P3: PERSISTENT PARTICLE PLANNING FOR CONSTRAINED DIFFUSION CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models provide expressive priors over trajectories, but adapting these priors to test-time constraints requires maintaining feasibility and consistency across successive control decisions. We introduce Persistent Particle Planning (P3), a sequential Monte Carlo framework for diffusion control that maintains a weighted population of candidate plans across replanning steps. At each control step, P3 shifts and partially re-noises the candidate trajectories, refines them under the latest observation, and uses constraint-aware weighting and resampling to select among alternative continuations without retraining the diffusion model. We consider denoising and replanning as one Feynman–Kac particle system and analyze it under an idealized repair. We prove that the re-noising depth controls how reliably a kept plan stays on its route, and that keeping a rare, well-separated route takes far fewer plans than rediscovering it by sampling from scratch. Experiments under multiple test-time constraint configurations show that population reuse reduces route switching and improves success without constraint violations. Because P3 refines earlier plans instead of redrawing them, it also needs fewer denoising iterations per replan. On maze-navigation tasks, it plans faster than both regenerated populations and methods that correct a single sampled plan by constrained optimization. Code and pretrained models are available at https://github.com/p3-username/p3-anon.

1 INTRODUCTION

Diffusion models represent multiple ways of completing a task as a distribution over trajectories (Janner et al., 2022; Chi et al., 2023). Requirements not included in the training data, such as an added obstacle, must then be imposed at test time, typically by correcting each sample during denoising, for instance by projecting it onto the feasible set (Christopher et al., 2024; Römer et al., 2025). Such corrections act on the samples of a single planning call.

A controller, however, solves a sequence of planning problems. In receding-horizon control, it executes the first action of a finite-horizon plan and replans from the next observation (Mayne et al., 2000), so consecutive problems share almost their entire horizon. Regenerating the plan from noise at every step discards that shared solution and lets the controller alternate between modes, for example passing an obstacle on one side at one step and on the other at the next. Diffusion controllers reduce changes between successive plans by ensembling or inpainting overlapping action chunks (Zhao et al., 2023; Black et al., 2025), selecting the sample most coherent with the previous decision (Liu et al., 2025), or warm-starting the next call from the previous plan by partial noising and denoising, which Diffuser introduced to cut planning cost (Janner et al., 2022) and which RDM (Zhou et al., 2023) and streaming diffusion policies (Høeg et al., 2025) build into replanning. Warm-starting a single plan reuses the previous trajectory, but constraint weighting cannot affect which plan is executed when there is only one candidate.

We introduce *persistent particle planning* (P3), which retains a weighted population of plans within and across replanning calls (Figure 1). Within each planning call, P3 runs sequential Monte Carlo (SMC) over the denoising chain (Gordon et al., 1993; Del Moral, 2004; Chopin & Papaspiliopoulos, 2020), as prior work does for conditional and reward-guided diffusion (Trippe et al., 2023; Wu et al., 2023; Kim et al., 2025; Singhal et al., 2025). Particles follow the model's own denoising steps, and the constraints act only by reweighting and resampling them according to the violations of their

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

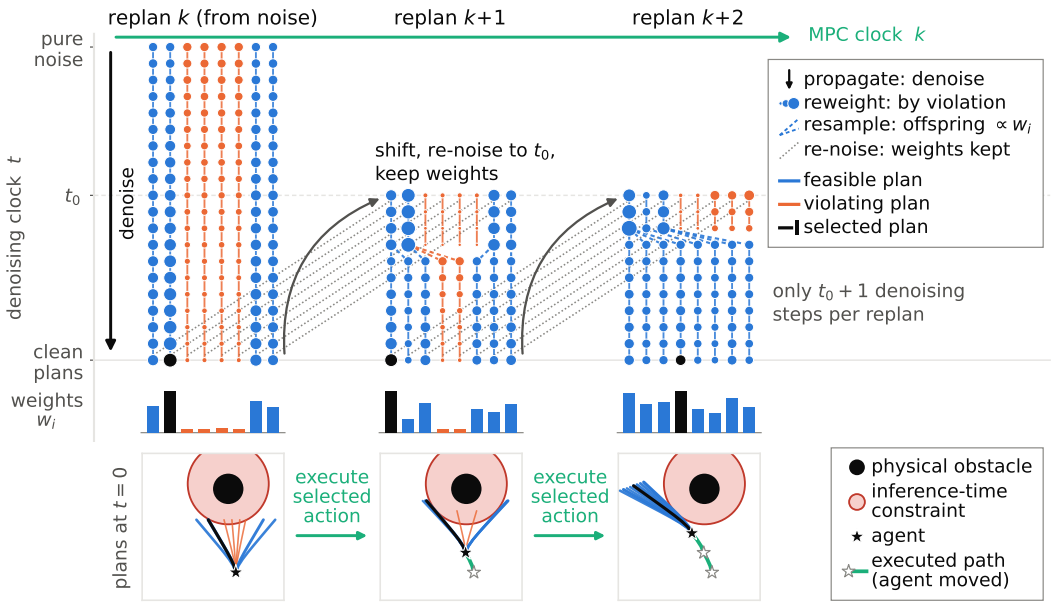


Figure 1: Persistent particle planning. Within each planning call (denoising clock, vertical), particles are denoised, weighted by their constraint violation and resampled. After the selected plan's first action is executed (control clock, horizontal), the population is shifted and re-noised to t_0 rather than redrawn from noise. Bottom: the corresponding plans around an obstacle and a test-time exclusion region.

predicted plans. Across calls, P3 applies the shift-and-re-noise warm start to every particle and carries the weights forward. The population and its selection history become part of the controller's state. To separate keeping plans from selecting among them, we run the two population samplers, best-of-N and SMC, both with and without carrying the population. Our contributions are:

- **Method.** P3 carries a constraint-weighted population of diffusion plans across control steps. It updates each plan by shifting and partially re-noising it instead of redrawing it, and it can rejuvenate the population with new plans. It needs no retraining, and a warm call runs only $t_0 + 1$ of the K denoising steps (Section 3).
- **Theory.** We consider denoising and replanning as a single Feynman–Kac model and analyze it under an idealized repair. The re-noising depth controls how reliably a kept plan stays on its route. A rare but well-separated route can be kept with far fewer plans than sampling from scratch needs to rediscover it. For independent acceptable plans drawn from the model, one exact repair at a fixed observation is at least as likely, on average, to leave an acceptable plan as drawing a fresh population of the same size (Section 3.4).
- **Evidence.** We evaluate on obstacle-avoidance and maze-navigation benchmarks whose constraints are added only at test time. Persistence helps most where a feasible route is hard to find from scratch. There, P3 reaches the goal without violations far more often than a population of the same size regenerated at every step. Elsewhere, it matches or exceeds regenerated populations. Controls separate the two effects. Keeping plans reduces route switching, and constraint weighting and resampling further raise success without violations on several tasks. On maze navigation, P3 also plans faster than regenerating the population or correcting a single plan by optimization (Section 4).

2 Related Work

Diffusion models for planning and control. In decision making, diffusion models have been used as trajectory generators whose samples are steered toward rewards or goals by guidance (Janner et al., 2022; Ajay et al., 2023), as visuomotor policies that denoise action chunks and execute them

in a receding-horizon loop (Chi et al., 2023), and as motion planners that combine the learned prior with collision costs (Carvalho et al., 2023). These models can represent multiple ways of completing a task (Shafiullah et al., 2022; Jia et al., 2024). Model-based variants derive the score from a dynamics model rather than from data (Pan et al., 2024). P3 keeps the pretrained model and its transitions fixed and acts only on which samples survive and which are reused.

