How Much Smoothness Do Few-Step Generative Robot Policies Give Up? A Controlled NFE Audit

Yufu Dong
College of Artificial Intelligence, Nankai University
yf_dong@mail.nankai.edu.cn

***Abstract*—Few-step generative robot policies reduce inference cost, but the executed motion still has to remain usable. Most work on few-step policies, including one-step diffusion, consistency models, and MeanFlow variants, reports task success and inference cost (NFE), but not the quality of the executed path. We audit this gap. Reusing three movement-smoothness measures from biomechanics (mean-squared jerk, log-dimensionless jerk, and spectral arc length), we sweep the number of inference steps for four generative-policy families with matched Transformer architectures, identical demonstrations, and a shared evaluation protocol, then plot smoothness against NFE on Push-T and three MetaWorld tasks (3 seeds, 95% CI). The cost of cutting steps depends strongly on *how* the policy generates actions: a DDIM diffusion policy does not form a usable one-step rollout (Push-T coverage 0.09, the highest mean-squared jerk we measured), whereas a standalone flow-matching policy loses little (coverage 0.89, the lowest measured roughness). Smoothness separates the families most sharply at low NFE, where success rate separates them least, so it carries information a success-only protocol discards. We also test a minimal mitigation, a jerk penalty that cuts one-step jerk by an order of magnitude at a task-dependent success cost. Roughness does not by itself predict failure, which suggests that smoothness should be reported separately rather than treated as a success proxy. We recommend reporting a smoothness curve and a per-method "NFE budget," and make the audit code available.**

I. Introduction

Generative policies have become a default choice for visuomotor imitation: a diffusion or flow model turns a noise sample into a short chunk of future actions [4, 10]. Their one drawback is speed. Sampling is iterative, and a control loop cannot always afford ten or fifty network evaluations per chunk. The response has been a wave of *few-step* methods: consistency distillation, one-step diffusion policies, and MeanFlow policies that learn an interval-averaged velocity and generate in a single evaluation [12, 23, 6, 16, 17]. Papers in this line are judged on two axes: does the task succeed, and how many function evaluations (NFE) does it take.

We think a third axis is missing. Success rate says *whether* the block reached the target; it says nothing about *how* the arm got there. Two policies with the same success rate can trace very different paths (Fig. 2), and a jerky one wears actuators, trips safety limits, and transfers poorly to hardware. Smoothness of motion is not a vague notion. Rehabilitation and biomechanics have measured it for decades with jerk-based scores and spectral arc length [1]. It is simply rarely applied to generative policies, and we could find no work that tracks it *as a function of the number of inference steps* across policy families.

We run that controlled audit. We vary the generative family and the inference step count, while holding the rest of the protocol fixed, and measure both success and smoothness:

- We set up a controlled NFE sweep with matched Transformer architectures, identical demonstrations, and the same evaluation episodes and sampling noise across every NFE. Within each family, any difference is attributable to step count.
- We find the few-step cost is strongly family-dependent. DDIM performs poorly at one step; a standalone flow-matching policy barely changes; MeanFlow sits between. The pattern holds on Push-T and replicates on MetaWorld.
- We show smoothness is the more discriminative metric at low NFE, and argue for reporting it alongside a per-method NFE budget. We release the code and audit harness.
- We characterize a minimal mitigation, a jerk penalty, that cuts one-step jerk by $20\times$ at a task-dependent success cost, and show chunk-roughness does not reliably predict failure: smoothness is a separate axis, not a success proxy.

Code and artifacts. The audit code and public artifacts are available in the public repository.

This is a measurement study, not a new sampler. Its value is in what the measurement reveals.

II. Related Work

Cutting the step count of generative models. The push toward few-step generation predates policies and falls into a few recognizable families. One reduces the number of ODE-solver steps for a fixed diffusion model (DDIM [20]). A second distills a multi-step model into one or few steps, as in consistency models [21] and their variants. A third straightens the generative path so that few Euler steps suffice: rectified flow [11], flow matching [10], and shortcut models [5]. A fourth learns the jump directly, as in MeanFlow's interval-averaged velocity [6, 7]. The four samplers we audit are chosen as clean representatives of these paradigms (DDIM; flow matching; MeanFlow), with matched Transformer architectures and identical training and evaluation protocols.

