

SwiftVLA: Unlocking Spatiotemporal Dynamics for Lightweight VLA Models at Minimal Overhead

Supplementary Material

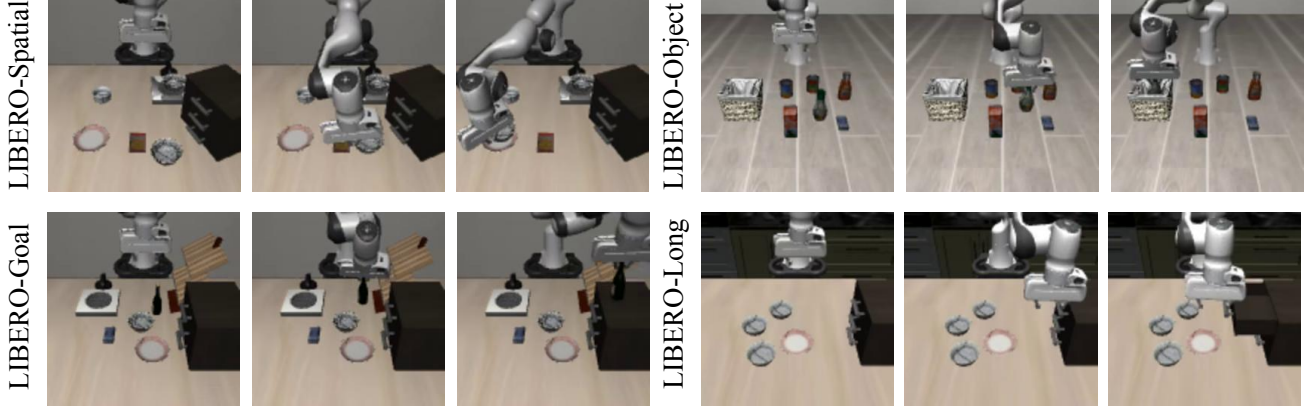


Figure 1. Examples from the LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long task suites [8].

A. Architecture Design.

We adopt SmolVLM [9] as the backbone network for robot environmental perception. SmolVLM [9] utilizes SigLIP [14] to encode visual features, which are then passed to the language decoder. Additionally, by leveraging global image information and employing pixel-shuffle operations, we constrain the number of visual tokens per frame to 64. To further accelerate inference, we skip certain computational layers in the VLM by using only the first 16 layers of the model. To enhance both computational efficiency and response speed, we adopt an attention pattern that alternates between self-attention [3] and cross-attention [2] modules rather than relying solely on either mechanism, following the design principles of SmolVLA [11]. For handling 4D features, we similarly limit the number of tokens to 64, ensuring consistent computational efficiency.

B. Implementation Details.

Baselines. We primarily compare our model with VLA models of different parameter scales, using them as baselines for evaluation.

π_0 [3] is a VLM [1] that incorporates Flow Matching [7] to predict action chunks. With a parameter count of 3.3 billion, it has been trained on a dataset comprising 10,000 hours of cross-embodiment robotics data. The architecture is inspired by Paligemma [1] and processes three images, sensorimotor states, and a language instruction as inputs.

TinyVLA [12] is designed to address the challenges of inference speed and data efficiency in existing VLA models. Unlike traditional models, TinyVLA [12] achieves

faster inference and improved data efficiency by initializing a high-performance multimodal policy backbone and incorporating a diffusion policy decoder during finetuning. With a model size around 1 billion parameters, TinyVLA [12] demonstrates advantages in both speed and data utilization,

SmolVLA [11] is a compact and efficient VLA model designed to reduce training and inference costs, making it suitable for real-world robotics applications. Optimized for consumer-grade GPUs, it retains competitive performance despite its small size. The model is pre-trained on community-collected datasets with fewer than 30k episodes and features an asynchronous inference stack for faster and more responsive control. SmolVLA [11] performs on par with larger models, offering a solution for robotics tasks in both simulated and real-world environments.

Pretraining Details. We pretrain our model on public datasets [4, 13] using a two-stage procedure. In the first stage, the model is trained without 4D inputs, *Fusion Tokens*, or the mask-and-reconstruct strategy, relying solely on robot actions for supervision. Training is performed for 100,000 steps with a global batch size of 256. The learning rate follows a cosine decay schedule, starting at 1×10^{-4} and decaying to 2.5×10^{-6} after a 200-step warm-up. We adopt the AdamW optimizer [6] with $\beta_1 = 0.85$ and $\beta_2 = 0.9$. The input images are resized to 512×512 pixels for compatibility with the vision-language encoder. In the second stage, the model is initialized from the first-stage checkpoint, and 4D inputs, *Fusion Tokens*, are enabled along with the mask-and-reconstruct strategy. Training continues for an additional 50,000 steps under the same optimizer settings, with a reduced learning rate of 5×10^{-5} .