Constraints at inference time. Test-time constraints are usually imposed by modifying the sampling path. Gradient guidance adds the gradient of a constraint cost to the score (Dhariwal & Nichol, 2021; Janner et al., 2022), control-barrier guidance enforces finite-time invariance of the denoising dynamics (Xiao et al., 2025), and projection methods map each iterate or its clean prediction onto the feasible set (Christopher et al., 2024; Römer et al., 2025); DiRecT argues that projecting noisy iterates over-constrains the sampler and instead enforces the constraint on the final sample through a receding-horizon optimal-control formulation (Giaretta et al., 2026). A separate line changes the model so that its state space is the constrained domain (Lou & Ermon, 2023; Fishman et al., 2023). These approaches modify the sampling dynamics or the samples themselves. P3 leaves the proposal untouched and reweights its reference path law. Unlike exact projection, reweighting does not place excluded mass on a previously null constraint boundary; its accumulated path potentials can still change the relative masses of feasible routes (Appendices A.2 and A.6).

Sequential Monte Carlo for diffusion sampling. SMC approximations to Feynman–Kac models have been used for conditional and reward-guided diffusion sampling. SMC over the denoising chain conditions on partial observations (Trippe et al., 2023); twisted proposals with telescoped potentials give asymptotically exact conditional samplers (Wu et al., 2023); and filtering formulations address linear inverse problems (Cardoso et al., 2024; Dou & Song, 2024). Related SMC methods are used for inference-time reward alignment (Li et al., 2024; Kim et al., 2025; Singhal et al., 2025; Skreta et al., 2025), diverse batch sampling through joint potentials (Corso et al., 2024), and search over initial noises (Ma et al., 2025). The choice of potential affects sampling efficiency, and twisting functions can be learned (Whiteley & Lee, 2014; Guarniero et al., 2017; Heng et al., 2020; Zhao et al., 2024; Wang et al., 2026). These methods sample one target with a population created from noise and discarded afterward. P3 uses a plain plug-in potential, to which learned twisting could be added, and keeps the population, its weights and its lineages across replanning calls; for fixed observations this is one Feynman–Kac model spanning denoising and replanning (Section 3).

Replanning, warm starts, and populations in control. Sampling-based model predictive control carries information across control steps. MPPI and cross-entropy planners shift and perturb the previous control sequence (Williams et al., 2017; Chua et al., 2018), SMC solves the open-loop problem at each step (Kantas et al., 2009; Piché et al., 2019), and Stein variational MPC keeps control-sequence particles between steps to preserve modes (Lambert et al., 2020); particle filters recover from a lost track by injecting fresh particles (Thrun et al., 2005) and diversify by invariant moves (Gilks & Berzuini, 2001). Diffuser warm-starts each plan by partially noising and denoising the previous one (Janner et al., 2022; Meng et al., 2022), RDM reuses one plan until the model's likelihood triggers regeneration (Zhou et al., 2023), and Streaming Diffusion Policy rolls a partially denoised trajectory forward (Høeg et al., 2025); all keep one trajectory. Diffusion-ES mutates high-reward diffusion samples by partial noising and denoising, and can warm-start its population with solutions from the previous control step (Yang et al., 2024). Guided flow-matching samples have also been refined by MPPI with warm starts (Mizuta & Leung, 2025). P3 weights and resamples particles during denoising, then carries the plans and their weights across control steps. It repairs this population by partial noising and denoising and can add fresh particles.

3 Persistent Particle Planning

3.1 Problem formulation

We consider a pretrained diffusion model over finite-horizon plans $x = (x_0, \dots, x_{H-1})$ with transitions $x_h = (a_h, o_h)$, where a_h is an action and o_h the predicted observation (e.g., positions and velocities). At control step k, the controller conditions generation on the current observation s_k, selects a plan, executes its first action, and replans after the next observation.

Constraints are supplied at inference time without retraining the diffusion model. Each constraint j contributes a residual $c_j(x) \geq 0$, evaluated in physical coordinates, which is the positive violation of an inequality or the absolute residual of an equality. The violation of a plan aggregates the residuals,

$$c(x) = A\big(c_1(x), \ldots, c_J(x)\big), \tag{1}$$

with any aggregate A that vanishes exactly when every residual does, such as their maximum or their sum; the residuals and aggregate used for each task are given in Appendices B and C.

3.2 Constraint-weighted diffusion sampling

A planning call maintains N noisy plans $\{y_t^{(i)}\}_{i=1}^N$. At each reverse iteration, the diffusion model predicts a clean plan

$$\hat{x}_t^{(i)} = D_\theta(y_t^{(i)}, t, s_k), \tag{2}$$

whose initial observation coordinates are pinned to s_k. We evaluate a constraint potential on each clean prediction,

$$\psi_t^{(i)} = \exp\Big(-\beta_t \lambda_t c(\hat{x}_t^{(i)})\Big), \tag{3}$$

whose strength $\beta_t \lambda_t$ increases as denoising progresses (Appendix B). The potentials accumulate in the particle weights,

$$\widetilde{w}_t^{(i)} = w_{\text{in}}^{(i)} \psi_t^{(i)}, \qquad W_t^{(i)} = \frac{\widetilde{w}_t^{(i)}}{\sum_j \widetilde{w}_t^{(j)}}, \tag{4}$$

where $w_{\text{in}}^{(i)}$ is the weight entering the iteration. When the effective sample size $1/\sum_i (W_t^{(i)})^2$ falls below $N/2$, the population is resampled according to W_t and the weights are reset to uniform; otherwise the normalized weights are retained. The particles then propagate through the learned reverse diffusion transition.

This procedure favors lineages whose predicted clean trajectories satisfy the constraints during denoising. Because the proposal is the original diffusion transition, no proposal-density correction is required. Accumulating intermediate potentials defines a path-dependent selection objective that is not equivalent to weighting only the final clean trajectory (Appendix A.2, paragraph "Current-plan marginal").

3.3 Reusing the population across control steps

The first planning call initializes particles independently from Gaussian noise and runs the complete reverse process. After executing an action, the controller advances every retained plan by one transition:

$$S(x) = \big(x_1, \ldots, x_{H-1}, x_{H-1}\big). \tag{5}$$

Duplicating the final transition restores the planning horizon. The next denoising call repairs this extended tail together with the rest of the trajectory.

Each shifted plan is partially re-noised:

$$y_{t_0}^{(i)} = \sqrt{\bar{\alpha}_{t_0}}\, S(x^{(i)}) + \sigma_{\text{init}} \sqrt{1 - \bar{\alpha}_{t_0}}\, \epsilon^{(i)}, \qquad \epsilon^{(i)} \sim \mathcal{N}(0, I). \tag{6}$$

Here $x^{(i)}$ is the final plan of particle i in the previous call, $\bar{\alpha}_t$ is the noise schedule's cumulative signal coefficient, which decreases from near 1 at $t = 0$ toward 0 at the noisiest level, and σ_{init} is the noise scale of the checkpoint's sampler (Appendix B). The first observation of each re-noised plan is then replaced by the new measurement s_k, as in every denoising step, and denoising continues from level t_0.

The re-noising level t_0 controls both trajectory reuse and computation. A small value retains more of the previous plan and requires fewer denoising iterations, whereas a large value perturbs the plan more and allows more iterations for adaptation. A warm call runs $t = t_0, \ldots, 0$, that is, $t_0 + 1$ of the K reverse iterations, on the original potential schedule. Particle weights are retained across replanning boundaries in the main configuration, so both the candidate trajectories and their selection history persist; resetting them is evaluated as an ablation.

After denoising, the controller executes the first action of the minimum-violation plan, $i^\star = \arg\min_i c(x^{(i)})$, and retains the entire population for the next control step. Algorithm 1 in Appendix B gives the full procedure.

