Drift-VLA: Fast Vision–Language–Action Policies with One-Step Drifting

Xingdong Zuo
Independent Project

July 5, 2026

Abstract

We present Drift-VLA, a simple method that trains a vision–language–action (VLA) policy to generate action chunks in a single forward pass. Flow-matching VLA policies such as SmolVLA produce an action chunk by integrating a learned velocity field over ten evaluations of the action expert, and this iterative sampling dominates inference latency. Instead of distilling a trained multi-step policy into a faster student, we train the one-step policy directly with a drifting objective: no teacher, no second training stage, and no new parameters. The main design question turns out to be what distance the drift loss should use over padded, heterogeneous action chunks. We find that matching each action dimension's chunk-long trajectory under its own scale is the only variant that is both strong and stable across training seeds; we call this grouping action-dimension drift. On LIBERO-Spatial, Drift-VLA reaches $90.2 \pm 1.3\%$ success with one expert pass, matching a seed-replicated ten-step flow-matching control ($88.0 \pm 1.5\%$) at 4.6× lower measured latency. The same recipe deploys on a low-cost real-world mobile manipulator, where ten repeated pick-and-place trials score 9/10, versus 7/10 for ten-step and 3/10 for one-step flow matching. Project page: `https://zuoxingdong.github.io/drift-vla`

1 Introduction

Modern VLAs generate action chunks with diffusion or flow-matching heads [2, 4, 13]. These iterative samplers are expressive, but they are slow: every action chunk costs several forward passes of the action expert, which limits the closed-loop control rate and stalls the robot at chunk boundaries [3]. This matters most for small VLAs intended for edge deployment. The standard fix is distillation: compress a trained multi-step teacher into a one- or few-step student [7, 9, 12, 14]. But distillation adds a second training stage, and the student inherits a teacher–student gap.

We ask a simpler question: can we train the one-step VLA policy directly? We take SmolVLA [13] and replace its flow-matching (FM) objective with the drifting objective of Deng et al. [6], which trains one-step generators by distribution matching. The change is a configuration-gated patch. The architecture, action interface, and checkpoint format are unchanged; the diffusion-style time input is pinned to a constant; and the same stochastic generator is used at training and deployment, so there is no train/test gap to distill away.

Making this work is not just an objective swap. The drift loss is built from distances between sampled action chunks, and a SmolVLA chunk is a padded 50×32 tensor that mixes end-effector translation, rotation, and gripper channels. The central design question is what should own a distance scale. Chunk-level matching lets large-magnitude coordinates dominate. Timestep matching ignores temporal structure within each coordinate. Action-dimension matching gives each action dimension's chunk-long trajectory its own scale, and it is the only variant we find to be both strong and replicable across training seeds.

We evaluate Drift-VLA under a controlled protocol on LIBERO-Spatial, with binomial, evaluation-seed, and training-seed noise floors stated up front. Drift-VLA matches ten-step flow matching at one tenth the network function evaluations (NFEs) and 4.6× lower measured latency. All headline claims are parity claims, not superiority claims.

2 Method

Flow matching baseline. For a demonstration chunk a and Gaussian noise ε, FM trains a velocity along $x_t = t\varepsilon + (1-t)a$:

$$\mathcal{L}_{\text{FM}} = \|v_\theta(x_t, t) - (\varepsilon - a)\|^2.$$

At inference, SmolVLA integrates this field over ten expert forwards. The iteration cannot simply be dropped at test time: truncating the best FM checkpoint from ten steps to one costs real success (88.0% → 84.0%).

Table 1: LIBERO-Spatial success (%). Means are over three evaluation seeds. NFE counts action expert forwards per chunk; latency is measured with the evaluated checkpoints on the same GPU for all rows.

Policy	Drift grouping	NFE	20k	30k	ms/chunk
FM, best grid[†]	–	10	–	88.0 ± 1.5	232.9
same checkpoint, truncated	–	1	–	84.0 ± 2.0	50.1
Public `smolvla_libero`[°]	–	1	–	80.0	–
Drift	timestep	1	78.8 ± 2.7	81.3 ± 3.1	≈50.6
Drift	chunk-level	1	86.5 ± 2.1	77.8 ± 1.6	≈50.6
Drift-VLA	**action-dimension**	**1**	**89.0 ± 1.4**	**90.2 ± 1.3**	**50.6**
+ KeyStone (K=8, C=4)	action-dimension	8×1	–	92.3 ± 0.5[§]	53.5

[°] single evaluation seed. [†] Best batch×LR FM configuration under the same protocol; independent training seeds for this setting score 86.0/93.0/89.0. [§] cluster-medoid selection over one-step samples; an independent three-seed re-evaluation scores 90.7 ± 1.7, so the gain over the 90.2 base is not statistically resolved.