Few-step generative policies. Robotics has adopted each paradigm. Diffusion Policy [4] introduced iterative action-chunk generation; consistency distillation yields Consistency

Policy [14] and the point-cloud ManiCM [12]; diffusion distillation yields one-step policies [23]; flow matching underlies recent action experts; and MeanFlow policies such as MP1 [16] and movement-primitive distillation such as FRMD [17] target single-step control. These works report task success and inference cost almost exclusively. Smoothness, where it appears, is an asserted benefit rather than a quantity measured as the step count is varied. A parallel and very recent line directly targets the smoothness of chunked policies: real-time chunking across chunk boundaries [2], lightweight post-hoc jerk minimization [19], and temporally-grounded priors that straighten the generative flow [8]. These propose fixes, but, like the acceleration methods above, evaluate them on success and latency rather than on a smoothness-versus-NFE curve. We therefore audit rather than add to them, and our jerk penalty (§IV-D) is included as a deliberately minimal baseline, not a competitor.

Evaluating beyond success rate. The case that success rate is too coarse is gaining ground. RoboEval [22] instruments manipulation benchmarks with jerk-based behavioral metrics and shows they stay discriminative after binary success saturates. This is the closest work to ours in spirit. Recent evaluation methodology also argues for statistically rigorous, sample-efficient comparison that looks past a single binary number [18]. Smoothness has also been used to rank *demonstrations* for data curation [9], and to regularize policies in continuous control [13]; FRMD [17] reports a curvature count, and CoLA-Flow [24] a jerk reduction, each for a single method at a fixed step count. A separate line does vary the inference budget and finds it matters: extra Euler steps can *degrade* a flow policy [3], and a fine-tuned policy's return tracks its NFE [15]. These track success or return, not kinematic smoothness. No prior work puts NFE on one axis and a kinematic smoothness metric on the other, across generative families under one controlled protocol. Our work is complementary to RoboEval: RoboEval makes the broader case that robot policies should be evaluated beyond binary success, while we ask a narrower, step-budgeted question—*how* trajectory quality changes as few-step generative policies reduce NFE, and whether that tradeoff differs across generative families.

III. Method: A Controlled NFE Audit

Four families, matched architecture. Every policy is an observation-conditioned Transformer over action chunks (Fig. 1). Only the generative formulation and sampler change: (1) *MeanFlow (u-head)*, an interval-averaged velocity sampled in few Euler steps [6]; (2) *Flow-Matching (v-head)*, the instantaneous-velocity head of the same MeanFlow network, integrated over many steps as a same-weights control; (3) *Flow-Matching (standalone)*, an independently trained straight-path flow-matching policy; (4) *Diffusion Policy (DDIM)*, an ϵ-prediction DDPM sampled by DDIM. The u/v pair shares weights. The standalone flow-matching and DDIM models use the same Transformer architecture, data budget, and evaluation protocol, so method differences are not caused by a larger model or a different benchmark split.

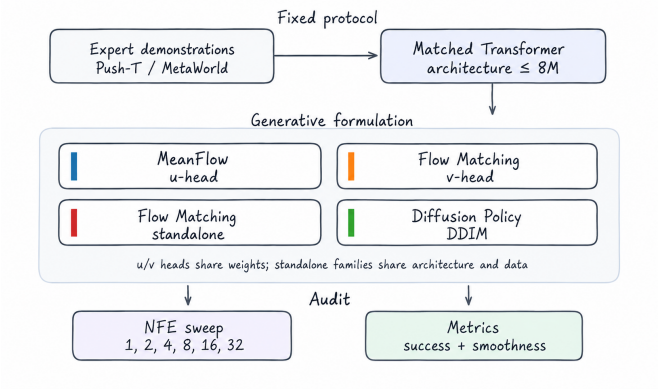

Fig. 1. Controlled NFE audit. We fix the data, architecture, seeds, and evaluation episodes, compare four generative formulations, and report both task success and executed-path smoothness.

What we hold fixed. For each family and each NFE in $\{1, 2, 4, 8, 16, 32\}$ we run the same evaluation episodes, with the same environment seeds and the same per-chunk sampling noise. We evaluate 50 Push-T episodes and 20 episodes per MetaWorld task for each seed and NFE. Step count is then the only thing that varies within a family. We train three seeds per family and report the mean with a 95% Student-t interval.