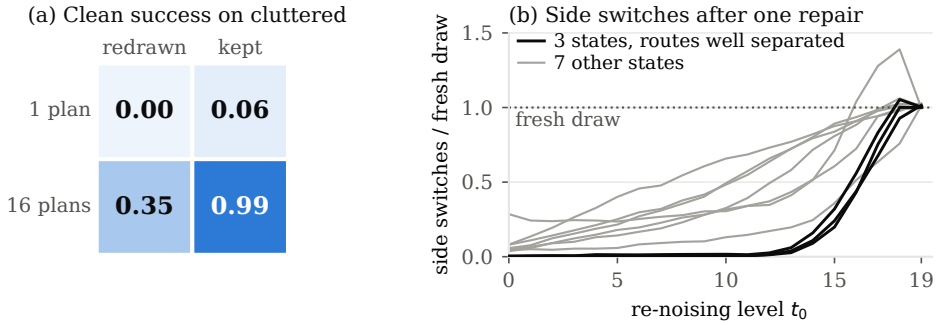


Figure 2: (a) Clean success on the cluttered configuration of Section 4, with one or 16 plans, redrawn or kept (Table 9). (b) Side switches after one learned repair, relative to a fresh draw, at ten fixed Avoiding states where fresh plans end on both sides of the centerline; black: the three states whose two routes are well separated.

3.4 Theoretical analysis

Particle target. Fix the observations and index denoising and shift-and-re-noise events by $r = 0, \ldots, R$. Write Y_r for the noisy plan at event r, with diffusion level $t(r)$, and $Y_{0:R}$ for its history. Y_0 is the initial Gaussian-noise plan with the first observation imposed. Let $\mathbb{P}$ be the law of this history without constraint weighting or resampling. With $\Lambda_t = \beta_t \lambda_t$, the path target is

$$\Gamma_R(dY_{0:R}) = \mathbb{P}(dY_{0:R}) \prod_{r=0}^{R} e^{-\Lambda_{t(r)} c(\hat{x}_{t(r)}(Y_r))}, \qquad \Pi_R = \Gamma_R / \Gamma_R(1). \tag{7}$$

P3 is the SMC particle system for this model; carried weights retain a lineage's selection history across calls (Proposition 2, Appendix A.2). With $N = 1$, constraint weighting cannot change the executed plan.

On cluttered, one of the obstacle-avoidance configurations of Section 4, the fraction of trials that reach the goal without violations is 0.35 with 16 fresh plans, 0.06 with one retained plan, and 0.99 with 16 retained plans (Figure 2(a)). Under an idealized, exact repair (assumption (E) below), adding more noise before denoising lowers the probability that a retained plan remains on its route. Proofs and diagnostics are in Appendix A.

Imagine drawing a route around an obstacle and then partly erasing it. If enough remains to tell which side it takes, filling it back in tends to preserve that choice. As more is erased, reconstruction becomes more like choosing a route from scratch. Theorem 1 shows that adding more noise makes repair less likely to preserve the original side and more likely to switch sides.

Exact repair (E). At a fixed observation, let p be the model's plan law. Re-noise $x \sim p$ without shifting, then draw $x' \sim p(\cdot \mid y)$ independently of x given y. Rescaling eq. (6) gives $y = \gamma_{t_0} x + \epsilon$, $\epsilon \sim \mathcal{N}(0, I)$, where $\gamma_t = \sigma_{\text{init}}^{-1} \sqrt{\bar{\alpha}_t / (1 - \bar{\alpha}_t)}$ decreases with t.

Theorem 1 (Retention under re-noising)**.** *Under (E), for any measurable A with $\nu = p(A) \in (0, 1)$, write $V_A = \mathrm{Var}[p(A \mid y)]$. (i) The retention and boundary-crossing probabilities are*

$$\Pr(x' \in A \mid x \in A) = \nu + \frac{V_A}{\nu}, \qquad \Pr(\mathbf{1}_A(x') \neq \mathbf{1}_A(x)) = 2\nu(1 - \nu) - 2V_A. \tag{8}$$

(ii) V_A decreases strictly with t_0 and tends to zero as $\gamma_{t_0} \to 0$, when the repair becomes a fresh draw from p.

The covariance identity in (i) is classical data augmentation covariance identity (Liu et al., 1994). Here, V_A measures how much the conditional route probability $p(A \mid y)$ varies across noisy plans; retention exceeds the fresh-draw probability by V_A / ν. At the three fixed states with well-separated routes, repair changes the plan's endpoint side at most 3% as often as independent redrawing for $t_0 \leq 12$. At $t_0 = 19$, the switch rates are approximately equal (Figure 2(b); Appendix A.5).

Corollary 1 (Rare-route retention). *Under (E), let p mix $\mathcal{N}(\mu_a, s^2 I)$ and $\mathcal{N}(\mu_b, s^2 I)$ with weights $0 < \pi \leq 1/2$ and $1 - \pi$. Here, s is the standard deviation in each plan coordinate within either component, assumed equal for the two components. If the original plan is sampled from component a, the repaired plan also belongs to component a with probability at least $1/2$ if and only if*

$$\kappa := \frac{\|\mu_a - \mu_b\|}{\sqrt{s^2 + \gamma_{t_0}^{-2}}} \geq \sqrt{2 \ln((1-\pi)/\pi)}. \tag{9}$$

Finding at least one plan from component a with probability at least $1/2$ requires $N \geq \ln 2 / \ln(1/(1-\pi))$ independent fresh draws.

A rarely chosen route can still be easy to recognize after adding noise if it is sufficiently different from the alternatives. Repair can then preserve it even though a fresh draw would be unlikely to find it. At $\pi = 0.01$, the $1/2$-retention threshold is $\kappa \approx 3.03$, whereas finding at least one plan from that component with the same probability takes 69 independent fresh draws. These are mixture labels; geometric route regions agree with them when the components are well separated.

Independent plans (S). Condition on the pre-planning history $\mathcal{G}_k$, which fixes the current observation and held plans. Final plans are independent. Each held plan is repaired with independent randomness, and fresh plans are drawn independently from the unweighted plan law. There is no within-call weighting or resampling.

Proposition 1 (Kept and fresh plans). *Let $A_k = \{z : c_k(z) \leq \delta\}$, $\delta \geq 0$, so the minimum-violation plan is acceptable iff the final population contains one. Under (S), m repairs of held plans z_i have acceptance probabilities $r_k(z_i)$, and n fresh draws have rate ν_k. Then:*

(i) The probability of no acceptable plan is

$$(1 - \nu_k)^n \prod_{i=1}^{m} (1 - r_k(z_i)). \tag{10}$$

Thus n fresh draws restore an acceptable plan with probability at least $1 - (1 - \nu_k)^n$, whatever is held.

(ii) For N fresh plans with mean marginal acceptance ν_k, the probability of finding one is at most $N\nu_k$, without independence; for independent draws with a common rate it is $1 - (1 - \nu_k)^N$.

(iii) Under (E), with ν, V_A as in Theorem 1, if the m held plans are i.i.d. from $p(\cdot \mid A)$, with independent repairs and n independent fresh draws from p, the one-step failure probability is $(1 - \nu - V_A/\nu)^m (1 - \nu)^n \leq (1 - \nu)^{m+n}$.

Under the assumptions in part (iii), keeping and repairing acceptable plans is at least as likely to leave us with an acceptable plan as drawing a fresh population of the same size.