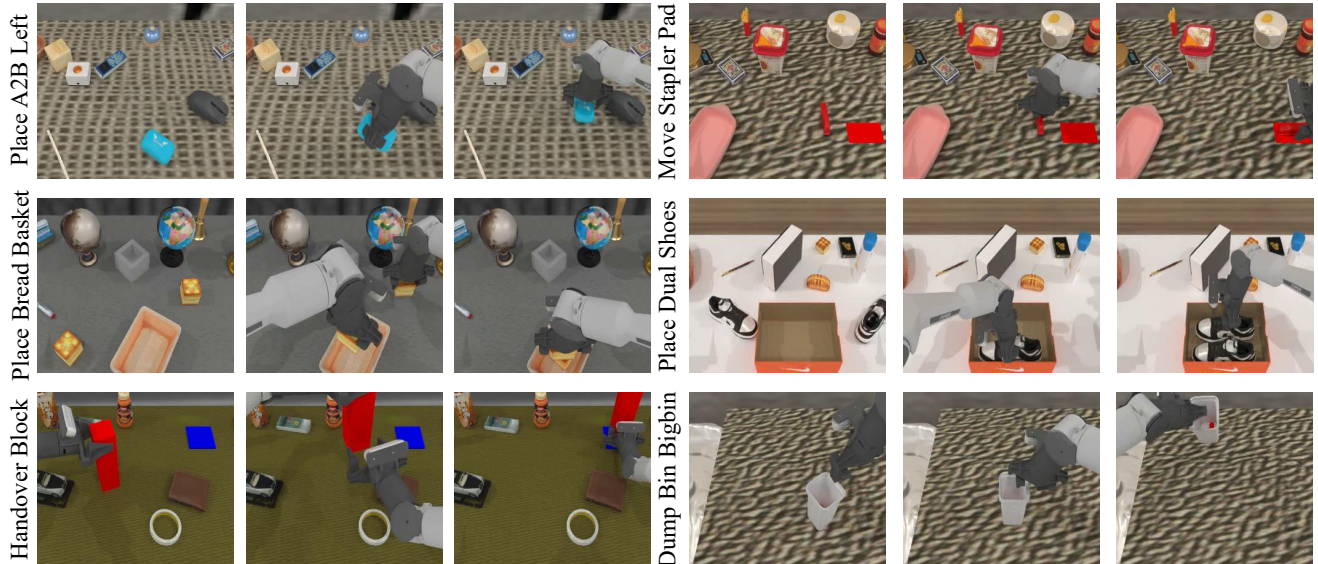


Figure 2. Examples from the RoboTwin 2.0 [5], including move Stapler Pad, Place A2B Left, Place Bread Basket, Place Dual Shoes, Dump Bin Bigbin, Handover Block.

Category	Task	Steps
Short-Horizon	Move Stapler Pad	112
	Place A2B Left	113
Medium-Horizon	Place Bread Basket	151
	Place Dual Shoes	155
Long-Horizon	Dump Bin Bigbin	283
	Handover Block	313

Table 1. Tasks and their step lengths across different horizon categories used in the RoboTwin 2.0 [5].

following a cosine decay schedule.

Finetuning Details. For all baseline methods, we train each model for 30,000 steps on the same dataset, keeping hyperparameters consistent with their original implementations to ensure a fair comparison. For SwiftVLA, we also adopt a two-stage finetuning strategy. In the first stage (the initial 10,000 steps), the model is supervised only with robot actions to stabilize adaptation within the action space. The learning rate follows a cosine decay schedule and is initialized at 1×10^{-4} . We use the AdamW optimizer [6] with $\beta_1 = 0.85$ and $\beta_2 = 0.9$. After completing the first stage, we enable the 4D inputs and *Fusion Tokens*, and incorporate the mask-and-reconstruct strategy in the second stage. This allows the model to further learn spatiotemporal feature fusion and higher-level structural understanding.

Simulation Tasks Setup. As shown in Fig. 1 and Fig. 2, we present the simulation tasks from LIBERO and RoboTwin 2.0 [5]. In RoboTwin 2.0, we further categorize the evalu-

ation tasks into short, medium, and long horizons based on the average number of steps required for completion. Tab. 1 provides the detailed categorization. Short-horizon tasks typically require fewer than 120 steps and rely primarily on localized spatial reasoning. Medium-horizon tasks average around 150 steps and involve sequential planning across multiple object interactions. Long-horizon tasks exceed 280 steps and exhibit higher temporal dependencies and compositional complexity. This categorization enables a systematic analysis of how different models generalize across varying horizon lengths, which is crucial for assessing robustness in multi-step manipulation scenarios.