Smoothness. On the executed end-effector path we compute three measures from the motor-control literature [1]: mean-squared jerk (MSJ), log-dimensionless jerk (LDLJ), and spectral arc length (SAL), where a less negative SAL means smoother motion. The metric code reads only the trajectory, so it is identical across families.

Benchmarks. Push-T (low-dim) [4] gives a 2-D pushing task with 206 demonstrations. We add three MetaWorld tasks, reach, push, and assembly [25], spanning a short reach to a long-horizon peg insertion, collecting expert demonstrations from the simulator's scripted policies. Models are small ($\leq$8M parameters) and train on a single GPU.

IV. Results

A. Push-T: the few-step cost depends on the method

Table I and Fig. 3 sweep NFE for all four families. At one step the families differ sharply. DDIM fails under this protocol: it covers 0.09 of the target and has the largest measured jerk in the study. The standalone flow-matching policy, by contrast, already reaches 0.89 coverage with the lowest-roughness trajectories, and changes little with more steps. This is consistent with the straight-path view that a well-learned velocity field integrates well in a single jump. MeanFlow's average-velocity head lands in between.

The sweep gives two observations. First, every family's smoothness improves and then saturates with more steps, but *where* it saturates, and how badly it fails below that, is method-specific. Second, the families fan out far more on smoothness than on success at low NFE: SAL ranges from -48 (MeanFlow, one step) to -9 (standalone FM), while the coverage gap is comparatively muted. Smoothness is the more sensitive readout of "too few steps." The same-weights

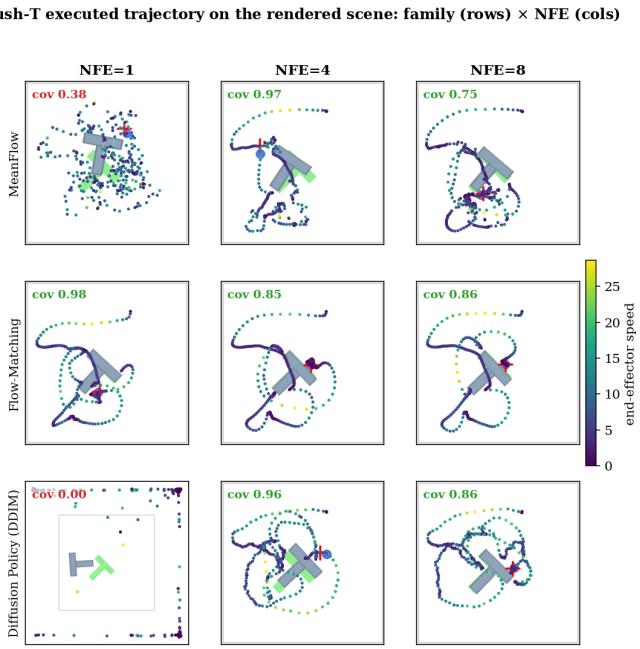

Fig. 2. Push-T rollouts for one fixed start state: generative family (rows) × inference steps (columns). The selected seed illustrates the same low-NFE ordering as the aggregate curves: standalone flow matching succeeds at one step, DDIM fails at one step and recovers with more steps, and MeanFlow lies between.

TABLE I

PUSH-T (3 SEEDS). COVERAGE AT ONE STEP AND A REPRESENTATIVE HIGH-NFE PLATEAU POINT. DDIM HAS LOW COVERAGE AT NFE=1; STANDALONE FLOW-MATCHING CHANGES LITTLE AND HAS THE SMOOTHEST HIGH-NFE PLATEAU AMONG THE AUDITED FAMILIES.

Family	NFE=1 cov	plateau cov	plateau SAL
MeanFlow (u-head)	0.28	0.81	−19.3
Flow-Matching (v-head)	0.52	0.80	−20.2
Diffusion Policy (DDIM)	0.09	0.92	−12.6
Flow-Matching (std.)	**0.89**	**0.93**	**−9.1**

u-vs-v comparison isolates this to the sampler: integrating the instantaneous velocity is smoother at low NFE than the interval-averaged one, with the network weights held fixed. Figure 2 makes the cost concrete across families: standalone flow matching already covers the target at one step, DDIM fails at one step and recovers with more evaluations, and MeanFlow sits between.