One-step drifting. Drifting trains the generator directly. For each observation, the policy samples G candidate chunks from one expanded VLM prefix KV-cache. For one matching problem with candidates $\{g_1, \ldots, g_G\}$ and demonstration a, we compute pairwise distances, normalize by the mean distance s, and form temperature-indexed row/column-coupled affinities (products of row- and column-softmaxes of the scaled distances; exact form in Deng et al. [6]). The affinities induce a mean-shift direction V: the demonstration attracts each live sample, and detached sibling samples repel it. The loss regresses live samples to a detached drift target,

$$\mathcal{L}_{\text{drift}} = \|g/s - \operatorname{sg}(g/s + V)\|^2, \qquad \frac{\partial \mathcal{L}}{\partial g} = -\frac{2}{s}V, \tag{1}$$

where s and V are computed under stop-gradient. Optimization moves every sample along $+V$, toward the demonstration and away from its siblings, while deployment stays a single generator call.

Drift decomposition. Let sampled actions be $B \times G \times T \times D$, with $T = 50$ chunk steps and $D = 7$ real LIBERO action dimensions after masking padded timesteps and dimensions. There are three natural ways to instantiate the matching problem. Timestep drift: one problem in $\mathbb{R}^D$ per chunk step. Chunk-level drift: one problem in $\mathbb{R}^{TD}$ over the complete chunk. Action-dimension drift: one problem per action dimension over its length-T trajectory. Action-dimension matching gives each physical channel its own distance scale and force normalization while still coupling the G samples across the whole horizon. We compute the drift target with autocast disabled: in ablation this fp32 guard is worth 8.2 success points for chunk-level matching at the 20k checkpoint (86.5 vs. 78.3 without it) while moving action-dimension drift by about one point.

Test-time selection. One-step sampling makes best-of-K inference nearly free: K candidate chunks cost one batched expert pass on the expanded prefix cache. We evaluate the geometry-guided selector of Dai et al. [5] (KeyStone): k-means over the K flattened chunks (C clusters), execute the medoid of the dominant cluster, with a unimodality guard falling back to the global medoid.

3 Experiments

Protocol. We evaluate on LIBERO-Spatial [11]: 10 tasks, 20 episodes per task, 200 episodes per evaluation, official `lerobot` evaluator, 256^2 observations, relative control, and one executed action per prediction. All trained policies are fresh SmolVLA instances, not `smolvla_base` fine-tunes, with a frozen pretrained 500M-class VLM and a from-scratch action expert, trained on the full `lerobot/libero` dataset at a pinned revision for 30k updates. Effective batch is 63–64, LR is 10^{-4}, and drift uses $G = 8$, temperatures $\{0.02, 0.05, 0.2\}$. Unless stated, means and standard deviations are over three evaluation seeds on a fixed checkpoint. A 200-episode estimate has binomial SE about 3.5 points; evaluation seeds add 1–4 points; retraining the FM control spans 86–93%. Latencies are medians of repeated rounds on one GPU, identical for all rows.

Does one-step drifting match ten-step flow matching? Yes, within our measured noise floors. Drift-VLA reaches 90.2 ± 1.3% at 30k with one expert forward per chunk (Table 1, Fig. 1). The strongest ten-step FM model we trained scores 88.0 ± 1.5%, and its independent training-seed replicates span 86–93%; the drift result sits inside that range, so we claim parity at one tenth the NFE, not a resolved improvement. The wall-clock gain is smaller than the NFE ratio because the VLM prefix is shared: latency decomposes into about 30 ms of prefix encoding plus 20 ms per expert pass, giving 50.6 ms/chunk for Drift-VLA versus 232.9 ms/chunk for ten-step FM (4.6×). The speedup is therefore hardware-dependent: the larger the prefix share, the smaller the ratio. A thermally limited laptop-class GPU gives 1.9×. Truncating FM to the same one-NFE budget costs 4.0 points; Drift-VLA avoids that loss.

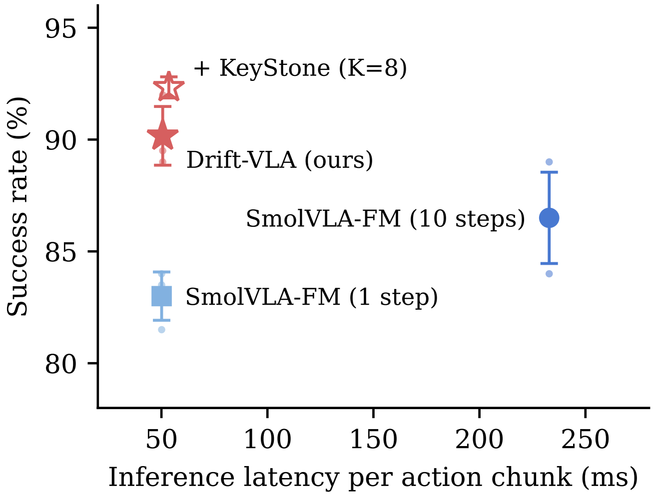

Figure 1: Success vs. measured per-chunk latency (30k checkpoints; three evaluation seeds, small dots = seeds). Action-dimension drift reaches ten-step FM quality at one expert forward pass and 4.6× lower latency; truncated one-step FM does not.

