

Appendix

Streaming Flow Policy

Simplifying diffusion/flow-matching policies by
treating action trajectories as flow trajectories

A Proof of Theorem 1

Integrating learned velocity fields can suffer from drift since errors accumulate during integration. We adding a stabilization term, we can correct deviations from the demonstration trajectory. The stabilizing velocity field is:

$$v_\xi(a, t) = \underbrace{-k(a - \xi(t))}_{\text{Stabilization term}} + \underbrace{\dot{\xi}(t)}_{\text{Path velocity}} \quad (7)$$

where $k > 0$ is the stabilizing gain. This results in exponential convergence to the demonstration:

$$\frac{d}{dt}(a - \xi(t)) = -k(a - \xi(t)) \quad (8)$$

$$\implies \frac{1}{a - \xi(t)} \frac{d}{dt}(a - \xi(t)) = -k \quad (9)$$

$$\implies \frac{d}{dt} \log(a - \xi(t)) = -k \quad (10)$$

$$\implies \log(a - \xi(t)) \Big|_0^t = - \int_0^t k dt \quad (11)$$

$$\implies \log \frac{a(t) - \xi(t)}{a_0 - \xi(0)} = -kt \quad (12)$$

$$\implies a(t) = \xi(t) + (a_0 - \xi(0))e^{-kt} \quad (13)$$

Since $a_0 \sim \mathcal{N}(\xi(0), \sigma_0^2)$ (see Eq. 1), and $a(t)$ is linear in a_0 , we have by linearity of Gaussian distributions that:

$$p_\xi(a | t) = \mathcal{N}(a | \xi(t), \sigma_0^2 e^{-2kt}) \quad (14)$$

□

B Decoupling stochasticity via latent variables

In order to learn multi-modal distributions during training, streaming flow policy as introduced in Sec. 3 requires a small amount of Gaussian noise added to the initial action. However, we wish to avoid adding noise to actions at test time. We now present a variant of streaming flow policy in an extended state space by introducing a latent variable $z \in \mathcal{A}$. The latent variable z decouples stochasticity from the flow trajectory, allowing us to sample multiple modes of the trajectory distribution at test time while deterministically starting the sampling process from the most recently generated action.

We now define a conditional flow in the extended state space $(a, z) \in \mathcal{A}^2$. We define the initial distribution by sampling a_0 and z_0 independently. a_0 is sampled from a vanishingly narrow Gaussian distribution centered at the initial action of the demonstration trajectory $\xi(0)$, but with a extremely small variance $\sigma_0 \approx 0$. z_0 is sampled from a standard normal distribution, similar to standard diffusion models [9] and flow matching [3].