B. MetaWorld: the pattern generalizes, and the jerk gap tracks task length

We repeat the audit on three MetaWorld tasks of increasing horizon, reach, push, and assembly, to test whether the Push-T ordering is an artifact of one environment. Figure 4 sweeps NFE for all four families on each task, and Table II gives the single-step operating point.

The MetaWorld tasks reproduce the Push-T ordering. On every task the DDIM diffusion policy has zero one-step success and the most negative SAL, recovering only once it is given two to four steps. The standalone flow-matching

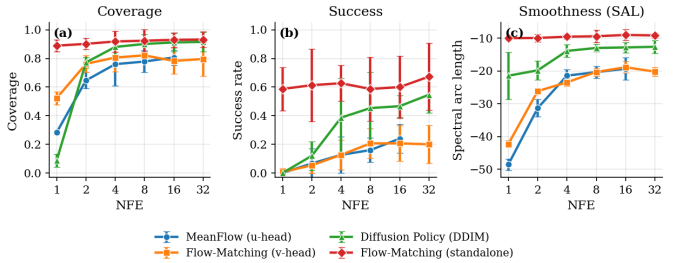

Fig. 3. Push-T. Coverage, success, and spectral arc length against NFE for four families (3 seeds, 95% CI). The few-step cost and the smoothness gap between families grow sharply as NFE drops, and differ by method.

TABLE II

METAWORLD AT NFE=1 (3 SEEDS): SUCCESS RATE / SPECTRAL ARC LENGTH. DDIM HAS ZERO SUCCESS AND THE LOWEST SAL ON EVERY TASK, AND ITS ONE-STEP ROUGHNESS DEEPENS WITH TASK HORIZON ($-7.0 \rightarrow -8.4 \rightarrow -11.6$). STANDALONE FLOW-MATCHING HAS THE HIGHEST ONE-STEP SUCCESS AND SAL.

Family	reach S / SAL	push S / SAL	assembly S / SAL
MeanFlow (u-head)	0.80/−4.1	0.92/−5.0	0.72/−7.2
Flow-Matching (v-head)	0.50/−4.2	0.92/−5.4	0.93/−7.1
Diffusion Policy (DDIM)	0.00/−7.0	0.00/−8.4	0.00/−11.6
Flow-Matching (std.)	**0.82/−4.0**	**0.93/−5.1**	**0.95/−6.6**

policy, by contrast, already succeeds at a single evaluation ($0.82/0.93/0.95$ on reach/push/assembly) and has the highest SAL among the audited families, with MeanFlow and the shared-weights v-head in between.

What the longer tasks add is a clear scaling. DDIM's one-step spectral arc length deepens from -7.0 on the short reach to -8.4 on push and -11.6 on the long assembly insertion, while the well-conditioned families stay essentially flat across NFE. The rougher the few-step sampler and the longer the motion, the more a success-only protocol misses. This is the regime where a smoothness axis is most useful. Above NFE ≥ 2 to 4 success saturates and the families' smoothness converges, so the discriminative signal lives at the low step counts these methods are built for. Figure 5 shows the rollouts in the simulator across families and budgets: only the DDIM policy fails at one step.

C. Cross-environment summary

Across four environments, reducing NFE changes behavior in a method-specific way rather than following a single step-count curve. A DDIM diffusion policy needs a non-trivial budget before it is either successful or smooth. It has zero or near-zero task score at one step everywhere and is the highest-roughness family throughout the low-NFE regime, whereas a straight-path flow-matching policy reaches the audited success and smoothness thresholds at a single evaluation on all four tasks. A useful way to report this is a per-method *NFE budget*: the smallest audited step count at which both task score and smoothness have reached the high-NFE plateau (Table III). Operationally, we mark an NFE as saturated when its task score (coverage on Push-T, success on MetaWorld) is within