3.5 Rejuvenation

Under (S), n new plans can restore an acceptable plan whatever is held (Proposition 1(i)). This motivates rejuvenation. At each boundary, P3 can replace its n lowest-weight particles with new plans, sampled from noise down to level $t_0 + 1$ by the same weighted sampler. They enter at t_0 with the population's mean weight and cost $n(K - 1 - t_0)$ extra evaluations per step. Rejuvenation is a heuristic outside our analysis. Its new plans replace weight-selected particles and are not independent draws (Appendix A).

4 Experiments

Section 4.2 compares P3 with the baselines and reports planning time. Section 4.3 separates keeping plans from selecting among them and reports ablations.

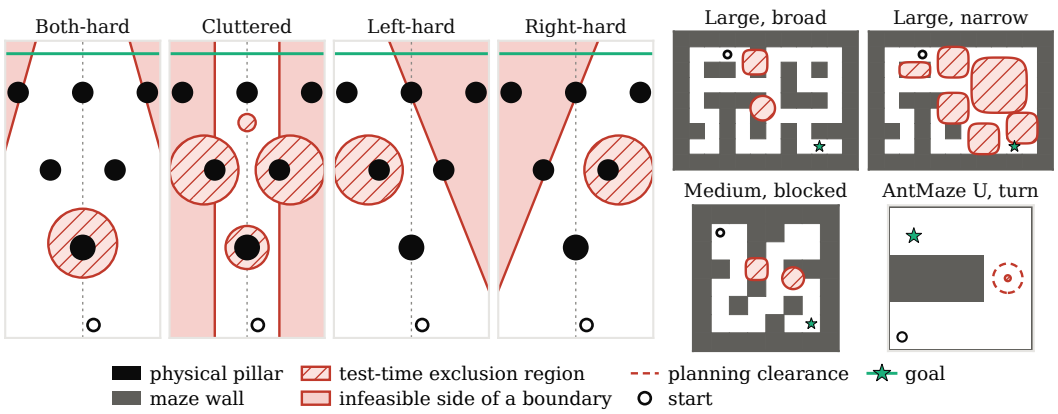


Figure 3: Tasks and test-time constraints. Left: D3IL Avoiding with its four added configurations; the dotted line is the centerline of the plan-side switch metric. Right: the large PointMaze with its broad and narrow presets (top), the medium PointMaze with our blocked preset, whose only clean route is the detour, and AntMaze U with an exclusion disc at the turn (bottom). None of the added constraints appears in the training data.

4.1 Setup

We evaluate on four tasks (Figure 3). **Avoiding** is the D3IL obstacle-avoidance task (Jia et al., 2024), in which a robot end effector crosses a field of six pillars to a goal line. The model plans over desired-position increments, desired positions and measured positions. Four configurations add static constraints (Figure 3). *Both-hard*, *left-hard* and *right-hard* were introduced by Römer et al. (2025), and *cluttered*, a narrow corridor with four exclusion regions, by Jung et al. (2025). **PointMaze** (de Lazcano et al., 2023; Fu et al., 2020) moves a force-actuated point mass between fixed start and goal cells, and the model plans over forces, positions and velocities. On the large layout, the *broad* and *narrow* obstacle presets clip the corridors of the unguided route; on the medium layout, *blocked* closes the direct corridor and leaves a detour that the demonstrations contain but the unguided model rarely executes. **AntMaze U** adds an exclusion disc at the turn of the U-shaped D4RL AntMaze (Fu et al., 2020), planned with a goal-conditioned state diffusion model and an inverse-dynamics head. Appendix C gives the shared protocol (horizons, re-noising depths, margins and trial counts), and Appendices F, G and H the maze layouts, models and presets.

Baselines. *Unguided* draws one cold sample without constraint handling, and *Projection* projects one onto the constraints during denoising, as in DPCC (Römer et al., 2025). *RDM + projection* keeps one plan and repairs or regenerates it as RDM (Zhou et al., 2023) does, projecting it onto the constraints (Appendix B). RDM as published does not use the added constraints. *DiRecT* (Giaretta et al., 2026) solves, in the second half of denoising, a constrained optimization on the model's clean-plan estimate and corrects the noisy plan by the result (Appendix B). *SafeDiffuser* (Xiao et al., 2025) applies a control-barrier QP to the planned positions at every denoising step (Appendix B). *Best-of-N* executes the minimum-violation plan among N cold samples, and *SMC* (cold SMC) is the sampler of Section 3.2 with a fresh population at every step; P3 adds trajectory and weight reuse to it, and rejuvenation on Avoiding. *Persistent best-of-N* is P3 without weighting or resampling (Appendix E.4). Settings per task are in Appendix C.

Metrics. *Clean success* is the fraction of trials that reach the goal without any violation of the added constraints, graded on the unpadded constraints; task success and the accumulated violation are in Appendix D. *Plan switches* count how often the selected plan's endpoint changes side of the workspace centerline on Avoiding, a route-consistency proxy rather than a homotopy test. Means and standard deviations are across model seeds (evaluation seeds on AntMaze); bold marks the best mean.

Table 1: Clean success under test-time constraints, mean ± standard deviation across five model seeds (AntMaze: six evaluation seeds). Avoiding columns: both-hard, cluttered, left-hard, right-hard. Upper rows plan one sample per step, lower rows a population. Trial counts and settings: Appendix C; additional metrics: Appendix D.

	Avoiding				Large maze		Medium	AntMaze
Method	Both	Clutt.	Left	Right	Broad	Narrow	Blocked	Turn
Unguided	0.00 ± 0.01	0.00 ± 0.00	0.09 ± 0.07	0.05 ± 0.08	0.06 ± 0.10	0.02 ± 0.05	0.00 ± 0.00	0.38 ± 0.11
RDM + projection	0.60 ± 0.11	0.09 ± 0.05	0.60 ± 0.10	0.36 ± 0.14	0.50 ± 0.11	0.72 ± 0.10	$\mathbf{0.24} \pm 0.09$	0.37 ± 0.15
Projection	0.85 ± 0.05	0.46 ± 0.26	0.83 ± 0.05	0.79 ± 0.10	0.66 ± 0.22	0.91 ± 0.15	0.11 ± 0.09	0.40 ± 0.04
DiRecT	0.92 ± 0.06	0.28 ± 0.12	0.84 ± 0.03	0.88 ± 0.10	$\mathbf{1.00} \pm 0.00$	$\mathbf{0.98} \pm 0.04$	0.12 ± 0.11	0.48 ± 0.09
SafeDiffuser	0.44 ± 0.17	0.16 ± 0.12	0.66 ± 0.31	0.34 ± 0.30	0.30 ± 0.36	0.04 ± 0.08	0.00 ± 0.01	0.39 ± 0.14
Best-of-N	0.92 ± 0.06	0.34 ± 0.35	0.88 ± 0.07	0.85 ± 0.19	0.74 ± 0.29	0.51 ± 0.38	0.01 ± 0.01	0.43 ± 0.11
SMC	0.97 ± 0.03	0.48 ± 0.41	0.91 ± 0.06	0.97 ± 0.04	0.75 ± 0.36	0.51 ± 0.39	0.00 ± 0.00	0.53 ± 0.11
Persistent best-of-N	0.93 ± 0.03	0.87 ± 0.14	0.86 ± 0.09	0.92 ± 0.13	0.75 ± 0.18	0.56 ± 0.36	0.20 ± 0.07	0.37 ± 0.06
P3 (ours)	$\mathbf{0.99} \pm 0.01$	$\mathbf{0.99} \pm 0.01$	$\mathbf{0.94} \pm 0.04$	$\mathbf{0.98} \pm 0.04$	0.89 ± 0.09	0.85 ± 0.21	0.21 ± 0.13	$\mathbf{0.62} \pm 0.10$

4.2 Main results

P3 attains the highest mean clean success in five of the eight columns of Table 1. Its gains over populations regenerated at every step are largest on cluttered, narrow and the medium maze, where a clean route is hard to find from scratch.

