SwiftFlow: An Efficient One-Step Policy Learning via Improved Mean Flow for Robotic Manipulation

Luyao Liu[1,2] **Wenlong Dong**[1,2] **Chengjie Zhang**[1,2] **Hong Zhang**[1,2,*]
[1]Southern University of Science and Technology
[2]Shenzhen Key Laboratory of Robotics and Computer Vision
{liuly2023, dongwl2023, zhangcj2023}@mail.sustech.edu.cn,
hzhang@sustech.edu.cn

Abstract

Generative modeling has become a mainstream paradigm for learning based robot manipulation research. Among these, diffusion-based policies achieve strong performance but typically require multiple iterative denoising steps to generate final action, leading to inefficient inference. Although flow-based methods eliminate the need for iterative denoising process, existing approaches often face challenges in maintaining stable optimization and high-quality action generation. In this paper, we propose SwiftFlow, an efficient one-step policy learning framework built upon improved mean flow for robotic manipulation. Specifically, SwiftFlow formulates policy learning as a conditional generation problem by jointly leveraging 3D point cloud observations and proprioceptive robot states. It stably learns mean velocity field through a principled regression objective and optimization process, enabling more reliable action generation while preserving efficient inference. To further regularize the learned implicit generation dynamics, we construct a mean velocity smoothness regularization term, which constrains rapid local variations in the learned mean velocity field. Since this regularization is imposed only during training, it hardly introduces additional inference overhead. Extensive experiments on Adroit and MetaWorld benchmarks demonstrate that our proposed method achieves stronger task performance together with highly efficient one-step inference, validating its effectiveness for robotic manipulation tasks.

1 Introduction

With the growing demand for autonomous robots that can perform complex physical tasks, robot manipulation [1, 2, 3] has become a fundamental problem in robot learning. Much of the recent progress in this area is driven by generative policy models, which can represent action distributions and produce temporally coherent action sequences from visual observations. In particular, Diffusion Policy [4] formulates action generation as a conditional denoising process and demonstrates significant improvements in visuomotor policy learning. Subsequently, DP3 [5] shows that 3D point cloud observations provide a compact and effective geometric representation for robot manipulation. In parallel, 3D Diffuser Actor [6] introduces a diffusion policy built on 3D denoising transformer, demonstrating that tokenized 3D scene representations and relative 3D attention can substantially benefit visuomotor policy learning. Despite their promising progress, these methods typically require iterative denoising steps to transform noise into executable action trajectories, resulting in increased inference latency.

Flow-based generative modeling [7, 8, 9] provides a promising direction for improving the efficiency of generative manipulation policies. Rather than reversing a stochastic diffusion process through iterative denoising, flow-based methods learn a continuous velocity field that

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

deterministically transports samples from a simple prior distribution toward the target action distribution. This formulation offers a natural alternative to diffusion-based generation and has recently inspired growing interest in efficient few-steps and one-step policy learning. For example, PointFlowMatch [10] investigates conditional flow matching for robotic policy learning and shows that it is particularly effective, highlighting the strong potential of flow-based generative modeling for manipulation task. What's more, FlowPolicy [11] adopts consistency flow matching to derive a 3D point cloud conditioned policy in a single inference step, improving efficiency while maintaining competitive manipulation performance. Building on this, MP1 [12] further extends MeanFlow [13] to robotic manipulation and learns mean velocity fields to realize a one-step action generation strategy. However, the original MeanFlow formulation still has notable limitations for robotic manipulation. Its training target depends not only on the underlying target field but also on the network prediction itself, making the learning objective less standard and the optimization less stable. As a result, the learned mean velocity field may exhibit undesirable local fluctuations, which can degrade action quality and lead to execution errors.

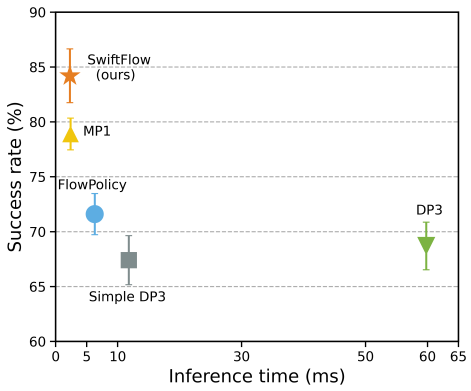


Figure 1: Comparison of different methods in terms of inference time and success rate on the Adroit and MetaWorld tasks. The plot shows that SwiftFlow exhibits strong overall results across both metrics.

Motivated by the above challenges, this paper proposes SwiftFlow, a point cloud conditioned one-step policy learning framework for robot manipulation based on improved mean flow [14]. In particular, SwiftFlow is a visuomotor policy that unifies 3D geometric perception and proprioceptive conditioning within an efficient one-step generative framework. Instead of explicitly modeling an instantaneous velocity field, our method parameterizes policy generation with interval-averaged velocity, thereby enabling one-step action trajectory generation. More importantly, improved mean flow provides a more principled training objective for fast generative modeling, leading to a more stable optimization process and a more reliable one-step mapping from latent noise to robot action. However, efficient one-step generation alone does not guarantee desirable transport dynamics. The learned mean velocity field may still exhibit rapid temporal variations across neighboring states, which can induce locally high-curvature transport trajectories and adversely affect action trajectory accuracy and control stability. To mitigate this issue, we further introduce a mean velocity smoothness regularization term which is used to constrain the abrupt variation of the predicted mean velocity field. Since this regularization is applied exclusively during training, it guides policy optimization without introducing any additional computational overhead at inference time. Consequently, the learned policy can still generate actions through a single forward pass, thereby preserving the low latency and computational efficiency of one-step policy generation. We validate the proposed method on challenging robot manipulation tasks and show that SwiftFlow yields strong empirical performance. Experimental results demonstrate that our method achieves effective policy learning while maintaining highly efficient one-step inference.