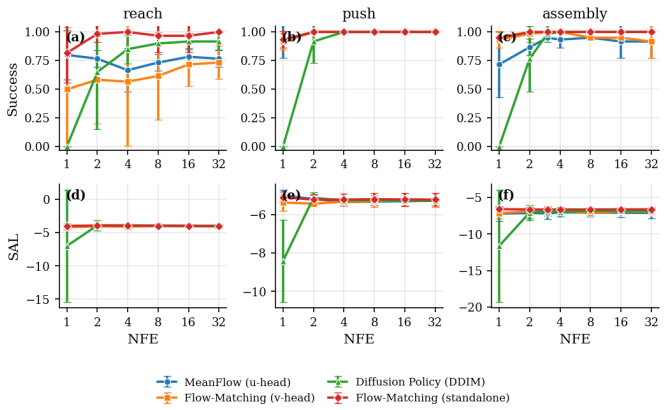

Fig. 4. MetaWorld reach, push, and assembly (3 seeds, 95% CI). Top: success rate vs. NFE; bottom: spectral arc length (higher = smoother) vs. NFE. The DDIM policy (green) has zero success and a jerk spike at one step on every task, and the one-step jerk spike deepens from reach to assembly; standalone flow-matching (red) maintains high success and has the highest SAL. The family ordering matches Push-T.

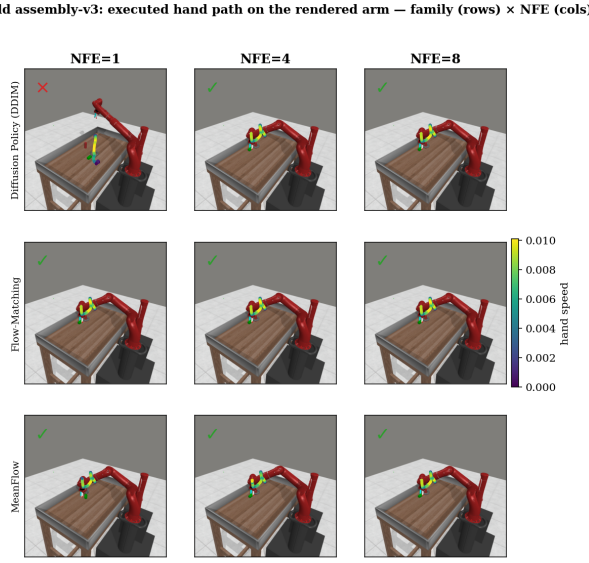

Fig. 5. MetaWorld assembly rollouts in the simulator: generative family (rows) × inference steps (columns). Each cell overlays the executed end-effector path (colored by speed) on the rendered Sawyer arm, with a green ✓ / red × for task success. Only the DDIM diffusion policy fails at one step (×) and recovers by NFE=4; flow-matching and MeanFlow succeed at every budget. The family-dependent few-step cost is visible in the rollouts.

5 percentage points of that family's best observed score and its SAL is within 3 units of that family's best observed SAL, on every environment. This summarizes the curves in one auditable number, and it differs across families by $4\times$ even when their saturated task scores are indistinguishable. A success-only summary would hide this difference.

D. A simple mitigation and its tradeoff

The audit suggests a small control experiment: if few-step samples are rough, can a cheap change buy smoothness back? We add a jerk penalty to the training loss, an SNR-weighted squared third-difference of the predicted clean action chunk,

TABLE III

PER-METHOD NFE BUDGET: THE SMALLEST AUDITED STEP COUNT AT WHICH BOTH TASK SCORE AND SMOOTHNESS REACH THE HIGH-NFE PLATEAU, TAKEN ACROSS THE FOUR ENVIRONMENTS. A STRAIGHT-PATH FLOW-MATCHING POLICY REACHES THE AUDITED THRESHOLD AT ONE STEP; DDIM NEEDS ROUGHLY FOUR.

Family	NFE budget
Flow-Matching (standalone)	1
MeanFlow (u-head)	4
Flow-Matching (v-head)	4
Diffusion Policy (DDIM)	4

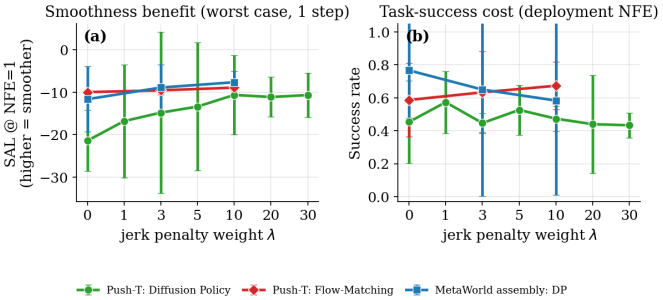

Fig. 6. Jerk-aware regularization ablation (3 seeds, 95% CI). Left: worst-case (NFE=1) spectral arc length rises (motion gets smoother) monotonically with the penalty weight λ, saturating near λ=10, across all three settings. Right: the success cost is task-dependent, negligible or positive on Push-T (deployment NFE=8), but a real tradeoff on long-horizon assembly (NFE=2).

and sweep its weight $\lambda \in \{0, 1, 3, 5, 10, 20, 30\}$ on Push-T (Diffusion Policy and flow-matching) and on MetaWorld assembly (3 seeds each). Figure 6 summarizes the outcome, and we report it as a knob rather than a new method.

