine, and D. Jayaraman, "Replab: A reproducible low-cost arm benchmark platform for robotic learning," *arXiv preprint arXiv:1905.07447*, 2019.

[14] [Online]. Available: https://www.generationrobots.com/media/panda-franka-emika-datasheet.pdf

[15] J. Kim, J. Kim, D. Lee, Y. Jang, and B. Kim, "Design of a low-cost and lightweight 6 dof bimanual arm for dynamic and contact-rich manipulation," *arXiv preprint arXiv:2502.16908*, 2025.

[16] H. Cui, Y. Yuan, Y. Zheng, and J. Hao, "Aharobot: A low-cost open-source bimanual mobile manipulator for embodied ai," *arXiv preprint arXiv:2503.10070*, 2025.

[17] S. Guist, J. Schneider, H. Ma, L. Chen, V. Berenz, J. Martus, H. Ott, F. Grüninger, M. Muehlebach, J. Fiene *et al.*, "Safe & accurate at speed with tendons: A robot arm for exploring dynamic motion," *arXiv preprint arXiv:2307.02654*, 2023.

[18] D. V. Gealy, S. McKinley, B. Yi, P. Wu, P. R. Downey, G. Balke, A. Zhao, M. Guo, R. Thomasson, A. Sinclair *et al.*, "Quasi-direct drive for low-cost compliant robotic manipulation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 437–443.

[19] M. J. 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.

[20] X. Li, M. Liu, H. Zhang, C. Yu, J. Xu, H. Wu, C. Cheang, Y. Jing, W. Zhang, H. Liu *et al.*, "Vision-language foundation models as effective robot imitators," *arXiv preprint arXiv:2311.01378*, 2023.

[21] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, and B. Ichter, "π_0: A vision–language–action flow model for general robot control," 2024.

[22] P. Intelligence, K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai *et al.*, "π0. 5: a vision-language-action model with open-world generalization, 2025," *URL https://arxiv. org/abs/2504.16054*.

[23] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[24] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 6892–6903.

[25] H. R. Walke, K. Black, T. Z. Zhao, Q. Vuong, C. Zheng, P. Hansen-Estruch, A. W. He, V. Myers, M. J. Kim, M. Du *et al.*, "Bridgedata v2: A dataset for robot learning at scale," in *Conference on Robot Learning*. PMLR, 2023, pp. 1723–1736.

[26] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.

[27] H. Zhou, X. Yao, Y. Meng, S. Sun, Z. Bing, K. Huang, and A. Knoll, "Language-conditioned learning for robotic manipulation: A survey," *arXiv preprint arXiv:2312.10807*, 2023.

[28] L. X. Shi, B. Ichter, M. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang, N. Fusai *et al.*, "Hi robot: Open-ended instruction following with hierarchical vision-language-action models," *arXiv preprint arXiv:2502.19417*, 2025.

[29] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "Fast: Efficient action tokenization for vision-language-action models," *arXiv preprint arXiv:2501.09747*, 2025.

[30] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[31] T. Z. Zhao, J. Tompson, D. Driess, P. Florence, K. Ghasemipour, C. Finn, and A. Wahid, "Aloha unleashed: A simple recipe for robot dexterity," *arXiv preprint arXiv:2410.13126*, 2024.

[32] J. Wen, Y. Zhu, J. Li, M. Zhu, Z. Tang, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen *et al.*, "Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation," *IEEE Robotics and Automation Letters*, 2025.

[33] H. Zhang, Y. Lu, I. Alkhouri, S. Ravishankar, D. Song, and Q. Qu, "Improving training efficiency of diffusion models via multi-stage

framework and tailored multi-decoder architecture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7372–7381.
[34] Z. Wang, Y. Jiang, H. Zheng, P. Wang, P. He, Z. Wang, W. Chen, M. Zhou *et al.*, "Patch diffusion: Faster and more data-efficient training of diffusion models," *Advances in neural information processing systems*, vol. 36, pp. 72 137–72 154, 2023.
[35] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.
[36] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
[37] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," *arXiv preprint arXiv:2304.13705*, 2023.
[38] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.
[39] J. Liu, H. Chen, P. An, Z. Liu, R. Zhang, C. Gu, X. Li, Z. Guo, S. Chen, M. Liu *et al.*, "Hybridvla: Collaborative diffusion and autoregression in a unified vision-language-action model," *arXiv preprint arXiv:2503.10631*, 2025.
[40] P. Wu, Y. Shentu, Z. Yi, X. Lin, and P. Abbeel, "Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 12 156–12 163.
[41] J. A. N. Cocota, H. S. Fujita, and I. J. da Silva, "A low-cost robot manipulator for education," in *2012 Technologies Applied to Electronics Teaching (TAEE)*. IEEE, 2012, pp. 164–169.
[42] R. Cadene, S. Alibert, F. Capuano, M. Aractingi, A. Zouitine, P. Kooijmans, J. Choghari, M. Russi, C. Pascal, S. Palma *et al.*, "Lerobot: An open-source library for end-to-end robot learning," in *The Fourteenth International Conference on Learning Representations*.
[43] H. Bruyninckx, "Open robot control software: the orocos project," in *Proceedings 2001 ICRA. IEEE international conference on robotics and automation (Cat. No. 01CH37164)*, vol. 3. IEEE, 2001, pp. 2523–2528.
[44] M. Quigley, "Ros: an open-source robot operating system. in open-source software," in *workshop of the international con-ference on robotics and automation (ICRA), 2009*, 2009.
[45] G. Bachmann and V. Nagarajan, "The pitfalls of next-token prediction," *arXiv preprint arXiv:2403.06963*, 2024.
[46] 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 *Forty-first International Conference on Machine Learning*, 2024.
[47] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 11 975–11 986.
[48] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.
[49] Z. Hou, T. Zhang, Y. Xiong, H. Pu, C. Zhao, R. Tong, Y. Qiao, J. Dai, and Y. Chen, "Diffusion transformer policy," *arXiv preprint arXiv:2410.15959*, 2024.
[50] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-Rank Adaptation of Large Language Models," *arXiv e-prints*, p. arXiv:2106.09685, Jun. 2021.
[51] P. Manceron, "IKPy." [Online]. Available: https://github.com/Phylliade/ikpy
[52] [Online]. Available: https://droidcam.app/
[53] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, M. Liao, F. Wei, Y. Deng, S. Xu, Y. Zhang *et al.*, "Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation," *arXiv preprint arXiv:2411.19650*, 2024.