The contributions of this paper are as follows:

- We propose SwiftFlow, a point cloud conditioned one-step policy learning framework for robot manipulation based on improved mean flow. By leveraging a more standard regression objective for average velocity prediction, SwiftFlow enables efficient and reliable action generation.
- We construct a mean velocity smoothness regularization that suppresses abrupt local variations in the learned mean velocity field along conditional generation trajectories. This regularization term improves the stability and coherence of one-step policy learning.
- Extensive experiments on 37 simulation tasks from Adroit and MetaWorld benchmark show that SwiftFlow achieves a strong balance between manipulation performance and inference efficiency, outperforming previous state-of-the-art methods by 5.3% in success rate. We further demonstrate its effectiveness in real-world experiments, where it also delivers stronger manipulation performance than prior approaches.

2 Related Work

2.1 Visuomotor Policy Learning for Robot Manipulation

Early visuomotor policy learning methods [15, 16, 17] for robot manipulation primarily rely on 2D image observations as visual inputs, where end-to-end policies are learned from RGB images and proprioceptive states to predict low-level control actions. Representative works such as BC-Z [18], RT-1 [19], ACT [20], and DP [4] demonstrate a common trend of improving visuomotor manipulation by learning action policies directly from image observations through scalable imitation learning and expressive action modeling. However, 2D observations remain indirect for manipulation, since they do not explicitly encode 3D geometry and spatial relations. To enhance spatial reasoning, recent studies have explored point cloud based manipulation policies, which provide a more direct representation of object shape, scene layout, and workspace geometry. Early methods such as C2F-ARM [21] and PerAct [22] discretize the workspace into voxels, while later approaches including Act3D [23] and RVT [24] improve the efficiency-accuracy trade-off through 3D action queries or multi-view back-projection. More recent generative policies, such as DP3 [5] and PointFlowMatch [10], further demonstrate the effectiveness of point cloud conditioning for geometry aware action prediction.

2.2 Diffusion-based Manipulation Policy

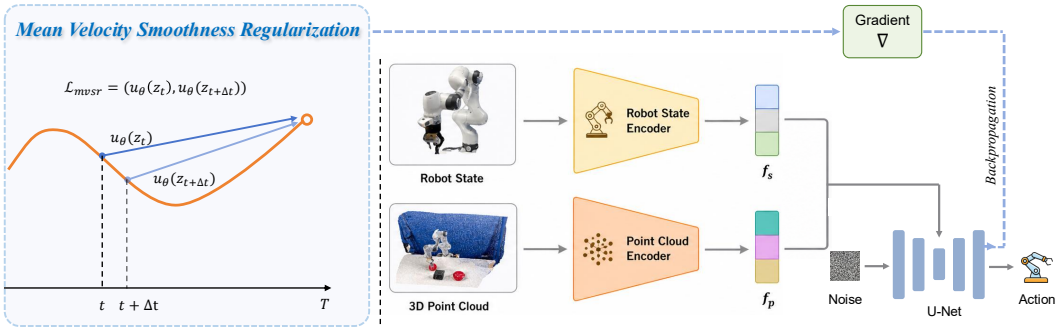
Motivated by the success of generative models [25, 26, 27] in image generation field, diffusion-based policies have emerged as a dominant paradigm for robot manipulation. Initially, DP [4] and DP3 [5] formulate action generation as a conditional denoising process and indicate that iterative diffusion can effectively model multimodal visuomotor behaviors. Several subsequent studies further extend this family in different directions. EquiDiff [28] introduces equivariant structure into diffusion policy learning to improve sample efficiency and generalization, while CAGE [29] improves data efficiency and robustness under visual distribution shifts through causal attention and a diffusion-based action prediction head. In addition, RDT [30] adopts a diffusion transformer for bimanual manipulation. More recently, FP3 [31] further pushes this direction by building a large-scale 3D diffusion transformer policy pretrained on point-cloud observations. Despite their strong performance, these methods still typically rely on iterative denoising or multi-step sampling during inference, which introduces nontrivial latency for closed-loop manipulation. This limitation motivates the development of more efficient policy learning paradigms that can generate high-quality actions in a single step without sacrificing expressiveness.

2.3 Flow-based Manipulation Policy

Flow-based generative modeling has recently emerged as an appealing alternative to diffusion-based policy learning for robot manipulation. Unlike diffusion models that rely on stochastic reverse denoising, Flow Matching [7] learns a continuous vector field, transporting samples from a simple noise distribution to the target action distribution through deterministic dynamics. This formulation balances expressive generative modeling with efficient sampling and has motivated several flow-based visuomotor policies. For example, PointFlowMatch [10] extends conditional flow matching to point cloud conditioned manipulation, demonstrating effective action generation from explicit 3D scene observations. Recent methods [32, 33, 11] further improve inference efficiency in flow-based manipulation. Consistency Flow Matching [34] introduces a self-consistency constraint on the velocity field, encouraging trajectories starting from different intermediate times to share a common endpoint. Building on this idea, FlowPolicy [11] adapts consistency flow matching to 3D manipulation and enables point cloud conditioned action decoding in a single forward step. More recently, MP1 [12] incorporates MeanFlow [13] into robotic manipulation by learning interval-averaged rather than instantaneous velocity fields, avoiding explicit ODE integration and supporting one-step action generation from point-cloud observations. These studies indicate that flow-based policy learning is a promising direction for efficient manipulation, particularly under low latency inference requirements.