Real-World Tasks Setup. As shown in Fig. 3 and Fig. 4, we illustrate the tasks used in our experiments, covering four manipulation tasks.

Clean the Desk: Bowls and plates with randomized colors are placed on the table. The robot must place both items into a basket while ensuring that the plate is positioned at the bottom.

Throw the Bottle: A plastic bottle with a randomly varying amount of liquid is placed in the scene, and the robot is required to pick it up and throw it into a trash bin.

Stack Bowls: Two bowls are positioned randomly on the table, and the robot is required to stack them correctly.

Fold the Cloth: A piece of clothing is laid flat on the table. The robot folds it following a predefined sequence and then moves the folded garment to a designated location.

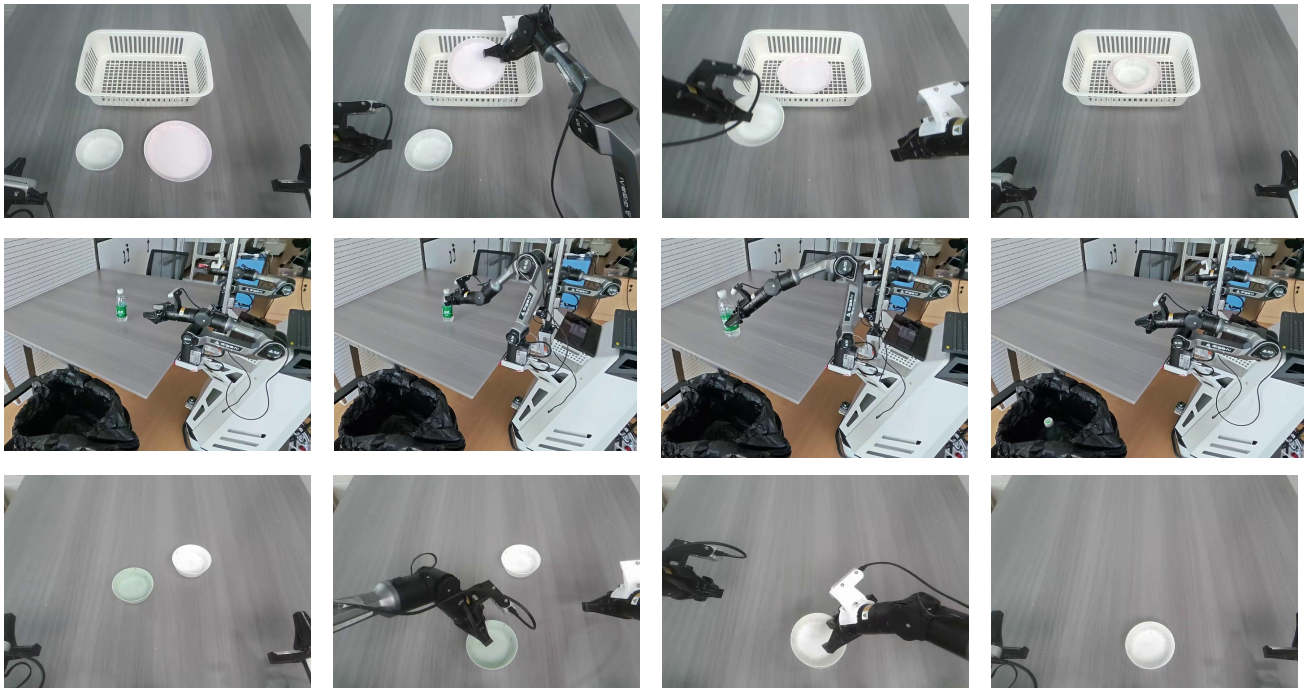


Figure 3. Real-world manipulation tasks used in our experiments. From top to bottom, the examples correspond to Clean the Desk, Throw the Bottle, and Stack Bowls.

Methods	Fold the Cloth	
	SR \uparrow	Length \downarrow
π_0 [3]	0.45	2550
SmolVLA [11]	0.05	3200
SmolVLA [†] [11]	0.30	2600
SwiftVLA	<u>0.60</u>	<u>2100</u>
SwiftVLA with 4D input	0.65	2010

Table 2. Comparison of task success rate and trajectory length for “Fold the Cloth”. The best results are marked in **bold**, and the second-best results are underlined. [†] denotes the model that is pre-trained and fine-tuned using the same configuration as SwiftVLA.

C. More Challenging Real-World Experimental Results.

To further evaluate the real-world performance of SwiftVLA, we investigate a more challenging manipulation task: Fold the Cloth. This task is difficult due to its long-horizon nature and the complex physical dynamics of deformable objects. As shown in Fig. 4, we illustrate the full execution process of this task in a real-world setting.