Initial sample

$$z_0 \sim \mathcal{N}(0, I) \quad (15)$$

$$a_0 \sim \mathcal{N}(\xi(0), \sigma_0^2) \quad (16)$$

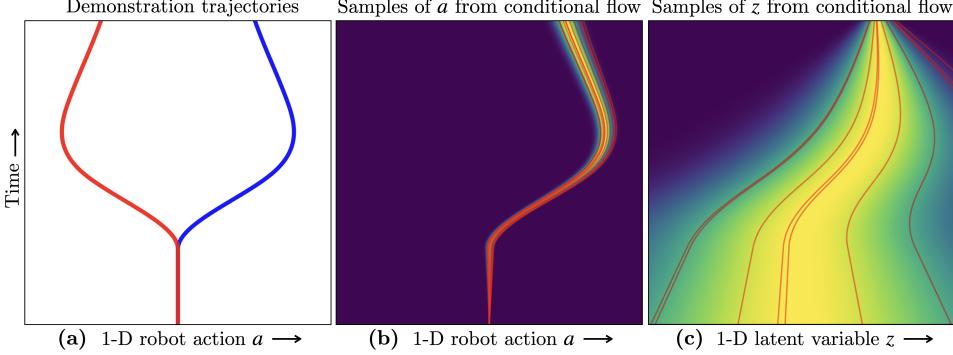


Figure 5: Constructing a conditional flow using auxiliary stochastic latent variables instead of adding noise to actions. In this toy example, the x -axis represents a 1-D action space, and the y -axis represents both trajectory time and flow time. (a) A toy bi-modal training set contains two trajectories shown in red and blue; the same as in Fig. 1a. Given a demonstration trajectory ξ from the training set (e.g. the demonstration in blue), we design a velocity field $v_\xi(a, z, t)$ that takes as input time $t \in [0, 1]$, the action a at time t , as well as an additional latent variable z . The latent variable is responsible for injecting noise into the flow sampling process, allowing the initial action $a(0)$ to be deterministically set to the initial action $\xi(0)$ of the demonstration. The latent variable $z(0) \sim \mathcal{N}(0, 1)$ is sampled from the standard normal distribution at the beginning of the flow process, similar to conventional diffusion/flow policies. The velocity field $v_\xi(a, z, t)$ generates trajectories in an extended sample space $[0, 1] \rightarrow \mathcal{A}^2$ where a and z are correlated and co-evolve with time. (b, c) Shows the marginal distribution of actions $a(t)$ and the latent variable $z(t)$, respectively, at each time step. Overlaid in red are the a - and z - projections, respectively, of trajectories sampled from the velocity field. The action evolves in a narrow Gaussian tube around the demonstration, while the latent variable starts from $\mathcal{N}(0, 1)$ at $t = 0$ and converges to the demonstration trajectory at $t = 1$; see App. B for a full description of the velocity field.

σ_0	Initial standard deviation	\mathbb{R}^+
σ_1	Final standard deviation	\mathbb{R}^+
k	Stabilizing gain	$\mathbb{R}_{\geq 0}$
σ_r	Residual standard deviation = $\sqrt{\sigma_1^2 - \sigma_0^2 e^{-2k}}$	$\mathbb{R}_{\geq 0}$

Table 4: Hyperparameters used in the stochastic variant of streaming flow policy that uses stochastic latent variables.

We assume hyperparameters σ_0 , σ_1 and k . They correspond to the initial and final standard deviations of the action variable a in the conditional flow. k is the stabilizing gain. Furthermore, we constrain them such that $\sigma_1 \geq \sigma_0 e^{-k}$. Then, let us define $\sigma_r := \sqrt{\sigma_1^2 - \sigma_0^2 e^{-2k}}$. Then we construct the joint flow trajectories of (a, z) starting from $(a(0), z(0))$ as:

Flow trajectory diffeomorphism

$$\begin{aligned} a(t | \xi, a_0, z_0) &= \xi(t) + (a_0 - \xi(0)) e^{-kt} + (\sigma_r t) z_0 \\ z(t | \xi, a_0, z_0) &= (1 - (1 - \sigma_1)t) z_0 + t \xi(t) \end{aligned} \tag{17}$$

The flow is a diffeomorphism from \mathcal{A}^2 to \mathcal{A}^2 for every $t \in [0, 1]$.

Note that $a(0 | \xi, a_0, z_0) = a_0$ and $z(0 | \xi, a_0, z_0) = z_0$, so the diffeomorphism is identity at $t = 0$. The marginal distribution at $t = 1$ for a and z is given by $a(1 | \xi) \sim \mathcal{N}(\xi(1), \sigma_1^2)$ and $z(1 | \xi) \sim \mathcal{N}(\xi(1), \sigma_1^2)$.

Intuitively, the variable a follows the shape of the action trajectory $\xi(t)$ with an error starting from $a_0 - \xi(0)$ and decreasing with an exponential factor due to the stabilizing gain. However, it uses the sampled noise variable $z_0 \sim \mathcal{N}(0, I)$ to increase the standard deviation from σ_0 around $\xi(0)$ to σ_1

around $\xi(1)$. This is done in order to sample different modes of the trajectory distribution at test time. On the other hand, the latent variable z starts from the random sample $z_0 \sim \mathcal{N}(0, I)$ but continuously moves closer to the demonstration trajectory $\xi(t)$, reducing its variance from 1 to σ_1 .

Since (a, z) at time t is a linear transformation of (q_0, z_0) , the joint distribution of (a, z) at every timestep is a Gaussian given by:

$$\begin{array}{c} \text{Joint distribution of } (a, z) \text{ at each timestep} \\ \hline \begin{bmatrix} a \\ z \end{bmatrix} = \underbrace{\begin{bmatrix} e^{-kt} & \sigma_r t \\ 0 & 1 - (1 - \sigma_1)t \end{bmatrix}}_A \begin{bmatrix} a_0 \\ z_0 \end{bmatrix} + \underbrace{\begin{bmatrix} \xi(t) - \xi(0)e^{-kt} \\ t\xi(t) \end{bmatrix}}_b \end{array} \quad (18)$$

$$p_\xi(a, z | t) = \mathcal{N}(A\mu_0 + b, A\Sigma_0 A^T) \quad (19)$$

$$= \mathcal{N}\left(\begin{bmatrix} \xi(t) \\ t\xi(t) \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{12} & \Sigma_{22} \end{bmatrix}\right) \text{ where} \quad (20)$$

$$\Sigma_{11} = \sigma_0^2 e^{-2kt} + \sigma_r^2 t^2 \quad (21)$$

$$\Sigma_{12} = \sigma_r t (1 - (1 - \sigma_1)t) \quad (22)$$

$$\Sigma_{22} = (1 - (1 - \sigma_1)t)^2 \quad (23)$$

Note that $\mu_0 = \begin{bmatrix} \xi(0) \\ 0 \end{bmatrix}$ and $\Sigma_0 = \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & 1 \end{bmatrix}$.

