

Appendix

Streaming Flow Policy

Simplifying diffusion/flow policies by treating robot trajectories as flow trajectories

Sunshine Jiang, Xiaolin Fang, Nicholas Roy, Tomás Lozano-Pérez, Leslie Kaelbling, Siddharth Ancha

Website: <https://streaming-flow-policy.github.io>[†]

APPENDIX I PROOF OF THEOREM 1

Integrating learned velocity fields can suffer from drift since errors accumulate during integration. We introduce a stabilizing variant that actively corrects deviations from the demonstration trajectory. The stabilizing velocity field is:

$$v_\xi(q, t) = \underbrace{-k(q - \xi(t))}_{\text{Stabilization term}} + \underbrace{\dot{\xi}(t)}_{\text{Path velocity}} \quad (6)$$

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

$$\frac{d}{dt}(q - \xi(t)) = -k(q - \xi(t)) \quad (7)$$

$$\implies \frac{1}{q - \xi(t)} \frac{d}{dt}(q - \xi(t)) = -k \quad (8)$$

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

$$\implies \log(q - \xi(t)) \Big|_0^t = - \int_0^t k dt \quad (10)$$

$$\implies \log \frac{q(t) - \xi(t)}{q_0 - \xi(0)} = -kt \quad (11)$$

$$\implies q(t) = \xi(t) + (q_0 - \xi(0))e^{-kt} \quad (12)$$

Since $q_0 \sim \mathcal{N}(\xi(0), \sigma_0^2)$, and $q(t)$ is linear in q_0 , we have by linearity of Gaussian distributions that:

$$p_\xi(q | t) = \mathcal{N}(q | \xi(t), \sigma_0^2 e^{-2kt}) \quad (13)$$

□

APPENDIX II

DECOUPLING STOCHASTICITY VIA LATENT VARIABLES

In order to learn multi-modal distributions during training, streaming flow policy as introduced in Sec. III requires a small amount of Gaussian noise added to the start configuration. However, we wish to avoid adding noise to configurations at test time. We now present a variant of streaming flow policy in an extended state space by introducing a latent variable $z \in C$. 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 with the current robot configuration.

σ_0	Initial standard deviation	$\mathbb{R}_{>0}$
σ_1	Final standard deviation	$\mathbb{R}_{>0}$
σ_r	Residual std. deviation = $(\sigma_1^2 - \sigma_0^2)^{1/2}$	$\mathbb{R}_{>0}$

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

We now define a conditional flow in the extended state space $(q, z) \in \mathcal{C}^2$. We define the initial distribution by sampling q_0 and z_0 independently. q_0 is sampled from a vanishingly narrow Gaussian distribution centered at the initial configuration 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].

$$z_0 \sim \mathcal{N}(0, I) \quad (14)$$

$$q_0 \sim \mathcal{N}(\xi(0), \sigma_0^2) \quad (15)$$

We assume hyperparameters σ_0 and σ_1 such that $\sigma_1 \geq \sigma_0$. They correspond to the initial and final standard deviations of the configuration variable q in the conditional flow. Let us define $\sigma_r := \sqrt{\sigma_1^2 - \sigma_0^2}$. Then we construct the joint flow trajectories of (q, z) starting from (q_0, z_0) as:

$$q(t | \xi, q_0, z_0) = q_0 + (\xi(t) - \xi(0)) + (\sigma_r t) z_0 \quad (16)$$

$$z(t | \xi, q_0, z_0) = (1 - (1 - \sigma_1)t) z_0 + t \xi(t) \quad (17)$$

Intuitively, the variable q starts from q_0 and flows along the shape of the demonstration trajectory $\xi(t)$. 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 $\xi(1)$. 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 the flow is a diffeomorphism, we can invert it and express (q_0, z_0) as a function of $(q(t), z(t))$:

$$z_0 = \frac{z(t) - t \xi(t)}{1 - (1 - \sigma_1)t} \quad (18)$$

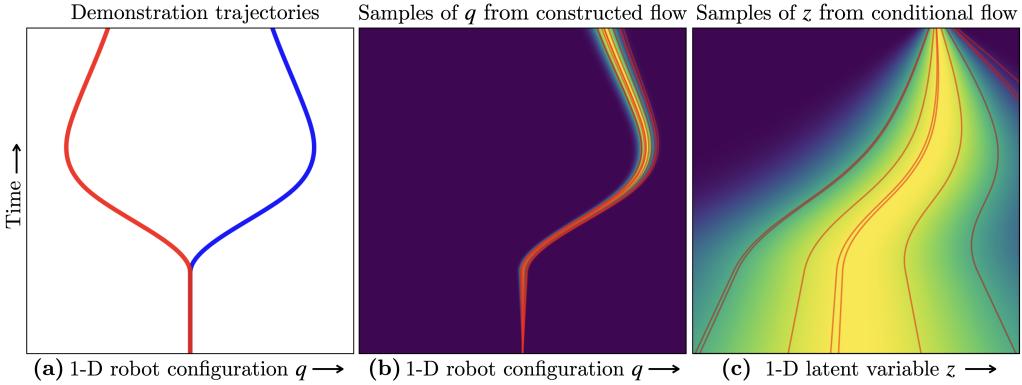


Fig. 2: Constructing a conditional flow using auxiliary stochastic latent variables instead of adding noise to robot configurations. In this toy example, the x -axis represents a 1-D configuration space, and the y -axis represents both trajectory time and flow time. **(a)** A toy bi-modal training set contains two trajectories shown in red and blue; the same as in Fig. 1a. Given a demonstration trajectory from the training set (e.g. the demonstration in blue), we design a velocity field $v(q, z, t | h)$ that takes as input time $t \in [0, 1]$, the configuration q 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 robot configuration $q(0)$ to be deterministically set to the initial configuration 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 $v(q, z, t | h)$ generates trajectories in an extended sample space $[0, 1] \rightarrow \mathcal{C}^2$ where q and z are correlated and co-evolve with time. **(b, c)** Shows the marginal distribution of configurations $q(t)$ and the latent variable $z(t)$, respectively, at each time step. Overlaid in red are the q - and z - projections, respectively, of trajectories sampled from the velocity field. The configuration 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. II for a full description of the velocity field.

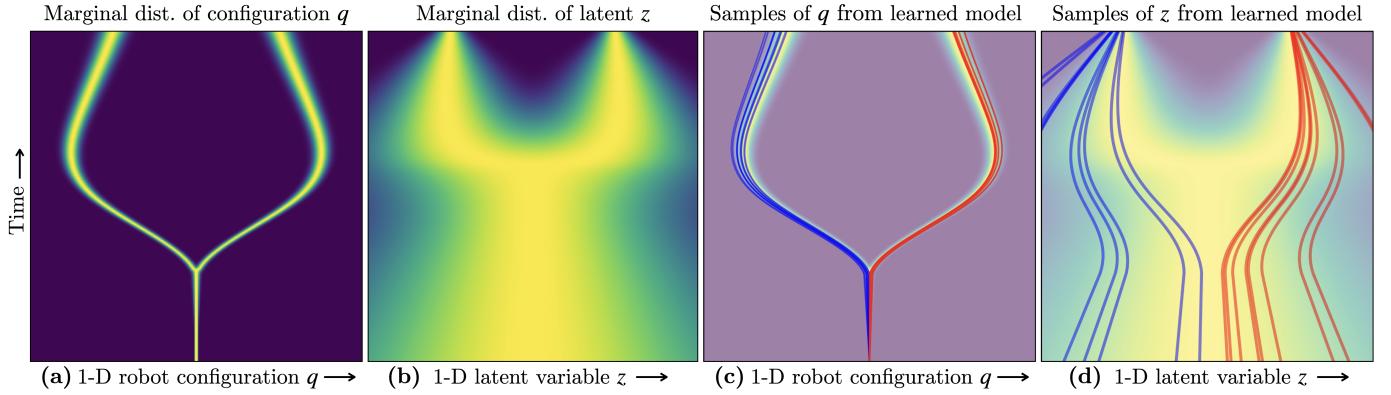


Fig. 3: The marginal velocity flow field $v_\theta(q, z, t | h)$ learned using the flow construction in Fig. 2. **(a, b)** shows the marginal distribution of configurations $q(t)$ and the latent variable $z(t)$, respectively, at each time step under the learned velocity field. **(c, d)** Shows the q - and z - projections, respectively, of trajectories sampled from the learned velocity field. By construction, $q(0)$ deterministically starts from the current robot configuration, 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 configuration (i.e. the current robot configuration), 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 q , and the initial random sample $z(0) \sim \mathcal{N}(0, 1)$ informs the direction q evolves in.