Avoiding. On cluttered, clean success is 0.99 for P3 and 0.48 for SMC. Single-plan correction methods reach only 0.46 (Projection), 0.28 (DiRecT) and 0.16 (SafeDiffuser) there. An 8-step plan does not span its exclusion regions, so the correction can put consecutive plans on different sides of a region. Projection switches plan side 12.4 times per episode there, against 7.7 for P3 (Appendix D, Table 7 and Figure 6). On both-hard, left-hard and right-hard, P3 and SMC with the same N and margin differ in mean clean success by at most 0.03.

At equal margins, P3 is cleaner than Projection on all four configurations, and a larger margin helps Projection only on both-hard (Appendix D, Table 6). RDM + projection is below Projection on all four Avoiding cases. On cluttered, the projection step fails more often when RDM tries to fix its old plan than when it starts with a new one (Appendix B).

Mazes. On narrow, P3 is clean on 0.85 of trials, against about half for fresh populations. Projection reaches 0.91 there. On the medium maze, RDM + projection, which repairs its projected plan at every step, P3 and persistent best-of-N carry their plans across control steps and complete the detour cleanly most often (0.24, 0.21 and 0.20, against at most 0.12). With $N = 4$, P3 has the highest clean success on AntMaze (0.62, against 0.53 for SMC and 0.43 for best-of-N). P3 enters the disc in 29% of episodes, compared with 38% for SMC. With $N = 32$ the two are close (0.61 and 0.60; Appendix H).

Planning time. On one RTX 4070 Ti GPU, all particles of a population are denoised in one batch, so a planning call takes about as long as sampling one plan with the same number of denoising iterations (Appendix D.1). A warm P3 call runs only $t_0 + 1$ of the K iterations. P3 takes 65, 93, 24 and 186 ms per control step on cluttered (without rejuvenation), narrow, the medium maze and AntMaze ($N = 32$), against 112, 118, 86 and 224 ms for SMC. On all four tasks it is also faster than sampling one unguided plan, and faster than Projection, DiRecT and SafeDiffuser. On narrow, for instance, DiRecT takes 467 ms and Projection 27 s per step.

4.3 Ablations

Table 1 runs best-of-N and SMC with and without persistence, which separates keeping plans from selecting among them. Keeping plans helps on its own. On cluttered, persistent best-of-N is clean on 0.87 of trials, against 0.34 for best-of-N. Selecting among kept plans adds to this. P3 reaches 0.99 on cluttered and 0.85 on narrow, against 0.87 and 0.56 for persistent best-of-N, and the two

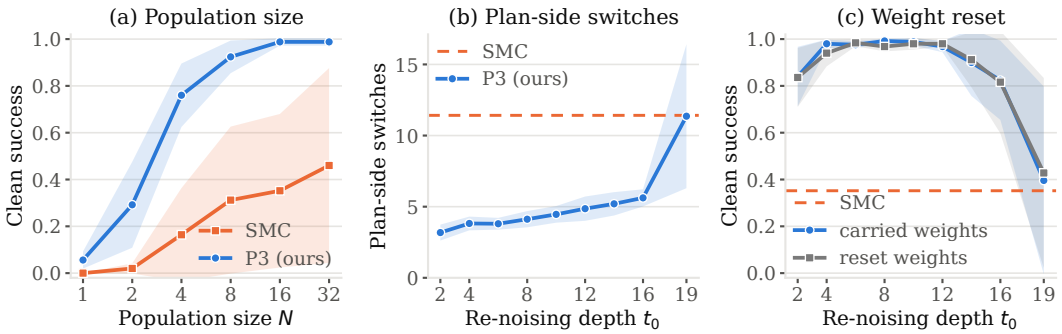


Figure 4: Ablations on cluttered, mean ± one standard deviation over five model seeds. (a) Clean success against the population size. (b) Plan-side switches and (c) clean success with carried or reset weights against the re-noising depth t_0, at $N = 16$. Dashed: SMC. Details: Appendix E.

are similar on the medium maze (0.21 and 0.20; Appendix E.4). At the Table 1 setting, persistence also lowers plan switching, by up to 2.5 times (Table 7).

Without rejuvenation, P3 is clean on 0.90 of cluttered trials, still far above SMC, but on both-hard it falls to 0.88, below SMC's 0.97 (Table 4). The ablations in Figure 4 run on cluttered with margin 0.01 and no rejuvenation. With 8 particles, P3 is cleaner than cold SMC with 32 (0.92 against 0.46) at about a seventh of the denoising cost (Table 9). Clean success is highest at intermediate depths (0.97–0.99 for $t_0 = 4$–12) and falls to 0.40 at $t_0 = 19$, where the re-noised input depends only weakly on the previous plan and switching returns to the cold-SMC level (Theorem 1). At that depth, with weights carried across steps or reset to uniform, clean success stays near cold SMC (0.40 and 0.43 against 0.35). Retaining weights alone does not reproduce the gains at intermediate depths (Appendix E.3).

5 Discussion and Limitations

Among population methods, persistence helps most when a clean route is rare under the model, as on cluttered and the narrow maze. A kept population can hold such a route once found, whereas a population regenerated at every step has to find it again. It adds little where clean routes are common and the population is large. On AntMaze the unguided model alone is clean on 0.38 of trials, and with $N = 32$ mean clean success is 0.61 for P3 and 0.60 for SMC. Each replan there shifts the kept plans by five steps at once (Appendix H). When a single plan already spans the obstacles, as on the large maze with 64-step plans, correcting that plan by constrained optimization also works well.

Our test-time constraints are static regions and boundaries on planar positions. The exact-repair and equal spherical Gaussian assumptions of Corollary 1 do not hold exactly for our learned planner, whose repair includes shifting and learned denoising and whose resampling creates shared ancestry (Appendix A.5). The corollary's threshold fails to predict retention at some higher noise levels, even at a well-separated Avoiding state, and we have not validated it for rare routes or on the mazes. Proposition 1(iii) compares one-step probabilities of finding an acceptable plan, averaged over the kept plans; it gives no pointwise, long-run, or executed-path safety guarantee.

6 Conclusion

P3 keeps a weighted population of diffusion plans across control steps. At every step, it shifts and partially re-noises the kept plans and selects among them by constraint weighting and resampling, without retraining. Under an idealized repair, we proved that the re-noising depth controls how reliably a kept plan stays on its route. Keeping a rare, well-separated route also takes far fewer plans than rediscovering it. In our experiments, keeping plans helped most where clean routes are rare, selecting among them added to this, and reuse made each replan cheaper. A diffusion planner need not start from scratch at every step. The plans it has already found are worth keeping.

Reproducibility Statement

Algorithm 1 and Appendix B specify P3 fully, including the potential schedule, the resampling rule, the noise-scale convention and action selection. Appendix B also describes our implementations of RDM + projection, DiRecT and SafeDiffuser; the latter two port the official code to our checkpoints. Appendix C gives, for every task, the horizon, number of denoising steps, population size, re-noising depth, margins, trial counts, episode limits and grading rule, and Appendices F, G and H describe the task-specific models, training data and constraint presets. Avoiding uses the public D3IL demonstrations and AntMaze the public D4RL `antmaze-umaze-v1` data. The PointMaze demonstrations come from the scripted waypoint controller described in Appendices F and G. Main results use five independently trained models per task, or one checkpoint with six blocks of evaluation seeds on AntMaze. Each trial's start state is fixed by its index, so all methods face the same trials. The planners' random draws differ across GPU models, so bit-exact reruns require the same device. Each theoretical result states its assumptions, and Appendix A gives complete proofs. Appendix D.1 reports the hardware and protocol of the timing measurements. Code, pretrained models and the scripts that produce the tables and figures are available at the anonymized repository linked in the abstract.