3 Method

We consider robot manipulation policy learning from demonstrations, where the goal is to predict an executable future action trajectory conditioned on the current observation. Rather than modeling



(a) Training with Regularization Objective *(b) One-step Robot Policy*

Figure 2: Overview of SwiftFlow. a) During training, we introduce a mean velocity smoothness regularization objective to constrain the predicted velocity field at neighboring time steps. b) SwiftFlow takes the robot state and 3D point cloud observation as input, extracts the state feature $\mathbf{f}_s$ and point-cloud feature $\mathbf{f}_p$ through separate encoders, and uses their fused representation to condition a U-Net-based generator that transforms noise into the target action in one step.

action generation as an iterative denoising process, we aim to generate the target action trajectory in a single forward pass, thereby reducing inference latency for real-time manipulation. To this end, we propose SwiftFlow, an efficient one-step policy learning framework for robotic manipulation built upon improved mean flow. As illustrated in Fig. 2, SwiftFlow takes the current observation of the environment as input, primarily represented by a 3D point cloud along with the robot's proprioceptive state, and directly predicts a chunk of future actions. Compared with original mean flow formulations, improved mean flow yields a better-conditioned optimization objective, leading to more stable learning of the mean velocity field and more reliable action generation. However, the learned mean velocity field may still vary sharply across neighboring temporal states, which can destabilize the implicit transport process. To address this issue, we propose mean velocity smoothness regularization to constrain the local oscillations of the learned mean velocity field, thereby stabilizing the implicit generation dynamics.

In the following, we detail the proposed method in three parts. First, Sec. 3.1 introduces the one-step policy learning framework based on improved Mean Flow. Then, we present the mean velocity smoothness regularization for stabilizing the learned mean velocity field in Sec. 3.2. Finally, we describe the overall training objective and the corresponding inference process in Sec. 3.3.

3.1 One-Step Policy Learning with Improved Mean Flow

Given the current observation, consisting of the scene point cloud $\mathbf{P}$ and the robot state $\mathbf{S}$, our goal is to predict a future action chunk $A \in \mathbb{R}^{H \times d_a}$ for manipulation, where d_a is the dimension of action. Then, the scene point cloud $\mathbf{P}$ and the robot state $\mathbf{S}$ are encoded by two separate encoders into latent feature space. Specifically, the point cloud encoder consists of a sequence of Linear layers interleaved with LayerNorm and ReLU activations. It transforms each input point into a latent point-wise representation, and the resulting features are aggregated via a max-pooling operation followed by a projection layer to obtain the geometric feature $\mathbf{f_p}$. In parallel, the robot state is embedded by MLP into the proprioceptive feature $\mathbf{f_s}$. These two modality-specific features are then fused into a condition representation $\mathbf{c} = [\mathbf{f_p}, \mathbf{f_s}]$, enabling the policy to jointly capture scene geometry and proprioceptive embodiment information for subsequent action generation. The central design of SwiftFlow is to cast policy learning as a one-step generative mapping from a noisy distribution to the clean action chunk. To this end, we adopt improved Mean Flow to build our policy rather than diffusion model or conventional flow matching paradigm.

Specifically, Improved Mean Flow formulates one-step generation through an average velocity field defined over a finite time interval, allowing the model to parameterize fast-forward generation from noisy states to target samples. Its training objective is constructed by rewriting the MeanFlow identity as a loss function on the instantaneous velocity, while retaining an average velocity predictor model as the underlying network parameterization. In this way, it casts learning into a standard regression problem defined on the noisy state, and preserves efficient one-step sampling through a single forward pass at inference time.

Following the MeanFlow formulation, we introduce the mean velocity field between two time steps r and t,

$$u(z_t, r, t) \triangleq \frac{1}{t-r} \int_r^t v(z_\tau, \tau)\, d\tau, \tag{1}$$

where $v(\cdot)$ denotes the underlying instantaneous velocity field. Rather than learning v directly, our method learns a neural predictor $u_\theta(z_t, r, t)$, which estimates the mean velocity field parameterizing the mapping from noisy action states to clean actions.

However, in the original MeanFlow, the training target depends on network prediction itself through the Jacobian-vector-product term, which makes the objective less standard from a regression perspective. Improved Mean Flow resolves this issue by reformulating training as a loss on the instantaneous velocity, re-parameterized by the average velocity predictor, so that the prediction function depends only on the legitimate noisy state. This leads to a better-posed optimization problem and more stable training while preserving the one-step sampling structure. Concretely, the improved Mean Flow identity can be rewritten as

$$v(z_t) = u(z_t) + (t-r)\frac{d}{dt}u(z_t). \tag{2}$$

This identity shows that the instantaneous velocity can be recovered from the average velocity together with its temporal variation. Therefore, although the model is parameterized by the average velocity field, it can still be trained through an equivalent instantaneous-velocity prediction objective. Let $\epsilon \sim \mathcal{N}(0, I)$ denote a Gaussian noise sample. By substituting the neural predictor u_θ, we obtain the corresponding composite predictor

$$V_\theta(z_t) = u_\theta(z_t) + (t-r)\,\mathrm{JVP}_{\mathrm{sg}}(u_\theta; \epsilon - A), \tag{3}$$

where $\mathrm{JVP}_{\mathrm{sg}}$ denotes the Jacobian-vector product of u_θ along the direction $\epsilon - A$, and the subscript sg indicates that stop-gradient is applied to the resulting derivative term. Under this form, the derivative contribution is absorbed into a computable correction defined on top of the average velocity predictor. Importantly, this reparameterization removes the self-referential dependence of the supervision target on the network prediction itself, yielding a prediction function that is evaluated on the legitimate noisy input z_t. As a result, training is cast as a better-conditioned regression problem while preserving the average velocity prediction mechanism implemented by u_θ.