$$q_0 = q(t) - (\xi(t) - \xi(0)) - (\sigma_r t) z_0 \quad (19)$$

Differentiating the flow equations with respect to t yields:

$$\dot{q}(t | \xi, q_0, z_0) = \dot{\xi}(t) + \sigma_r z_0 \quad (20)$$

$$\dot{z}(t | \xi, q_0, z_0) = \xi(t) + t\dot{\xi}(t) - (1 - \sigma_1) z_0 \quad (21)$$

To compute the corresponding velocity field, we substitute (q_0, z_0) in terms of (q, z) at time t .

$$v_\xi^q(q, z, t) = \dot{\xi}(t) + \frac{\sigma_r(z - t\xi(t))}{1 - (1 - \sigma_1)t} \quad (22)$$

$$v_\xi^z(q, z, t) = \xi(t) + t\dot{\xi}(t) - \frac{1 - \sigma_1}{1 - (1 - \sigma_1)t}(z - t\xi(t)) \quad (23)$$

Importantly, the evolution of q and z is inter-dependent i.e. the sample z_0 determines the evolution of q . Furthermore, the marginal probability distribution $p_\xi^q(q, t)$ has a simple form given by:

$$p_\xi(q | t) = \mathcal{N}(q | \xi(t), \sigma_0^2(1 - t^2) + \sigma_1^2 t^2) \quad (24)$$

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 configurations induced by the learned

velocity field are close to training distributions. However, this formulation allows us to select extremely small values of σ_0 , essentially deterministically starting from the current configuration q_{curr} . The stochasticity injected by sampling $z_0 \in \mathcal{N}(0, I)$, as well as the correlated evolution of q and z ensures that we sample a diverse distribution of trajectories in q starting from the same configuration $q_0 = q_{\text{curr}}$. This phenomenon is illustrated via a 1-D toy example in Figs. 2 and 3, with details in captions.

APPENDIX III CAVEATS

In this section, we discuss some limitations of our approach.

A. SFP does not match joint distribution, only per-timestep marginal distributions

Our flow matching framework ensures that the learned distribution over trajectories conditioned on the history matches the training distribution in terms of marginal distributions of configurations at each timestep $t \in [0, 1]$. We however, do not guarantee that the joint distribution of configurations within a trajectory matches that of the training distribution. This is in contrast to diffusion policy, that is able to match the joint distribution since diffuses over entire trajectories.

Figs. 4 and 5 illustrates a toy example where streaming flow policy matches marginal distributions but not the joint distribution. The x -axis represents 1-D robot configurations, and the y -axis represents flow time ($t \in [0, 1]$). Fig. 4a shows two trajectories in blue and red, of shapes “S” and “Z” respectively. The trajectories intersect at $t = 0.5$. The learned flow field is shown in Fig. 4c, and the induced marginal distribution over configurations is shown in Fig. 4d. The marginal distribution of configurations matches the training distribution at each $t \in [0, 1]$. Trajectories sampled from the flow field are shown in Fig. 4d. The trajectory distribution contains two modes of equal probability: trajectories that always lie either in $q < 0$ (shown in blue), or in $q > 0$ (shown in red). The shapes formed by sampled trajectories — “E” and “3” respectively — do not match the shapes of trajectories in the training data.

A similar phenomenon is illustrated in Fig. 5 using the “stochastic” variant of streaming flow policy (see App. II) trained on the same dataset of intersecting trajectories. While the marginal distribution of configuration again matches with the training distribution, the trajectories contain *four* modes, with shapes “S”, “Z”, “E” and “3”. Due to stochasticity introduced by decoupled latent variables, trajectories in blue (alternatively, red) that start from $q < 0$ (alternatively, $q > 0$) are able to split in either direction at the point of intersection.