AI Use Statement

Generative AI tools were used in this work to give feedback on experimental design, assist in analyzing and interpreting experimental results, help develop parts of the theoretical framework, suggest some mathematical claims and key steps in their proofs, assist in writing proofs, implement and debug software, produce figures, search for and summarize related literature, format references, draft sections of the manuscript, and edit the text for clarity and structure.

Other theoretical ideas and mathematical claims were developed by the authors. We have not used generative AI tools to propose or refine research hypotheses, translate text, or generate, clean, or reformat datasets. Qualitative or thematic data analysis is not applicable to this work.

The authors reviewed and verified all AI-assisted technical content: mathematical proofs were checked line-by-line by hand, software and implementations were verified through test suites, and literature citations were checked against original sources. The authors take full responsibility for the final content of this work.

References

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations*, 2023.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. In *Advances in Neural Information Processing Systems*, 2025.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 24(2):265–272, 1953.

Gabriel Cardoso, Yazid Janati El Idrissi, Sylvain Le Corff, and Eric Moulines. Monte Carlo guided denoising diffusion models for Bayesian linear inverse problems. In *International Conference on Learning Representations*, 2024.

João Carvalho, An T. Le, Mark Baierl, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023.

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*, 2023.

Nicolas Chopin and Omiros Papaspiliopoulos. *An Introduction to Sequential Monte Carlo*. Springer, Cham, 2020.

Jacob K. Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Gabriele Corso, Yilun Xu, Valentin De Bortoli, Regina Barzilay, and Tommi Jaakkola. Particle guidance: Non-I.I.D. diverse sampling with diffusion models. In *International Conference on Learning Representations*, 2024.

Rodrigo de Lazcano, Kallinteris Andreas, Jun Jet Tai, Seungjae Ryan Lee, and Jordan Terry. Gymnasium robotics. http://github.com/Farama-Foundation/Gymnasium-Robotics, 2023.

Pierre Del Moral. *Feynman–Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, New York, 2004.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, 18(1):252–278, 2012.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Zehao Dou and Yang Song. Diffusion posterior sampling for linear inverse problem solving: A filtering perspective. In *International Conference on Learning Representations*, 2024.

Nic Fishman, Leo Klarner, Valentin De Bortoli, Emile Mathieu, and Michael Hutchinson. Diffusion models for constrained domains. *Transactions on Machine Learning Research*, 2023.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Mathieu Gerber, Nicolas Chopin, and Nick Whiteley. Negative association, ordering and convergence of resampling methods. *The Annals of Statistics*, 47(4):2236–2260, 2019.

Paolo Giaretta, Zeyang Li, and Navid Azizan. DiRecT: Safe diffusion-based planning via receding-horizon denoising. *arXiv preprint arXiv:2606.15359*, 2026.

Walter R. Gilks and Carlo Berzuini. Following a moving target—Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63 (1):127–146, 2001.

Neil J. Gordon, David J. Salmond, and Adrian F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2): 107–113, 1993.

Pieralberto Guarniero, Adam M. Johansen, and Anthony Lee. The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647, 2017.

Jeremy Heng, Adrian N. Bishop, George Deligiannidis, and Arnaud Doucet. Controlled sequential Monte Carlo. *The Annals of Statistics*, 48(5):2904–2929, 2020.

Sigmund H. Høeg, Yilun Du, and Olav Egeland. Streaming diffusion policy: Fast policy synthesis with variable noise diffusion models. In *IEEE International Conference on Robotics and Automation*, 2025.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, 2022.

Xiaogang Jia, Denis Blessing, Xinkai Jiang, Moritz Reuss, Atalay Donat, Rudolf Lioutikov, and Gerhard Neumann. Towards diverse behaviors: A benchmark for imitation learning with human demonstrations. In *International Conference on Learning Representations*, 2024.

Wonsuhk Jung, Utkarsh A. Mishra, Nadun Ranawaka Arachchige, Yongxin Chen, Danfei Xu, and Shreyas Kousik. Joint model-based model-free diffusion for planning with constraints. In *Conference on Robot Learning*, 2025.

Nikolas Kantas, Jan M. Maciejowski, and Andrea Lecchini-Visintini. Sequential Monte Carlo for model predictive control. In *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*. Springer, 2009.

Sunwoo Kim, Minkyu Kim, and Dongmin Park. Test-time alignment of diffusion models without reward over-optimization. In *International Conference on Learning Representations*, 2025.

Alexander Lambert, Adam Fishman, Dieter Fox, Byron Boots, and Fabio Ramos. Stein variational model predictive control. In *Conference on Robot Learning*, 2020.

Xiner Li, Yulai Zhao, Chenyu Wang, Gabriele Scalia, Gokcen Eraslan, Surag Nair, Tommaso Biancalani, Shuiwang Ji, Aviv Regev, Sergey Levine, and Masatoshi Uehara. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Jun S. Liu, Wing Hung Wong, and Augustine Kong. Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes. *Biometrika*, 81 (1):27–40, 1994.

Yuejiang Liu, Jubayer Ibn Hamid, Annie Xie, Yoonho Lee, Max Du, and Chelsea Finn. Bidirectional decoding: Improving action chunking via guided test-time sampling. In *International Conference on Learning Representations*, 2025.

Aaron Lou and Stefano Ermon. Reflected diffusion models. In *International Conference on Machine Learning*, 2023.

Nanye Ma, Shangyuan Tong, Haolin Jia, Hexiang Hu, Yu-Chuan Su, Mingda Zhang, Xuan Yang, Yandong Li, Tommi Jaakkola, Xuhui Jia, and Saining Xie. Inference-time scaling for diffusion models beyond scaling denoising steps. *arXiv preprint arXiv:2501.09732*, 2025.

David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

Kazuki Mizuta and Karen Leung. Unified generation-refinement planning: Bridging guided flow matching and sampling-based MPC for social navigation. *arXiv preprint arXiv:2508.01192*, 2025.

Chaoyi Pan, Zeji Yi, Guanya Shi, and Guannan Qu. Model-based diffusion for trajectory optimization. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Alexandre Piché, Valentin Thomas, Cyril Ibrahim, Yoshua Bengio, and Chris Pal. Probabilistic planning with sequential Monte Carlo methods. In *International Conference on Learning Representations*, 2019.

Ralf Römer, Alexander von Rohr, and Angela P. Schoellig. Diffusion predictive control with constraints. In *Learning for Dynamics and Control Conference*, 2025.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Raghav Singhal, Zachary Horvitz, Ryan Teehan, Mengye Ren, Zhou Yu, Kathleen McKeown, and Rajesh Ranganath. A general framework for inference-time scaling and steering of diffusion models. In *International Conference on Machine Learning*, 2025.

Marta Skreta, Tara Akhound-Sadegh, Viktor Ohanesian, Roberto Bondesan, Alán Aspuru-Guzik, Arnaud Doucet, Rob Brekelmans, Alexander Tong, and Kirill Neklyudov. Feynman–Kac correctors in diffusion: Annealing, guidance, and product of experts. In *International Conference on Machine Learning*, 2025.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

Brian L. Trippe, Jason Yim, Doug Tischer, David Baker, Tamara Broderick, Regina Barzilay, and Tommi Jaakkola. Diffusion probabilistic modeling of protein backbones in 3D for the motif-scaffolding problem. In *International Conference on Learning Representations*, 2023.

Weixin Wang, Yu Yang, Wei Deng, and Pan Xu. Inference-time alignment of diffusion models via trust-region iterative twisted sequential Monte Carlo. *arXiv preprint arXiv:2605.25123*, 2026.