The results in Tab. 2 present a comparison of success rates achieved by different methods on the cloth folding task, executed on the AgileX PiPER six-degree-of-freedom robotic arm with computational support provided by an

NVIDIA RTX 4090 GPU. SwiftVLA demonstrates strong and reliable performance, while similar models such as SmolVLA [11] achieve very low success rates. These results highlight the advantages of incorporating 4D features when handling deformable objects and long-horizon manipulation tasks.

D. Supplementary Video

We provide a video that compares SwiftVLA and π_0 [3] across multiple tasks. Please refer to the file located at [video/comparison.mp4](#) for more details. The video consists of the following segments:

- **4-40s:** Demonstrates the comparison between SwiftVLA and π_0 [3] on the “Fold the Cloth” task using an NVIDIA Jetson Orin platform [10].
- **40-60s:** Displays the comparison between SwiftVLA and π_0 [3] on the “Throw the Bottle” task using an NVIDIA Jetson Orin platform [10].
- **60-72s:** Compares SwiftVLA and π_0 [3] on the “Clean the Desk” task using an NVIDIA Jetson Orin [10].
- **72-90s:** Highlights the superior error-correction capability of SwiftVLA over π_0 is particularly evident when handling deformable objects. In the video, we compare the two algorithms on the “Fold the Cloth” task using an NVIDIA Jetson Orin platform [10], focusing on how each model adjusts after failure. Compared to π_0 , SwiftVLA

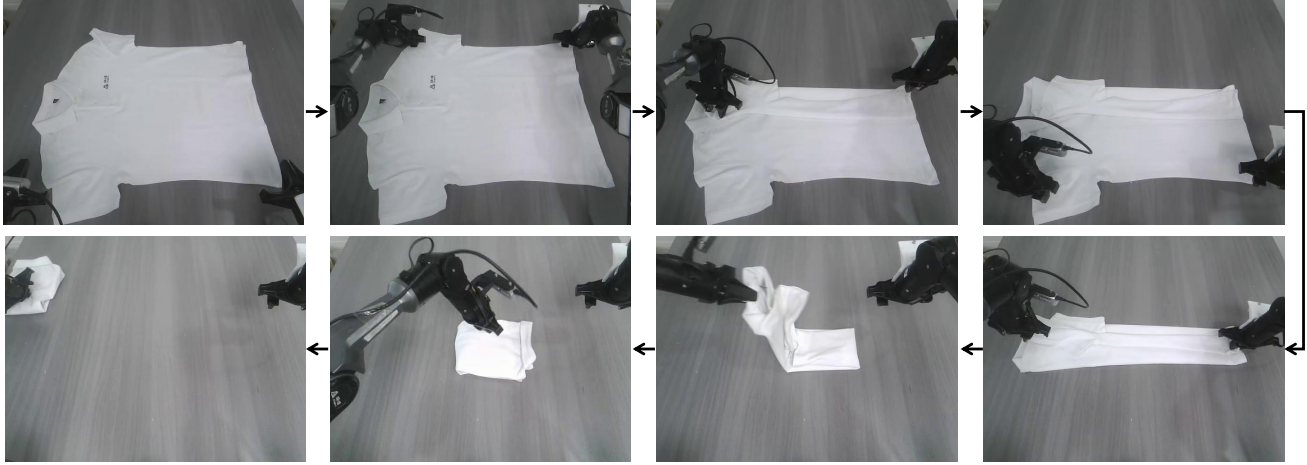


Figure 4. Real-world execution process of the Fold the Cloth task, which requires long-horizon reasoning and precise manipulation of deformable objects.

recovers more quickly, with smoother motion trajectories that enable more fluid and accurate handling of deformable objects.

- **90-132s:** Shows additional examples of SwiftVLA on the “Fold the Cloth” task using an NVIDIA Jetson Orin [10].

References

- [1] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarelli, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 1
- [2] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025. 1
- [3] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. pi0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 1, 3
- [4] Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, et al. Agibot world colosseum: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*, 2025. 1
- [5] Tianxing Chen, Zanzin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Qiwei Liang, Zixuan Li, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025. 2
- [6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1, 2
- [7] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 1
- [8] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023. 1
- [9] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025. 1
- [10] NVIDIA Jetson Orin Series Technical Brief. NVIDIA Corporation, 2022. Technical Brief v1.2, TB_10749-001_v1.2. 3, 4
- [11] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025. 1, 3
- [12] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Zhibin Tang, Kun Wu, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *IEEE Robotics and Automation Letters*, 2025. 1
- [13] Kun Wu, Chengkai Hou, Jiaming Liu, Zhengping Che, Xiaozhu Ju, Zhuqin Yang, Meng Li, Yinyu Zhao, Zhiyuan Xu, Guang Yang, et al. Robomind: Benchmark on multi-embodiment intelligence normative data for robot manipulation. *arXiv preprint arXiv:2412.13877*, 2024. 1
- [14] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 1