For our SwiftFlow, the corresponding training objective is formulated as

$$\mathcal{L}_{\mathrm{iMF}} = \mathbb{E}\Big[\|V_\theta(z_t) - (\epsilon - A)\|^2\Big]. \tag{4}$$

This objective supervises the network in the instantaneous-velocity space using the standard target $(\epsilon - A)$, while the underlying model u_θ remains parameterized by the mean velocity field. Consequently, SwiftFlow inherits the stable optimization property of improved Mean Flow and preserves the single evaluation decoding pathway needed for efficient one-step policy inference.

3.2 Mean Velocity Smoothness Regularization

While improved Mean Flow provides an efficient and principled objective for one-step action generation, the primary learning objective alone does not explicitly constrain the rapid local variation of the learned mean velocity field across neighboring states along the conditional generation trajectory. In robot manipulation, however, action generation is highly sensitive to small geometric changes in the scene and to local perturbations in the latent action state. Under such conditions, a mean velocity field that varies too abruptly across nearby temporal states may induce unstable implicit generation dynamics, making the one-step prediction model brittle and reducing the reliability of the generated action chunk. To mitigate this issue, we introduce a mean velocity smoothness regularization term that explicitly constrains local difference of the learned mean velocity field during training process.

Importantly, this regularization is imposed directly on the mean velocity field. Our goal is to stabilize the underlying field that parameterizes one-step generation. Since SwiftFlow reconstructs actions through the predicted mean velocity field, enforcing smoother local variation in this field leads to more stable action generation dynamics while preserving the one-step structure of the model.

Concretely, we measure the discrepancy between mean velocity predictions evaluated at two neighboring temporal states under the same condition. We therefore penalize excessive discrepancy between

Table 1: **Performance of different methods on 37 tasks.** We evaluate the performance of our method on 3 Adroit and 34 MetaWorld tasks with three random seeds, comparing it to SOTA methods.

Methods	NFE	Adroit			MetaWorld				Average
		Hammer	Door	Pen	Easy (21)	Medium (4)	Hard (4)	Very Hard (5)	
DP [4]	10	16 ± 10	34 ± 11	13 ± 2	50.7 ± 6.1	11.0 ± 2.5	5.25 ± 2.5	22.0 ± 5.0	35.2 ± 5.3
Adaflow [35]	–	45 ± 11	27 ± 6	18 ± 6	49.4 ± 6.8	12.0 ± 5.0	5.75 ± 4.0	24.0 ± 4.8	35.6 ± 6.1
CP [36]	1	45 ± 4	31 ± 10	13 ± 6	69.3 ± 4.2	21.2 ± 6.0	17.5 ± 3.9	30.0 ± 4.9	50.1 ± 4.7
DP3 [5]	10	100 ± 0	56 ± 5	46 ± 10	87.3 ± 2.2	44.5 ± 8.7	32.7 ± 7.7	39.4 ± 9.0	68.7 ± 4.7
Simple DP3 [5]	10	98 ± 2	40 ± 17	36 ± 4	86.8 ± 2.3	42.0 ± 6.5	38.7 ± 7.5	35.0 ± 11.6	67.4 ± 5.0
FlowPolicy [11]	1	98 ± 1	61 ± 2	54 ± 4	84.8 ± 2.2	58.2 ± 7.9	40.2 ± 4.5	52.2 ± 5.0	71.6 ± 3.5
MP1 [12]	1	100 ± 0	69 ± 2	58 ± 5	88.2 ± 1.1	68.0 ± 3.1	58.1 ± 5.0	67.2 ± 2.7	78.9 ± 2.1
SwiftFlow	1	90 ± 3	75 ± 6	56 ± 7	91.9 ± 3.3	74.5 ± 11.7	64.7 ± 13.6	82.0 ± 2.2	84.2 ± 6.0

Table 2: **Inference time of different methods on 37 tasks.** We evaluate the inference time of our method on 3 Adroit and 34 MetaWorld tasks with three random seeds, comparing it to previous methods.

Methods	NFE	Adroit /ms			MetaWorld /ms				Average
		Hammer	Door	Pen	Easy (21)	Medium (4)	Hard (4)	Very Hard (5)	
DP3 [5]	10	62.1 ± 1.8	61.3 ± 0.6	61.0 ± 0.1	59.4 ± 0.8	59.4 ± 1.1	59.1 ± 0.1	59.2 ± 0.3	59.5 ± 1.0
Simple DP3 [5]	10	12.2 ± 0.6	11.8 ± 0.2	11.7 ± 0.3	11.7 ± 0.2	11.9 ± 0.2	11.8 ± 0.2	11.9 ± 0.2	11.8 ± 0.2
FlowPolicy [11]	1	6.1 ± 0.0	6.2 ± 0.0	6.2 ± 0.1	6.3 ± 0.1	6.3 ± 0.1	6.2 ± 0.1	6.3 ± 0.1	6.3 ± 0.1
MP1 [12]	1	2.3 ± 0.1	2.4 ± 0.0	2.4 ± 0.0	2.4 ± 0.1	2.4 ± 0.1	2.5 ± 0.1	2.4 ± 0.1	2.4 ± 0.1
SwiftFlow	1	2.4 ± 0.1	2.3 ± 0.1	2.4 ± 0.0	2.3 ± 0.1	2.4 ± 0.0	2.3 ± 0.0	2.3 ± 0.1	2.3 ± 0.0