Nick Whiteley and Anthony Lee. Twisted particle filters. *The Annals of Statistics*, 42(1):115–141, 2014.

Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M. Rehg, Byron Boots, and Evangelos A. Theodorou. Information theoretic MPC for model-based reinforcement learning. In *IEEE International Conference on Robotics and Automation*, 2017.

Luhuan Wu, Brian L. Trippe, Christian A. Naesseth, David M. Blei, and John P. Cunningham. Practical and asymptotically exact conditional sampling in diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. SafeDiffuser: Safe planning with diffusion probabilistic models. In *International Conference on Learning Representations*, 2025.

Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-ES: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential Monte Carlo. In *International Conference on Machine Learning*, 2024.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems*, 2023.

Siyuan Zhou, Yilun Du, Shun Zhang, Mengdi Xu, Yikang Shen, Wei Xiao, Dit-Yan Yeung, and Chuang Gan. Adaptive online replanning with diffusion models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A Proofs and implementation diagnostics

A.1 Setting and notation

Plans lie in $\mathbb{R}^{H\times T}$ with the Euclidean norm. $\mathrm{TV}(\mu,\nu)=\sup_A|\mu(A)-\nu(A)|$ is the total-variation distance between probability measures, and Φ is the standard normal cdf. C_k pins the observed coordinates at control step k (and goal coordinates on the mazes), and $\hat{x}_t(y)=C_k(\mathrm{clip}(D_\theta(y,t,s_k)))$, where clip clips PointMaze predictions to $[-1,1]$ and is the identity on Avoiding and AntMaze. Write $R_{k,t}$ for the pinned learned reverse kernel at $t\geq 1$, $F_k(y)=\hat{x}_0(y)$ for the terminal map, and $B_k(z,\cdot)$ for shift, re-noising and pinning in eq. (6). The unweighted warm repair is $\widetilde{M}_k = B_kR_{k,t_0}\cdots R_{k,1}F_k$; a fresh plan has law $p_k=\mu^{(k)}R_{k,K-1}\cdots R_{k,1}F_k$, where $\mu^{(k)}=\mathrm{Law}(C_k(\sigma_{\mathrm{init}}\epsilon))$. These kernels are specified by Algorithm 1.

Let $Y_0\sim\mu^{(0)}$ and let Y_r be the noisy-plan state at event r, as in eq. (7). Within a call, event r at level $t(r)$ has transition $Q_r=R_{k,t(r)+1}$; at a boundary it has $Q_r(y,\cdot)=B_k(F_{k-1}(y),\cdot)$. Thus $\mathbb{P}=\mu^{(0)}\prod_{r\geq 1}Q_r$ is the reference chain in eq. (7). We distinguish the potential score c^{pot} from

the selector score c^{sel} when their margins differ, and set $\psi_t(y) = e^{-\Lambda_t c^{\mathrm{pot}}(\hat{x}_t(y))}$. Finite predictions give $0 < \psi_t \leq 1$. Plain P3 means P3 without rejuvenation.

For (E), fix $p = p_k$ and write

$$\gamma_t = \sigma_{\mathrm{init}}^{-1}\sqrt{\frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}}, \qquad G_t(x,\cdot) = \mathcal{N}(\gamma_t x, I), \qquad M^\circ(x, dx') = \int G_{t_0}(x, dy)\, p(dx' \mid y). \tag{11}$$

The cosine schedule makes γ_t strictly decreasing; at $K = 20$ and $\sigma_{\mathrm{init}} = 0.5$, $\gamma_0 = 22.28$, $\gamma_{10} = 1.689$, and $\gamma_{19} = 0.0049$. The history $\mathcal{G}_k$ contains the observations, earlier populations and random draws, and any replaced slots before call k. Assumption (S) conditions on this history and removes within-call weighting and resampling. The Feynman–Kac identification fixes the observations externally; conditioning on an entire endogenous closed-loop observation sequence does not give the same law.

A.2 The particle target

Proposition 2 (Feynman–Kac representation)**.** *For fixed observations, plain P3 with carried weights is the SMC particle system with ESS-triggered systematic resampling for eq. (7). (a) At $N = 1$ it executes the unweighted warm sampler, whatever the constraints; at $N = 2$ constraints act only through the final choice between two unweighted lineages. Both statements hold pathwise in closed loop. (b) If a boundary kernel forgets its input, the target from that boundary onward is the restarted target. Given the pre-boundary history, carried weights leave the first-event unnormalized importance sum unbiased and multiply its variance by N/ESS relative to uniform weights.*

Proof. At event r, particles move by Q_r, multiply their entering weights by $\psi_{t(r)}$, and resample if $\mathrm{ESS} < N/2$. No proposal ratio is needed. Systematic resampling is unbiased: the interval of particle i, of length W_i, contains NW_i resampling points in expectation. These are the SMC operations for $\Gamma_R = \mu^{(0)}\psi_{t(0)}\prod_{r=1}^{R} Q_r\psi_{t(r)}$ (Del Moral, 2004; Chopin & Papaspiliopoulos, 2020). Execution takes the minimum-score member, not a draw from Π_R.

(a) For normalized nonnegative weights, $\mathrm{ESS} \geq 1$, so the strict trigger cannot fire at $N \leq 2$. Moves ignore the scores and execution ignores the weights. Couple runs with the same Gaussian draws and environment noise: at $N = 1$ induction over calls gives identical actions under every constraint configuration; at $N = 2$ it gives the planner that keeps two unweighted slots and selects their minimum-score member. The same argument applies to RDM as published, whose triggers never read the constraints (Appendix B).

(b) Let event b have $Q_b(y,\cdot) = \eta_k$. The path measure factorizes into the pre-boundary factor and $\eta_k(dY_b)\psi_{t_0}(Y_b)\prod_{r>b} Q_r\psi_{t(r)}$. Normalization cancels the first factor, giving the restarted target. Given $\mathcal{G}_k$, carried weights W_i are fixed and $Y_b^{(i)} \sim \eta_k$ are i.i.d. Hence, for bounded f and $\psi = \psi_{t_0}$,

$$\widehat{S}(f) = \sum_i W_i\psi(Y_b^{(i)})f(Y_b^{(i)}), \qquad \mathbb{E}[\widehat{S}(f) \mid \mathcal{G}_k] = \eta_k(\psi f), \qquad \mathrm{Var}(\widehat{S}(f) \mid \mathcal{G}_k) = \frac{\mathrm{Var}_{\eta_k}(\psi f)}{\mathrm{ESS}(W)}. \tag{12}$$

Uniform weights replace $\mathrm{ESS}(W)$ by N. The normalized ratio $\widehat{S}(f)/\widehat{S}(1)$ is generally biased; neither the unbiasedness nor the exact variance factor is claimed for it or for later events. □

Current-plan marginal. Let R_k end call k, $x^k = F_k(Y_{R_k})$, and $\mathbb{P}_k = \mathrm{Law}_{\mathbb{P}}(x^k)$. The current-plan marginal is $\pi_k(dx) \propto \Psi_k(x)\mathbb{P}_k(dx)$, with $\Psi_k(x) = \mathbb{E}_{\mathbb{P}}[\prod_{r\leq R_k}\psi_{t(r)}(Y_r) \mid x^k = x]$. Thus P3 reweights its reference law, but need not preserve its conditional law on feasible plans: their intermediate predictions can have different violations. Terminal-only weighting gives $\pi_k(dx) \propto e^{-\Lambda_0 c^{\mathrm{pot}}(x)}\mathbb{P}_k(dx)$; telescoped potentials give that target as well (Wu et al., 2023; Singhal et al., 2025).