B. Streaming flow policies exhibit compositionality

While the loss of fidelity to the joint distribution of configurations can be interpreted as a limitation of streaming flow policy, another perspective is to think of our method as providing *compositionality* over training demonstrations. The sampled trajectories can be composed of pieces across the training data.

For many robotics tasks, compositionality might be both valid and desirable. For example, in quasi-static tasks where the robot moves slowly, if two demonstration trajectories are valid, then the compositions across these trajectories are often also valid. Under this assumption, compositionality allows the flow model to learn many valid combinations of partial trajectories with fewer demonstrations.

What constraints on trajectories reflected in the training data can streaming flow policy learn? Streaming flow policy is unable to capture global constraints that can only be represented in the joint distribution. However, it can learn certain local constraints.

C. SFPs can learn arbitrary position constraints

Robot configurations $q \in Q \subseteq \mathcal{C}$ may be constrained to lie in a subset $Q \subseteq \mathcal{C}$ of the configuration space. For example, Q may reflect joint limits of a robot arm. Then, a well-trained streaming flow policy should learn this constraint as well.

To see why, consider Eq. 5 which states the learned marginal density of configurations $p^*(q | t, h) = \int_{\xi} p_{\xi}(q | t) p_{\mathcal{D}}(\xi | h) d\xi$ at time t is a weighted average of marginal densities of conditional flows $p_{\xi}(q | t)$. Recall that we construct $p_{\xi}(q | t)$ to be thin Gaussian tubes around demonstration trajectories ξ . Assume that the thickness of the Gaussian tube is sufficiently small that $q \notin Q \implies p_{\xi}(q | t) < \epsilon$, for some small $\epsilon > 0$ and for all ξ, t . Then we have from Eq. 5 that $p_{\xi}(q | t) < \epsilon \implies p^*(q | t, h) < \epsilon$ for all $t \in [0, 1]$. Therefore, the probability of sampling a configuration q that violates the constraint Q is extremely low.

D. SFPs can learn convex velocity constraints

Theorem 2 of Lipman et al. [3] implies that the minimizer of the conditional flow matching loss $v^* := \arg \min_v \mathcal{L}_{\text{CFM}}(v, p_{\mathcal{D}})$ has the following form:

$$\begin{aligned} v^*(q, t | h) &= \int_{\xi} v_{\xi}(q, t) \underbrace{\frac{p_{\mathcal{D}}(\xi | h) p_{\xi}(q | t)}{\int_{\xi'} p_{\mathcal{D}}(\xi' | h) p_{\xi'}(q | t) d\xi'}}_{p_{\mathcal{D}}(\xi | q, t, h)} d\xi \\ &= \int_{\xi} v_{\xi}(q, t) p_{\mathcal{D}}(\xi | q, t, h) d\xi \\ &\approx \int_{\xi} \dot{\xi}(t) p_{\mathcal{D}}(\xi | q, t, h) d\xi \quad (\text{assuming } k \approx 0) \end{aligned} \quad (25)$$

Intuitively, the target velocity field v^* at (q, t) is a weighted average of conditional flow velocities $v_{\xi}(q, t)$ over demonstrations ξ . The weight for ξ is the Bayesian posterior probability of ξ , where the prior probability $p_{\mathcal{D}}(\xi | h)$ is the probability of ξ given h in the training distribution, and the likelihood $p_{\xi}(q | t)$ is the probability that the conditional flow around ξ generates q at time t .

Under sufficiently small values of k , we have from Eq. 1 that $v_{\xi}(q, t) \approx \dot{\xi}(t)$. Note that v^* is then a convex combination of demonstration velocities $\dot{\xi}(t)$. Consider convex constraints over velocities $\dot{\xi}(t) \in C$ i.e. $\dot{\xi}(t)$ is constrained to lie in a convex set C for all ξ with non-zero support $p_{\mathcal{D}}(\xi) > 0$ and for all $t \in [0, 1]$. This is the case, for example, when robot