these two predictions and define the mean velocity smoothness regularization as

$$\mathcal{L}_{\text{mvsr}} = \mathbb{E}\left[\left\|u_\theta(z_t) - u_\theta(z_{t+\Delta t})\right\|^2\right], \tag{5}$$

where z_t and $z_{t+\Delta t}$ are two neighboring samples drawn from the same condition trajectory at time step t and $t + \Delta t$, respectively. It suppresses rapid local changes of the mean velocity field along the conditional generation trajectory. In other words, if two nearby latent states correspond to nearly the same generation context, the model is encouraged to produce compatible mean velocity predictions rather than sharply changing its estimated average displacement. This local consistency helps stabilize the implicit one-step generation dynamics induced by improved Mean Flow and reduces undesirable oscillations in the learned field.

3.3 Training Objective and Inference

The final training objective of SwiftFlow combines the improved Mean Flow objective with the proposed mean velocity smoothness regularization. Formally, the overall loss is defined as

$$\mathcal{L} = \mathcal{L}_{\text{iMF}} + \omega \mathcal{L}_{\text{mvsr}}, \tag{6}$$

where ω balances the contribution of the regularization term. Practically, we employ the time-dependent weight $\omega(t) = (1-t)^2$ so that earlier timesteps yield smoother flow, whereas timesteps near $t = 1$ focus more on accuracy. The first term is responsible for learning an accurate one-step mapping from noisy action states to clean actions through mean velocity field prediction, while the second term constrains excessive local variation in the learned mean velocity field. Together, these two objectives improve both the fidelity and the stability of one-step policy learning.

Importantly, the proposed regularization term is introduced exclusively during training and is disabled at inference stage. Consequently, during inference, SwiftFlow strictly follows the original one-step generation rule of improved Mean Flow, thereby preserving its computational efficiency and incurring no additional inference overhead. Specifically, given the corresponding conditional features and a Gaussian noise sample $\epsilon \sim \mathcal{N}(0, I)$, SwiftFlow generates the entire action chunk $\hat{A}$ in a single forward pass:

$$\hat{A} = \epsilon - u_\theta(\epsilon, 0, 1). \tag{7}$$

where $r = 0, t = 1$. This preserves the one-step generation structure of improved Mean Flow and enables efficient 1-NFE inference. In particular, the mean velocity smoothness regularization is utilized only during training stage to shape the learned field and is completely removed at test time. As a result, SwiftFlow preserves the efficiency advantage of one-step generation while benefiting from a more stable mean velocity field learned during optimization.

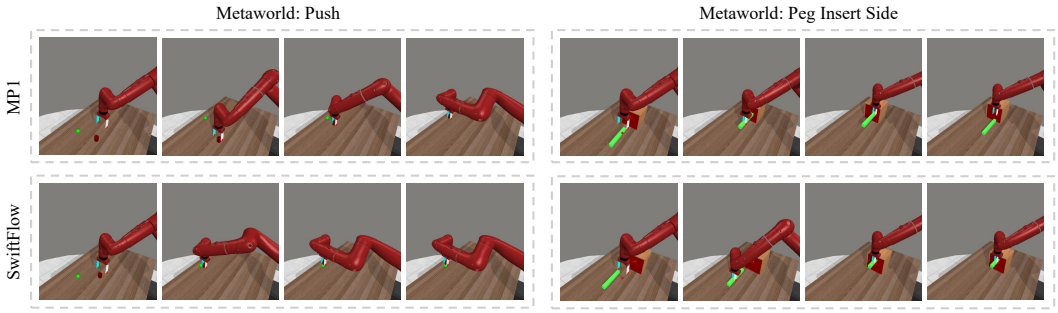


Figure 3: Qualitative comparison of the proposed SwiftFlow and the previous SOTA method MP1 on two tasks from MetaWorld. SwiftFlow successfully generates actions to complete these tasks effectively. But MP1 fails to finish the "peg insert side" task.

4 Experiment

4.1 Experiment Setup

Simulation Benchmarks. We evaluate SwiftFlow on two widely used robot manipulation benchmarks consisting of Adroit [37] and MetaWorld [38]. These two environments provide complementary testbeds for assessing policy learning in manipulation. Adroit emphasizes dexterous control with a high-degree-of-freedom anthropomorphic hand, requiring precise coordination in a challenging continuous action space. In contrast, MetaWorld consists of a diverse suite of goal-conditioned manipulation tasks and is widely used to assess cross-task generalization. Evaluating on both benchmarks allows us to examine the proposed method under both high-dimensional dexterous manipulation and broad multi-task manipulation settings.

Baselines. We compare SwiftFlow against a representative set of recent generative manipulation policies, including DP[4], CP[36], AdaFlow[35], DP3[5], MP1[12], and FlowPolicy[11]. These baselines cover both iterative diffusion-based methods and more efficient flow matching formulations.

Implementation Details. In our implementation, the policy input consists of a 3D point cloud observation together with the robot proprioceptive state. Point clouds are downsampled using farthest point sampling before being fed into the policy network, while image-based baselines use resized RGB observations under the same benchmark setting. Meanwhile, we utilize an observation horizon of 2 steps as policy input, maintain the most recent 4 states as temporal context, and predict a 4 step action sequence at each decision step. All methods are trained using the AdamW optimizer with a learning rate of 1×10^{-4}. The batch size is set to 128. Besides, the model is trained for 3000 epochs on Adroit and 1000 epochs on MetaWorld. Our model and all compared methods are developed based on the PyTorch framework and performed on a single NVIDIA RTX 3090 GPU.