Since the flow is a diffeomorphism, we can invert it and express (a_0, z_0) as a function of $(a(t), z(t))$:

Inverse of the flow diffeomorphism

$$z_0 = \frac{z - t\xi(t)}{1 - (1 - \sigma_1)t} \quad (24)$$

$$a_0 = \xi(0) + (a - \xi(t) - (\sigma_r t)z_0) e^{kt}$$

At time t , the velocity of the trajectory starting from (a_0, z_0) can be obtained by differentiating the flow diffeomorphism in Eq. 17 with respect to t :

Velocity in terms of (a_0, z_0)

$$\begin{aligned} \dot{a}(t | \xi, a_0, z_0) &= \dot{\xi}(t) - k(a_0 - \xi(0)) e^{-kt} + \sigma_r z_0 \\ \dot{z}(t | \xi, a_0, z_0) &= \xi(t) + t\dot{\xi}(t) - (1 - \sigma_1)z_0 \end{aligned} \quad (25)$$

The flow induces a velocity field at every (a, z, t) . The conditional velocity field $v_\theta(a, z, t | h)$ by first inverting the flow transformation as shown in Eq. 24, and plugging that into Eq. 25, we get:

Conditional velocity field

$$\begin{aligned} v_\xi^a(a, z, t) &= \dot{\xi}(t) - k(a - \xi(t)) + \frac{\sigma_r(1 + kt)}{1 - (1 - \sigma_1)t} (z - t\xi(t)) \\ v_\xi^z(a, z, t) &= \xi(t) + t\dot{\xi}(t) - \frac{1 - \sigma_1}{1 - (1 - \sigma_1)t} (z - t\xi(t)) \end{aligned} \quad (26)$$

Importantly, the evolution of a and z is inter-dependent *i.e.* the sample z_0 determines the evolution of a . Furthermore, the marginal probability distribution $p_\xi^a(a, t)$ can be deduced from the joint distribution in Eq. 20 and is given by:

$$p_\xi(a | t) = \mathcal{N}(a | \xi(t), \sigma_0^2 e^{-2kt} + \sigma_r^2 t^2) \quad (27)$$

In other words, q evolves in a Gaussian tube centered at the demonstration trajectory $\xi(t)$ with a standard deviation that varies from σ_0 at $t = 0$ to σ_1 at $t = 1$. The fact that the marginal distribution lies close to the demonstration trajectories, from Eq. 5 ensures that the per-timestep marginal distributions over actions induced by the learned velocity field are close to training distribution. However, this formulation allows us to select extremely small values of σ_0 , essentially deterministically starting from the last generated action a_{prev} . The stochasticity injected by sampling $z_0 \in \mathcal{N}(0, I)$, as well as the correlated evolution of a and z ensures that we sample a diverse distribution of actions in a starting from the same action a_{curr} . This phenomenon is illustrated via a 1-D toy example in Figs. 5 and 6, with details in captions.