The smoothing is real and agrees across all three of our independent metrics. On Push-T DDIM at one step, raising λ drives mean-squared jerk down by up to $20\times$ ($7.2\times10^8 \to 3.5\times10^7$ across the sweep), log-dimensionless jerk from -18.6 to -16.0, and SAL from -21.4 to -10.7; the SAL and LDLJ gains are already near their plateau by λ=10. On the same one-step Push-T rollout, the path MSJ drops from 6075 to 1951 after the penalty, consistent with removing high-frequency motion.

The penalty is useful, but its success cost is task-dependent (Fig. 6, right). On Push-T, multi-step success is unchanged or slightly improved (DDIM $0.45 \to 0.57$ at λ=1; flow-matching $0.59 \to 0.67$ at λ=10). On long-horizon assembly the same penalty trades success away ($0.77 \to 0.58$ at λ=10). And the smoothing is largest at NFE=1, where DDIM does not succeed anyway, so the regularizer helps most in the regime where it matters least for that family. Taken literally, a jerk penalty reliably removes the high-frequency component of few-step motion and is worthwhile for families that already succeed at low NFE, but it does not convert a too-few-steps policy into a working one.

E. Does roughness predict failure? Not reliably

A tempting corollary is that chunk-jerk could double as a free, label-free runtime signal: flag a rough chunk as a

likely failure and spend an extra step on it. We tested this directly, logging per-episode the jerk of the generated chunk against the episode outcome and measuring the AUC of jerk predicting failure (3 seeds). It does not hold up. For MeanFlow the signal is seed-unstable (AUC 0.45 to 1.00); for DDIM it is consistently *inverted* (AUC 0.21 to 0.36): its failures are *underactuated*, low-jerk, under-travelled motion that never engages the task, rather than jittery. Few-step roughness and few-step failure are distinct phenomena. That is exactly the point of the audit: smoothness carries information that success does not, and cannot be folded into it.

V. Discussion and Limitations

What a success-only protocol hides. Picking a one-step policy on success alone, one would reject DDIM and might accept MeanFlow at NFE=2, where coverage looks acceptable but the motion is still markedly rougher than the saturated regime. The smoothness axis makes that visible.

Limitations. Push-T coverage is a lenient max-over-episode score, so we interpret it together with success rate and emphasize relative trends rather than absolute coverage values. Our MeanFlow variant is a controlled, equal-budget implementation checked against the published objective, but not a fully optimized reproduction of dedicated MeanFlow-policy systems; the result should therefore be read as an equal-budget audit of this implementation, not as a general ranking of MeanFlow. The strong standalone flow-matching result should be tested at larger scale, on more tasks, and on hardware. Finally, all experiments are simulated.

Reporting recommendation. For few-step generative policies, report the task metric together with a smoothness-versus-NFE curve. We also recommend a per-method NFE budget: the smallest audited step count at which both task score and smoothness are within a preset tolerance of that method's high-NFE plateau. This keeps the comparison tied to the deployment question: how many network evaluations are enough for both success and usable motion?

VI. Conclusion

Cutting inference steps is not free, and the cost is method-dependent. How much task performance and motion smoothness a policy gives up depends on its generative formulation, and smoothness exposes that structure most clearly at the low step counts these methods are designed for, where success rate alone does not. The audit runs on one GPU, and we release it so that smoothness can be reported alongside success rate.

AI-use disclosure

The author used AI-assisted tools for language editing, figure-layout iteration, and editorial consistency checks. All scientific claims, experiments, figures, references, and final text were reviewed and approved by the author, who takes responsibility for the content.