Evaluation Metrics. We report success rate as the primary metric for manipulation performance. To reduce the effect of training stochasticity, each method is trained with three random seeds, and the final results are reported as the mean and standard deviation across seeds. Following prior work, we evaluate 20 episodes per 200 training epochs and record the mean of the five highest success rates. Finally, the overall success rate and standard deviation for the task are computed across all three seeds. To evaluate inference efficiency, we additionally report the number of function evaluations (NFE) and the inference time. Since SwiftFlow is designed for one-step policy generation, these efficiency metrics are critical for quantifying its practical advantage in real-time manipulation.

4.2 Simulation Results

In Table 1, we show the average success rate of all methods on 37 simulation tasks from Adroit and MetaWorld. Overall, SwiftFlow achieves the best performance among all compared methods, reaching an average success rate of 84.2%. Compared with MP1, SwiftFlow improves the overall success rate by 5.3%, while yielding larger gains over FlowPolicy and DP3. More importantly, the improvement is not limited to easier tasks, but remains consistent across more challenging settings. In particular, SwiftFlow achieves 74.5%, 64.7%, and 82.0% on the Medium, Hard, and Very Hard

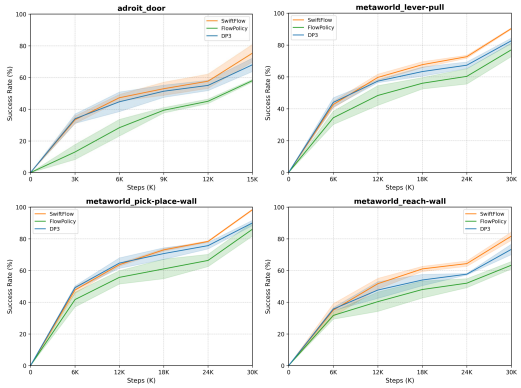


Figure 4: Success rate curves of different methods on multiple MetaWorld tasks. We compare the performance of SwiftFlow, FlowPolicy, and DP3 on four tasks.

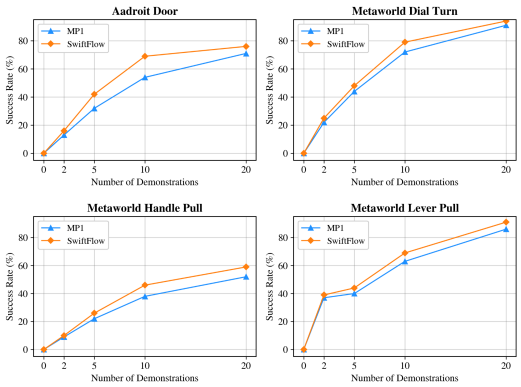


Figure 5: Ablation study on the number of expert demonstrations. We select four tasks to explore the influence of different numbers of demonstrations on SwiftFlow and MP1.

subsets of MetaWorld, respectively, all of which are clearly higher than those of MP1 and FlowPolicy. These results show that SwiftFlow is able to produce more reliable action predictions in difficult manipulation scenarios. We attribute these gains to two key properties of SwiftFlow. First, by adopting the improved mean flow formulation, SwiftFlow learns a more standard one-step generative objective, which leads to more stable optimization and more accurate action generation than existing one-step baselines. Second, the proposed mean velocity smoothness regularization explicitly constrains rapid local variations in the learned mean velocity field, encouraging smoother and more coherent trajectory generation. Moreover, SwiftFlow achieves the lowest average inference time of 2.3 ms with NFE = 1, as demonstrated in Table 2. Compared with baselines, SwiftFlow is substantially faster than DP3 and Simple DP3, and also clearly outperforms FlowPolicy, demonstrating the efficiency advantage of one-step policy generation. Although the gain over MP1 is modest, this comparison result indicates that the improvements introduced by SwiftFlow do not sacrifice inference efficiency. This favorable efficiency stems from the fact that SwiftFlow directly predicts the target action sequence in a single step, without multi-step denoising or iterative numerical integration, and the improved mean flow formulation retains a simple inference procedure without introducing additional sampling overhead. As a result, SwiftFlow not only preserves the efficiency advantage of one-step policies, but also improves the quality of the generated actions.

We present the training success rate trajectories of SwiftFlow, FlowPolicy, and DP3 on four tasks, as demonstrated in Fig. 4. Across all four tasks, the success rates of all methods increase as training proceeds, but SwiftFlow consistently follows a more favorable trend. In particular, it rises more rapidly during the early stages of training and ultimately reaches the highest final performance among the compared methods. We also provide a qualitative comparison between SwiftFlow and MP1, as shown in Fig. 3. While both methods complete the *Push* task successfully, SwiftFlow also solves the more challenging *Peg Insert Side* task with a coherent insertion trajectory. MP1 fails to achieve the required alignment, highlighting the stronger precision and task robustness of our proposed policy. These results indicate that SwiftFlow not only learns effective manipulation policies more efficiently, but also exhibits stronger training stability and better final policy quality. than existing flow-based and diffusion-based baselines.

