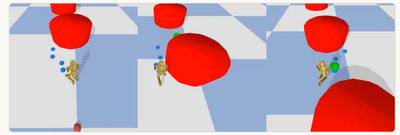
Imitation-Free Diffusion Policy Training for Humanoid Footstep Planning

Radman Rakhshandehroo[1] · Nicholas Ioannidis[2] · Michiel van de Panne[1]

[1]University of British Columbia [2]Simon Fraser University

KEYWORDS · DIFFUSION POLICY · IMITATION-FREE LEARNING · HUMANOID LOCOMOTION · FOOTSTEP PLANNING · VIABILITY FILTERING



QUESTION A frozen ALLSTEPS controller steps where it is told, but on its own it trips on hurdles, walks into obstacles, and stumbles onto raised platforms. So we plan the footsteps for it. Can that planning pipeline skip its offline data-collection step? We swap the diffusion model for a hand-designed stochastic sampler and train the viability filter on-policy from environment rollouts. **The broader question: does plan quality depend more on the evaluator than on the generator?**

Three pipelines

Behaviour-cloned diffusion (1) is the textbook recipe. Ioannidis et al. (2025) bolt a learned viability filter onto its output (2). We drop the offline dataset entirely and let a stochastic sampler propose plans for an online-trained filter to score (3).

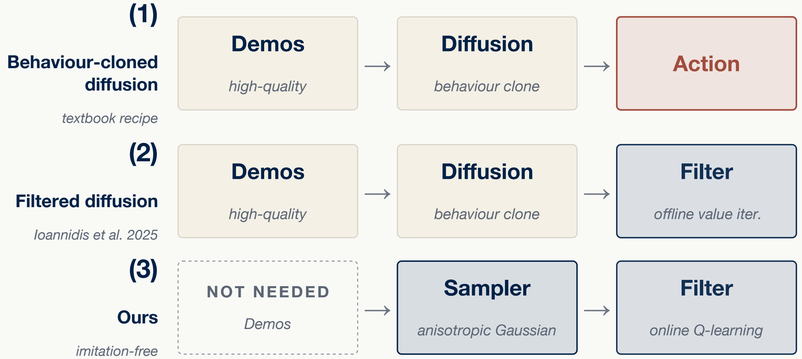


Figure 1. Three pipelines. Moving (1)→(2)→(3) trades a fancier generator for more inference-time sampling. The scorer takes over the work and the offline dataset goes away.

Generator vs. evaluator capacity

Generator capacity is how much structure the proposer encodes (random < Gaussian < diffusion). **Evaluator capacity** is how sharply the filter separates viable from unviable plans. The prior baseline sits top-right (strong generator + strong evaluator). Ours is the top-left: noise generator, strong evaluator. BC and Random fix the floor of the design space.

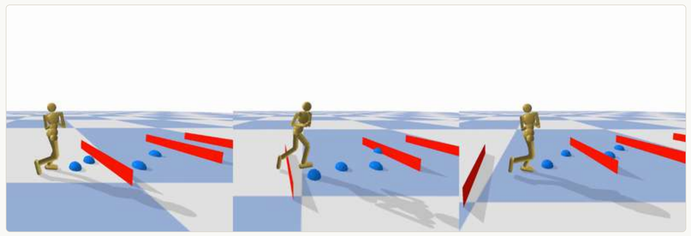
Figure 4. Generator × evaluator capacity. Ours collapses the generator to noise and asks whether the evaluator can still carry the load.

Generator

We replace the diffusion model with a hand-designed anisotropic Gaussian sampler. At every footstep, 100 candidate plans are drawn from

$$p(x) \propto \exp\left\{-\frac{1}{2}\left(\frac{\Delta_{\parallel}^2}{\sigma_{\parallel}^2}+\frac{\Delta_{\perp}^2}{\sigma_{\perp}^2}\right)\right\}$$

with $\sigma_{\parallel}$ along travel and $\sigma_{\perp}$ perpendicular to it ($\sigma_{\parallel} > \sigma_{\perp}$). **The sampler has no learned weights and never sees a demonstration.**



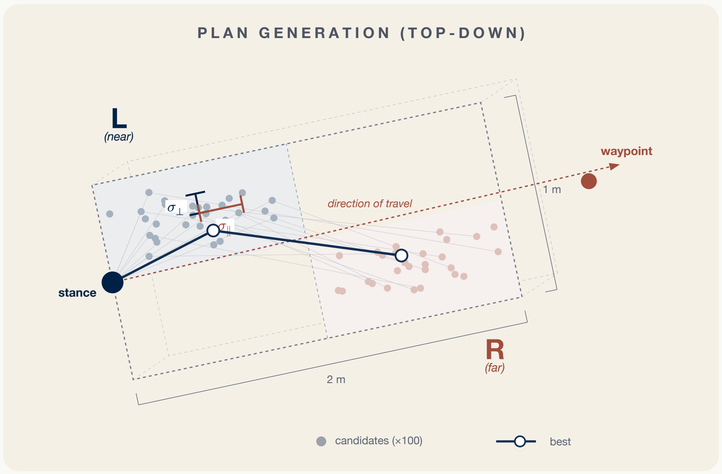


Figure 2. Plan generation, in context. Top: a hurdle terrain in PyBullet, with the humanoid we plan for. Bottom: a 1 × 2 m rectangle is projected from the stance foot, so its lateral midline lines up with the stance→waypoint line. L candidates sample above the midline, R below. The selected plan (green) goes to the controller.

Evaluator

We keep the same alive/fail Q-function as [1]. It reads the current character state plus an ego-centric height map of the terrain around each candidate step. Training is on-policy: Q-learning with prioritized replay, ramped through a four-level performance-gated curriculum. A frozen ALLSTEPS controller then executes the winning plan.

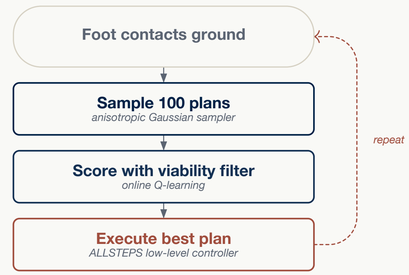


Figure 3. Planning loop. Sample → score → execute, repeating per footstep. The filter learns online from rollout outcomes.

Takeaway

The evaluator matters more than the generator.

Swap the trained diffusion model for a hand-designed stochastic sampler and we still get viable footstep plans, as long as the viability filter trains on-policy. The scorer was doing most of the work all along.

IF OURS MATCHES

The offline dataset was doing less work than assumed; the evaluator alone is enough.

IF OURS LOSES

The diffusion prior encodes structure that rollouts alone cannot recover.

Key references

1. Ioannidis, Reda, Cohan, van de Panne. **Diffusion-based planning with learned viability filters.** PACMCGIT 8(4), 2025.
2. Chi et al. **Diffusion policy.** RSS 2023.
3. Janner, Du, Tenenbaum, Levine. **Planning with diffusion.** ICML 2022.
4. Xu, Shi, Yin, Peng. **PARC.** SIGGRAPH 2025.
5. Xie, Ling, Kim, van de Panne. **ALLSTEPS.** CGF 39(8), 2020.