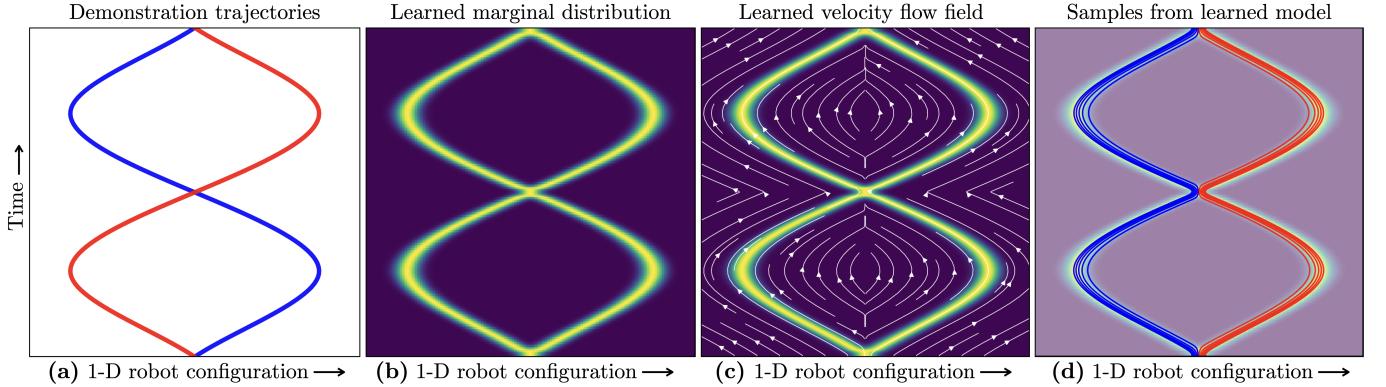


Fig. 4: A toy example illustrating how streaming flow policy matches marginal distribution of configurations in the trajectory at all time steps, but not necessarily their joint distribution. The x -axis represents a 1-D configuration space, and the y -axis represents both trajectory time and flow time. **(a)** The bi-modal training set contains two intersecting demonstration trajectories, illustrated in blue and red, with shapes “S” and “Z” respectively. **(b)** The marginal distribution of configurations at each time step learned by our streaming flow policy. The marginal distributions perfectly match the training data. **(c)** The learned velocity flow field $v_\theta(q, t | h)$ that yeilds the marginal distributions in **(b)**. **(d)** Trajectories sampled from the learned velocity field. Trajectories that start from $q < 0$ are shown in blue, and those starting from $q > 0$ are shown in red. The sampled trajectories have shapes “E” and “3”, with equal probability. These shapes are different from the shapes “S” and “Z” in the training distribution, although their margin distributions are identical.

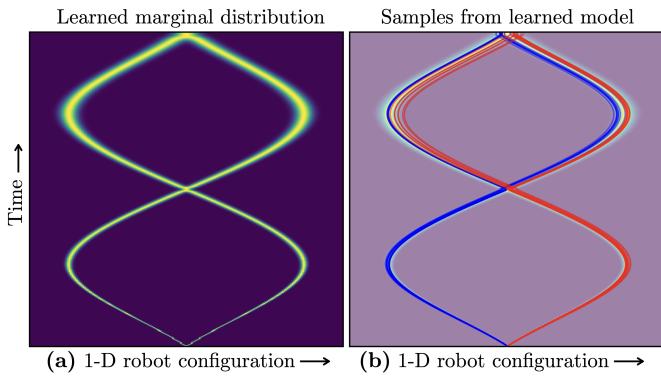


Fig. 5: Different variants of streaming flow policy can produce different joint distributions of configurations that are consistent with the marginal distributions in the training data. This example is produced using the stochastic version of streaming flow policy, described in App. II. **(a)** The marginal distribution of configurations at each time step learned by the stochastic streaming flow policy matches the training data. **(b)** Samples from the trained policy produces *four* modes with shapes “S”, “Z”, “E” and “3”, whereas the training data contains only two modes with shapes “S” and “Z”. Due to stochasticity introduced by decoupled latent variables, trajectories in blue (alternatively, red) that start from $q < 0$ (alternatively, $q > 0$) are able to split in either direction at the point of intersection.

joint velocities lie in a closed interval $[v_{\min}, v_{\max}]$. Then, Eq. 25 implies that v^* also lies in C .