4.3 Ablation Study

As shown in Table 3, we demonstrate ablation results for the mean velocity smoothness regularization on serval MetaWorld tasks. Removing $\mathcal{L}_{\text{mvsr}}$ leads to a consistent performance drop on all evaluated tasks, while the full SwiftFlow model achieves a higher success rate. These results verify that $\mathcal{L}_{\text{mvsr}}$ makes a meaningful contribution to policy learning. The improvement is particularly pronounced on tasks such as Disassemble, Stick Pull, and Handle Pull, where successful execution requires temporally coherent motion and accurate local control. This observation suggests that constraining rapid local variations in the learned mean velocity field helps the policy generate more reliable action sequences, thereby reducing unstable predictions that may otherwise accumulate into task failure. Overall, the ablation study confirms that the proposed $\mathcal{L}_{\text{mvsr}}$ term is an important component of SwiftFlow, contributing to higher task performance.

We further study how the amount of training data affects policy learning by varying the number of expert demonstrations and comparing SwiftFlow with MP1. As shown in the Fig. 5, both methods exhibit improved success rates as the number of demonstrations increases, indicating that richer supervision provides more information for learning effective manipulation policies. Across the evaluated tasks, SwiftFlow shows a more favorable performance trend and achieves stronger results overall than MP1 over a broad range of demonstration settings. This observation suggests that the proposed method can more effectively translate increased demonstration data into policy improvement, leading to better final task performance across different training scales.

Table 3: Ablation study on mean velocity smoothness regularization. $w/o\mathcal{L}_{\text{mvsr}}$ indicates that the designed mean velocity smoothness regularization term is removed.

Method	Handle Pull	Coffee Pull	Hammer	Soccer	Stick Pull	Disassemble	Pick Place	Reach Wall
$w/o\ \mathcal{L}_{\text{mvsr}}$	52.7	96.3	98.0	14.7	71.7	65.0	80.3	80.7
SwiftFlow	55.3	97.7	98.7	16.3	73.7	71.0	81.0	82.1

4.4 Real-World Experiments

Our real-world experimental setup is illustrated in Fig. 6. The platform consists of a Franka robotic arm and an Intel RealSense L515 RGB-D camera for visual observation. Then, we evaluate our proposed method on four real-world manipulation tasks, namely *Hang Cup, Clean Pan, Pick Fruit, and Open Drawer*. For each task, we collect 12 demonstration trajectories for training, and use an action chunk size of 16 during both policy learning and execution process. Following prior real-world experimental settings, we run 10 trials for each method on every task to ensure a fair and consistent comparison. Under this unified setting, we conduct a comparison between SwiftFlow and MP1.

The quantitative comparison results are shown in Table 4. In general, SwiftFlow achieves better performance than MP1 across all evaluated real-world tasks, indicating that our proposed method transfers effectively from simulation to realistic manipulation scenarios. The advantage is particularly evident on tasks that require more reliable visuomotor coordination and execution precision, suggesting that SwiftFlow produces more robust action sequences under real-world perception noise. These results demonstrate that our proposed method not only improves policy quality in simulation, but also yields stronger practical performance on a physical robot, demonstrating better competence in real-world deployment. Additionally, demonstration videos of the real-world experiments will be provided in the supplementary material.



Figure 6: Real setup.

Table 4: Success rates (%) of different methods in real-world experiments.

Method	Hang Cup	Clean Pan	Pick Fruit	Open Drawer	Average
MP1 [12]	80%	70%	80%	60%	72.5%
SwiftFlow	90%	90%	100%	90%	92.5%

5 Conclusion

In this paper, we present SwiftFlow, an efficient one-step policy learning framework for robotic manipulation that integrates 3D point cloud perception with action generation policy based on improved mean flow. By directly learning a conditional mapping from a noise distribution to executable action chunks, SwiftFlow eliminates the costly iterative denoising process used in conventional generative policies and enables fast policy inference while preserving the capacity to model complex manipulation behaviors. To further improve the quality of one-step generation, we introduce a mean velocity smoothness regularization that stabilizes training by discouraging rapid local variations in the learned mean velocity field, leading to more coherent action trajectories without incurring any additional inference cost. Extensive experiments in both simulation and real-world environments demonstrate that SwiftFlow achieves superior performance to representative baselines across diverse manipulation tasks. Overall, these results suggest that one-step generative policy learning is a promising direction for scalable robot manipulation and indicate that SwiftFlow is a practical solution for efficient, stable, and deployable visuomotor policy.

Acknowledgments and Disclosure of Funding

This work was supported in part by the Shenzhen Science and Technology Program (No. SGDX20240115111759002), in part by the High-Level Special Funds (G03034K003) from Southern University of Science and Technology, in part by the Open Research Fund (GML-KF-24-15) from Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), and in part by the Shenzhen Association for Science and Technology (No. XHXS2025-003). The authors declare no competing interests.

References

[1] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *Journal of Machine Learning Research*, vol. 17, no. 39, pp. 1–40, 2016. [Online]. Available: http://jmlr.org/papers/v17/15-522.html

[2] P. Sharma, L. Mohan, L. Pinto, and A. Gupta, "Multiple interactions made easy (mime): Large scale demonstrations data for imitation," in *Conference on robot learning*. PMLR, 2018, pp. 906–915.

[3] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.

[4] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. C. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, Jul. 2023.

[5] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[6] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," in *8th Annual Conference on Robot Learning*, 2024.

[7] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *The Eleventh International Conference on Learning Representations*, 2023.