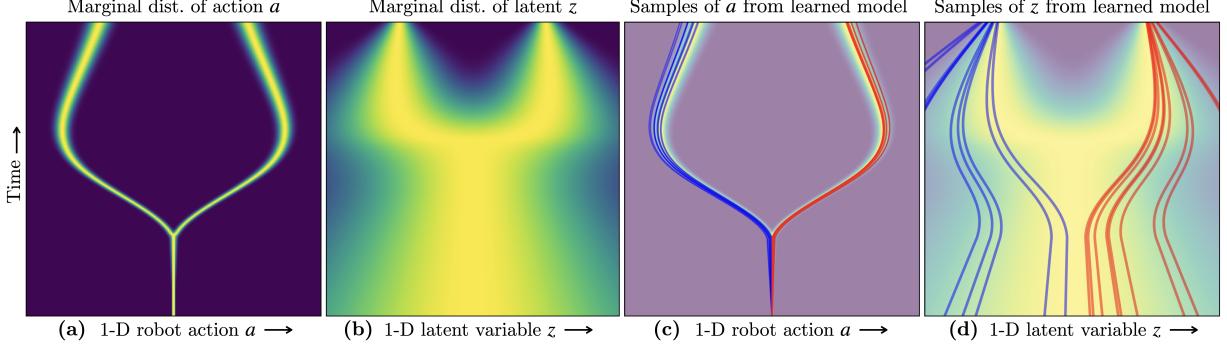


Figure 6: The marginal velocity flow field $v_\theta(a, z, t | h)$ learned using the flow construction in Fig. 5. **(a, b)** shows the marginal distribution of actions $a(t)$ and the latent variable $z(t)$, respectively, at each time step under the learned velocity field. **(c, d)** Shows the a - and z - projections, respectively, of trajectories sampled from the learned velocity field. By construction, $a(0)$ deterministically starts from the most recently generated action, whereas $z(0)$ is sampled from $\mathcal{N}(0, 1)$. Trajectories starting with $z(0) < 0$ are shown in blue, and those with $z(0) > 0$ are shown in red. The main takeaway is that in **(c)**, even though all samples deterministically start from the same initial action (*i.e.* the most recently generated action), they evolve in a stochastic manner that covers both modes of the training distribution. This is possible because the stochastic latent variable z is correlated with a , and the initial random sample $z(0) \sim \mathcal{N}(0, 1)$ informs the direction a evolves in.

C Action Horizon

In Fig. 7, we analyze the effect of action chunk size on the performance of streaming flow policy, under various benchmark environments: (1) Robomimic: Can, (2) Robomimic: Square, (3) Push-T with state input and (4) Push-T with image input. The x -axis shows the chunk size in log scale. The y -axis shows the relative decrease in performance compared to that of the best performing chunk size. All scores are less than or equal to zero, where higher is better. In 3/4 environments, the performance peaks at chunk size 8, and 1/4 environments peak at chunk size 6. The performance decreases as the chunk size deviates from the optimum. Our results match with findings from Chi et al. [1], suggesting that behavior cloning policies have a “sweet spot” in the chunk size of the action trajectories. We recommend choosing a larger chunk size (*i.e.* closer to open-loop execution) when the environment dynamics are deterministic and stable. Smaller chunk sizes should be used in stochastic environments with high uncertainty, where the policy may benefit from a tighter feedback loop.

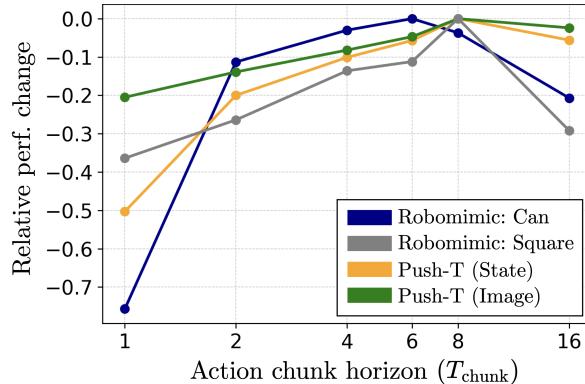


Figure 7: Analysis of the effect of action chunk size on the performance of streaming flow policy, under various benchmark environments. x -axis shows the chunk size, in log scale. y -axis shows the relative decrease in performance compared to that of the best performing chunk size. All scores are less than or equal to zero, where higher is better. In 3/4 environments, the performance peaks at chunk size 8, and the other environment peaks at chunk size 6. The performance decreases as the chunk size increases or decreases from the optimum.

D Push-T experiments with image inputs and action imitation

In this section, we perform experiments in the Push-T environment [1, 16] using images as observations, and imitating actions instead of states (see Sec. 6 for a discussion on state imitation vs. action imitation). This was missing in Table 5 of the main paper.

The conclusions from the table are essentially the same as in the main paper. Streaming flow policy performs nearly as well as the best performing baseline *i.e.* diffusion policy with 100 DDPM inference steps. However, streaming flow policy is significantly faster than diffusion policy. It is also faster than the remaining baselines, while also achieving a higher task success rate.

		Push-T with image input	
		Action imitation Avg/Max scores	Latency
		↑	↓
1	DP [1]: 100 DDPM steps	83.8% / 87.0%	127.2 ms
2	DP [1]: 10 DDIM steps	80.8% / 85.5%	10.4 ms
3	Flow matching policy [5]	71.0% / 72.0%	12.9 ms
4	Streaming DP [14]	80.5% / 83.9%	77.7 ms
5	SFP (Ours)	82.5% / 87.0%	08.8 ms

Table 5: Imitation learning accuracy on the Push-T [1] dataset with images as observation inputs, and imitating action trajectories. █ Our method (in green) compared against █ baselines (in red). See text for details.