References

[1] Sivakumar Balasubramanian, Alejandro Melendez-Calderon, Agnes Roby-Brami, and Etienne Burdet. On the analysis of movement smoothness. *Journal of NeuroEngineering and Rehabilitation*, 12(112), 2015.

[2] Kevin Black, Manuel Y. Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *arXiv preprint arXiv:2506.07339*, 2025. URL https://arxiv.org/abs/2506.07339.

[3] Zidong Chen, Zihao Guo, Peng Wang, ThankGod Itua Egbe, Yan Lyu, and Chenghao Qian. Dense-jump flow matching with non-uniform time scheduling for robotic policies: Mitigating multi-step inference degradation. *arXiv preprint arXiv:2509.13574*, 2025. URL https://arxiv.org/abs/2509.13574.

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

[6] 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. URL https://arxiv.org/abs/2505.13447.

[7] Zhengyang Geng et al. Improved mean flows: On the challenges of fast-forward generative models. *arXiv preprint arXiv:2512.02012*, 2025. URL https://arxiv.org/abs/2512.02012.

[8] Sinjae Kang, Chanyoung Kim, Kaixin Wang, Li Zhao, and Kimin Lee. WarmPrior: Straightening flow-matching policies with temporal priors. *arXiv preprint arXiv:2605.13959*, 2026. URL https://arxiv.org/abs/2605.13959.

[9] Soham Kulkarni, Raayan Dhar, and Yuchen Cui. Learning from the best: Smoothness-driven metrics for data quality in imitation learning. *arXiv preprint arXiv:2604.23000*, 2026. URL https://arxiv.org/abs/2604.23000.

[10] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

[11] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023.

[12] Guanxing Lu et al. ManiCM: Real-time 3d diffusion policy via consistency model for robotic manipulation. *arXiv preprint arXiv:2406.01586*, 2024. URL https://arxiv.org/abs/2406.01586.

[13] Siddharth Mysore, Bassel Mabsout, Renato Mancuso, and Kate Saenko. Regularizing action policies for smooth

control with reinforcement learning. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[14] 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.

[15] Allen Z. Ren, Justin Lidard, Lars L. Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024. URL https://arxiv.org/abs/2409.00588.

[16] Juyi Sheng, Ziyi Wang, Peiming Li, and Mengyuan Liu. MP1: Meanflow tames policy learning in 1-step for robotic manipulation. *arXiv preprint arXiv:2507.10543*, 2025. URL https://arxiv.org/abs/2507.10543.

[17] Xirui Shi and Jun Jin. FRMD: Fast robot motion diffusion with consistency-distilled movement primitives for smooth action generation. *arXiv preprint arXiv:2503.02048*, 2025. URL https://arxiv.org/abs/2503.02048.

[18] David Snyder, Apurva Badithela, Nikolai Matni, George Pappas, Anirudha Majumdar, Masha Itkina, and Haruki Nishimura. Beyond binary success: Sample-efficient and statistically rigorous robot policy comparison. *arXiv preprint arXiv:2603.13616*, 2026. URL https://arxiv.org/abs/2603.13616.

[19] Dongwoo Son and Suhan Park. LiPo: A lightweight post-optimization framework for smoothing action chunks generated by learned policies. *arXiv preprint arXiv:2506.05165*, 2025. URL https://arxiv.org/abs/2506.05165.

[20] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

[21] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *International Conference on Machine Learning (ICML)*, 2023.

[22] Yi Ru Wang, Carter Ung, Jiafei Duan, Ranjay Krishna, Dieter Fox, Siddhartha Srinivasa, et al. RoboEval: Where robotic manipulation meets structured and scalable evaluation. *arXiv preprint arXiv:2507.00435*, 2025. URL https://arxiv.org/abs/2507.00435.

[23] Zhendong Wang et al. One-step diffusion policy: Fast visuomotor policies via diffusion distillation. *arXiv preprint arXiv:2410.21257*, 2024. URL https://arxiv.org/abs/2410.21257.

[24] Songwei Wu, Zhiduo Jiang, Wandong Sun, Guanghu Xie, Rui Zhao, Hong Liu, and Yang Liu. CoLA-Flow policy: Temporally coherent imitation learning via continuous latent action flow matching for robotic manipulation. *arXiv preprint arXiv:2601.23087*, 2026. URL https://arxiv.org/abs/2601.23087.

[25] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.