[8] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *The Eleventh International Conference on Learning Representations*, Kigali, Rwanda, 2023, iCLR 2023. [Online]. Available: https://openreview.net/forum?id=XVjTT1nw5z

[9] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, "Improving and generalizing flow-based generative models with minibatch optimal transport," *Transactions on Machine Learning Research*, 2024, also presented at Frontiers4LCD Workshop at ICML 2023. [Online]. Available: https://openreview.net/forum?id=CD9Snc73AW

[10] E. Chisari, N. Heppert, M. Argus, T. Welschehold, T. Brox, and A. Valada, "Learning robotic manipulation policies from point clouds with conditional flow matching," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, P. Agrawal, O. Kroemer, and W. Burgard, Eds., vol. 270. PMLR, 06–09 Nov 2025, pp. 982–993. [Online]. Available: https://proceedings.mlr.press/v270/chisari25a.html

[11] Q. Zhang, Z. Liu, H. Fan, G. Liu, B. Zeng, and S. Liu, "Flowpolicy: Enabling fast and robust 3d flow-based policy via consistency flow matching for robot manipulation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 14, pp. 14 754–14 762, 2025.

[12] J. Sheng, Z. Wang, P. Li, and M. Liu, "Mp1: Meanflow tames policy learning in 1-step for robotic manipulation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 40, no. 22, pp. 18 532–18 539, 2026.

[13] Z. Geng, M. Deng, X. Bai, J. Z. Kolter, and K. He, "Mean flows for one-step generative modeling," in *Advances in Neural Information Processing Systems*, 2025, neurIPS 2025 Oral.

[14] Z. Geng, Y. Lu, Z. Wu, E. Shechtman, J. Z. Kolter, and K. He, "Improved mean flows: On the challenges of fastforward generative models," *arXiv preprint arXiv:2512.02012*, 2025. [Online]. Available: https://arxiv.org/abs/2512.02012

[15] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 512–519.

[16] E. Johns, "Coarse-to-fine imitation learning: Robot manipulation from a single demonstration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 4613–4619.

[17] Y. Wang, G. Ajaykumar, and C.-M. Huang, "See what i see: Enabling user-centric robotic assistance using first-person demonstrations," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 639–648.

[18] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[19] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, and J. Hsu, "Rt-1: Robotics transformer for real-world control at scale," *Robotics: Science and Systems XIX*, 2023.

[20] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, Jul. 2023.

[21] S. James, K. Wada, T. Laidlow, and A. J. Davison, "Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 13 729–13 738.

[22] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, K. Liu, D. Kulic, and J. Ichnowski, Eds., vol. 205. PMLR, 14–18 Dec 2023, pp. 785–799. [Online]. Available: https://proceedings.mlr.press/v205/shridhar23a.html

[23] T. Gervet, Z. Xian, N. Gkanatsios, and K. Fragkiadaki, "Act3d: 3d feature field transformers for multi-task robotic manipulation," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 3949–3965. [Online]. Available: https://proceedings.mlr.press/v229/gervet23a.html

[24] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, "Rvt: Robotic view transformer for 3d object manipulation," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Tan, M. Toussaint, and K. Darvish, Eds., vol. 229. PMLR, 06–09 Nov 2023, pp. 694–710. [Online]. Available: https://proceedings.mlr.press/v229/goyal23a.html

[25] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 11 918–11 930.

[26] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[27] J. Ho, A. N. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems 33*, 2020.

[28] D. Wang, S. Hart, D. Surovik, T. Kelestemur, H. Huang, H. Zhao, M. Yeatman, J. Wang, R. Walters, and R. Platt, "Equivariant diffusion policy," in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 270. PMLR, 2025, pp. 48–69.

[29] S. Xia, H. Fang, C. Lu, and H.-S. Fang, "Cage: Causal attention enables data-efficient generalizable robotic manipulation," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025, pp. 13 242–13 249.

[30] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "Rdt-1b: a diffusion foundation model for bimanual manipulation," in *The Thirteenth International Conference on Learning Representations*, 2025.

[31] R. Yang, G. Chen, C. Wen, and Y. Gao, "Fp3: A 3d foundation policy for robotic manipulation," 2025. [Online]. Available: https://arxiv.org/abs/2503.08950

[32] G. Yan, J. Zhu, Y. Deng, S. Yang, R.-Z. Qiu, X. Cheng, M. Memmel, R. Krishna, A. Goyal, X. Wang, and D. Fox, "Maniflow: A general robot manipulation policy via consistency flow training," in *Proceedings of The 9th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Lim, S. Song, and H.-W. Park, Eds., vol. 305. PMLR, 2025, pp. 2268–2293. [Online]. Available: https://proceedings.mlr.press/v305/yan25a.html

[33] F. Zhang and M. Gienger, "Affordance-based robot manipulation with flow matching," *arXiv preprint arXiv:2409.01083*, 2024. [Online]. Available: https://arxiv.org/abs/2409.01083

[34] L. Yang, Z. Zhang, Z. Zhang, X. Liu, M. Xu, W. Zhang, C. Meng, S. Ermon, and B. Cui, "Consistency flow matching: Defining straight flows with velocity consistency," *arXiv preprint arXiv:2407.02398*, 2024.

[35] X. Hu, B. Liu, X. Liu, and Q. Liu, "Adaflow: imitation learning with variance-adaptive flow-based policies," in *Proceedings of the 38th International Conference on Neural Information Processing Systems*, ser. NIPS '24. Red Hook, NY, USA: Curran Associates Inc., 2024.

[36] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, "Consistency policy: Accelerated visuomotor policies via consistency distillation," in *Robotics: Science and Systems*, 2024.

[37] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine, "Learning complex dexterous manipulation with deep reinforcement learning and demonstrations," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018.

[38] T. Yu, D. Quillen, Z. He, R. Julian, K. Hausman, C. Finn, and S. Levine, "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, L. P. Kaelbling, D. Kragic, and K. Sugiura, Eds., vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 1094–1100.