Which drift grouping works? Only action-dimension matching is both strong and stable (Table 1, bottom). Timestep drift stays in the low 80s. Chunk-level drift is strong at 20k but collapses by 30k. Action-dimension drift is high at both checkpoints, and a second end-to-end training seed reproduces it within one point at both, while timestep drift varies by 5.5–9.5 points across training seeds. The training-seed floor in this benchmark is large enough that stability is part of the result. The practical recommendation is simple: for padded, heterogeneous VLA action chunks, give each real action dimension its own drift metric scale.

What does test-time selection add? A little, at almost no cost. On the 30k winner, KeyStone with K=8, C=4 costs 2.9 ms (53.5 ms/chunk, still 4.4× below ten-step FM). It scores 92.3 ± 0.5% in one three-seed evaluation and 90.7 ± 1.7% in an independent re-run. This is consistent with a small gain over the 90.2 base, but it does not clear the seed-noise bar, so we do not claim it as an improvement.

Does it run on a real robot? Yes, qualitatively. We train the same action-dimension drift recipe (G=8) on teleoperated demonstrations of a shopping task with LeKiwi, a low-cost open-source mobile manipulator: pick a requested item and place it into a basket, from three RGB cameras. The platform mixes arm-joint, gripper, and base-velocity channels, which is exactly the unit heterogeneity that motivates action-dimension drift. On the robot's laptop-class GPU, a one-step chunk costs 132 ms versus 246 ms for ten-step FM (the 1.9× ratio above), and KeyStone selection over eight one-step samples costs 172 ms, still below a single ten-step chunk; the faster chunks give the policy more frequent correction opportunities at chunk boundaries. On the salt-bottle item we repeated the trial ten times per policy from the same initial scene: Drift-VLA with KeyStone succeeds 9/10, ten-step FM 7/10, and truncated one-step FM 3/10. Ten trials resolve only large gaps, so we read this as a clear win over FM at the matched one-pass budget and rough parity with ten-step FM, consistent with the simulation results. Side-by-side rollouts on all three items are on the project page.

4 Related Work

Diffusion and flow-matching action heads are standard in visuomotor policies and VLAs [1, 2, 4, 10, 13]. Acceleration is usually obtained by distillation or consistency-style training [7, 9, 12, 14]; drifting instead trains the one-step generator directly [6]. DBP/DBPO concurrently applies drift-style one-step learning to robot policies and studies chunk- and step-wise drift spaces [8]. Our work differs in the VLA setting and in isolating a third decomposition, action-dimension drift, which is motivated by heterogeneous padded action coordinates and is the stable choice in our SmolVLA controlled comparison.

5 Limitations and Conclusion

The evidence is intentionally scoped: one simulated suite (LIBERO-Spatial), one SmolVLA configuration, 200-episode evaluations, three evaluation seeds, and two training seeds for the winning variant. Real-robot evidence is one task with ten rollouts per policy, plus qualitative rollouts on two further items. We do not compare head-to-head against a full distillation pipeline, and the latency ratio is hardware-dependent: it shrinks as the shared prefix dominates. One mechanism remains open: part of the action-dimension variant's gain over an earlier loop implementation of the same loss is not explained by the fp32 target guard, which fully accounts for the chunk-level gain; because the action-dimension gain replicates across training seeds and checkpoints, we scope all claims to the exact released kernel.

Within this scope, the recipe is simple: one additive drift-loss patch turns SmolVLA into a native one-step policy at ten-step flow-matching quality, with 4.6× lower measured latency and no distillation. Code, patches, per-seed logs, and winning checkpoints are archived for exact reproduction.

References

[1] Johan Bjorck, Fernando Castaneda, 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.

[2] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[3] Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint arXiv:2506.07339*, 2025.

[4] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Robotics: Science and Systems (RSS)*, 2023.

[5] Yinwei Dai, Zhuofu Chen, Lijie Yang, and Ravi Netravali. Geometry guided self-consistency for physical AI. *arXiv preprint arXiv:2605.08638*, 2026.

[6] Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

[7] Kevin Frans, Danijar Hafner, Sergey Levine, and Pieter Abbeel. One step diffusion via shortcut models. In *International Conference on Learning Representations (ICLR)*, 2025.

[8] Yuxuan Gao, Yedong Shen, Shiqi Zhang, Wenhao Yu, Yifan Duan, Jia Pan, Jiajia Wu, Jiajun Deng, and Yanyong Zhang. Drift-based policy optimization: Native one-step policy learning for online robot control. *arXiv preprint arXiv:2604.03540*, 2026.

[9] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J. Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

[10] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. OpenVLA: An open-source vision-language-action model. In *Conference on Robot Learning (CoRL)*, 2024.

[11] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

[12] Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. In *Robotics: Science and Systems (RSS)*, 2024.

[13] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, and Remi Cadene. SmolVLA: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025.

[14] Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.