Weight resetting and forgetting. Resetting discards the accumulated lineage weights: up to common normalization, each is the product of potentials since its most recent resampling, or initialization if it has never been resampled. Resetting weights can help or hurt. The implemented boundary

never forgets exactly, but for two plans z, z' the Gaussian total-variation formula and contraction under Markov kernels give

$$\mathrm{TV}(B_k(z,\cdot), B_k(z',\cdot)) \le 2\Phi\big(\tfrac{1}{2}\gamma_{t_0}\|Sz - Sz'\|\big) - 1.$$

Thus dependence on any fixed pair of shifted plans vanishes as $\gamma_{t_0} \to 0$. This is a one-boundary statement, not an episode bound.

Scope. For fixed observations and bounded violations, standard consistency results apply with multinomial resampling, including adaptive resampling under the conditions of Del Moral et al. (2012). Stratified resampling at every event is also consistent, whereas systematic resampling can fail depending on particle order (Gerber et al., 2019); we claim no convergence theorem for our slot order or closed-loop feedback. Rejuvenation also lies outside Proposition 2: newcomers replace weight-selected slots and receive assigned weights.

A.3 Proofs of the retention results

Under (E), with p_Y the noisy marginal, the joint law is

$$p(dx)G_{t_0}(x, dy)p(dx' \mid y) = p_Y(dy)p(dx \mid y)p(dx' \mid y). \tag{13}$$

Thus (x, x') is exchangeable and both marginals are p. For any $f \in L^2(p)$, conditional independence gives the data-augmentation identity (Liu et al., 1994)

$$\mathrm{Cov}(f(x), f(x')) = \mathrm{Var}(\mathbb{E}[f(x) \mid y]). \tag{14}$$

Proof of Theorem 1. (i) Put $g(y) = p(A \mid y)$. Since $\mathbb{E}g = \nu$, $\Pr(x, x' \in A) = \mathbb{E}[g^2] = \nu^2 + V_A$. Dividing by ν gives retention; exchangeability gives crossing probability $2(\nu - \nu^2 - V_A)$. The Gaussian likelihood is positive, so $0 < g < 1$ and $V_A < \nu(1-\nu)$.

(ii) Write $\gamma = \gamma_{t_0}$ and consider $0 < \gamma' < \gamma$. Couple $z = x + \gamma^{-1}\epsilon$ and $z' = z + c\epsilon'$, where $c^2 = \gamma'^{-2} - \gamma^{-2}$ and the noises are independent. The noisier channel is a garbling of the first (Blackwell, 1953). For $U = \mathbb{E}[f(x) \mid z]$, total variance gives

$$\mathrm{Var}(U) = \mathrm{Var}(\mathbb{E}[f(x) \mid z']) + \mathbb{E}[\mathrm{Var}(U \mid z')].$$

If equality holds between the first two terms, then $U = g_1(z) = g_2(z + c\epsilon')$ almost surely. The conditional Gaussian has an everywhere-positive density, so Fubini's theorem forces g_2, and hence U, to be constant. It follows that the convolution of the finite signed measure $(f - \mathbb{E}f)\,dp$ with $\mathcal{N}(0, \gamma^{-2}I)$ vanishes. Its Fourier transform is multiplied by $e^{-\|\omega\|^2/(2\gamma^2)} \neq 0$, so uniqueness of Fourier transforms implies $f = \mathbb{E}f$ p-almost surely. For $f = \mathbf{1}_A$ this contradicts $\nu \in (0,1)$. Therefore V_A increases strictly with γ, and decreases strictly with t_0.

For the fresh limit, total variation contracts under the posterior kernel $y \mapsto p(\cdot \mid y)$. Writing Φ for the standard normal cdf, convexity and the equal-covariance Gaussian formula give

$$\mathrm{TV}(M^\circ(x,\cdot), p) \le \mathrm{TV}(G_{t_0}(x,\cdot), p_Y) \le \int \left[2\Phi\left(\frac{\gamma\|x - \tilde{x}\|}{2}\right) - 1\right] p(d\tilde{x}). \tag{15}$$

The integrand is bounded by one and tends to zero with γ, so dominated convergence gives convergence to p for every x. A second application, using $V_A = \int_A (M^\circ(x, A) - \nu)p(dx)$, gives $V_A \to 0$. □

Proof of Corollary 1. Let $L \in \{a, b\}$ be the mixture label and draw (L', x') jointly from the posterior given y. This realizes (E), with L' the repaired plan's component. Given L, the rescaled observation has law $\mathcal{N}(\gamma\mu_L, \bar{\sigma}^2 I)$, $\bar{\sigma}^2 = 1 + \gamma^2 s^2$. Bayes' rule gives

$$p(L = a \mid y) = \mathrm{sig}\left(\mathrm{logit}\,\pi + \frac{\|y - \gamma\mu_b\|^2 - \|y - \gamma\mu_a\|^2}{2\bar{\sigma}^2}\right), \qquad \mathrm{sig}(u) = \frac{1}{1 + e^{-u}}.$$

Under $L = a$, the likelihood-ratio term is $\kappa^2/2 + \kappa\xi$ with $\xi \sim \mathcal{N}(0,1)$. Consequently

$$\Pr(L' = a \mid L = a) = \mathbb{E}\big[\mathrm{sig}(\mathrm{logit}\,\pi + \kappa^2/2 + \kappa\xi)\big]. \tag{16}$$

The function $u \mapsto \mathbb{E}[\mathrm{sig}(u + \kappa\xi)]$ is strictly increasing and equals $1/2$ at $u = 0$ by symmetry. Retention is therefore at least $1/2$ iff $\mathrm{logit}\,\pi + \kappa^2/2 \ge 0$, proving the threshold. For independent fresh draws, discovery probability is $1 - (1-\pi)^N$, giving the stated count. At $\pi = 0.01$ the thresholds are 3.0315 and 68.97, hence 69 draws. For $s > 0$, $\kappa < \|\mu_a - \mu_b\|/s$, so reducing the noise cannot overcome insufficient intrinsic route separation. □

Mixture components and route regions. Let $D = \mu_a - \mu_b \neq 0$ and $A = \{x : \langle x - (\mu_a + \mu_b)/2, D\rangle > 0\}$. For $s > 0$, the probability that $\mathbf{1}_A(x)$ disagrees with $\mathbf{1}_{\{L=a\}}$ is $e = \Phi(-\|D\|/(2s))$. By stationarity the same is true after repair, so $|\Pr(x, x' \in A) - \Pr(L = L' = a)| \leq 2e$. Thus a geometric interpretation requires small overlap; for conditional retention of a rare route, the error must also be small relative to its mass.

Linear-feature diagnostic. For a unit vector u with $s_u^2 = \mathrm{Var}(u^\top x) \in (0, \infty)$, eq. (14) implies

$$\mathrm{corr}(u^\top x, u^\top x') = 1 - \frac{\mathbb{E}[(u^\top x - \mathbb{E}[u^\top x \mid y])^2]}{s_u^2} \geq \frac{s_u^2 \gamma^2}{1 + s_u^2 \gamma^2}.$$

Indeed, the conditional mean minimizes squared error; restricting to affine predictors from $u^\top y = \gamma u^\top x + u^\top \epsilon$ gives error $s_u^2/(1 + s_u^2\gamma^2)$. Figure 5(a) compares the measured correlations with this lower bound. It does not bound switching for non-Gaussian plans; the indicator identity of Theorem 1 does.

A.4 Kept and fresh plans

Here $c_k = c^{\mathrm{sel}}$ and $A_k = \{c_k \leq \delta\}$. On the mazes $\delta = 0$ is admissible. On Avoiding the score includes consistency equalities that do not vanish exactly; $0 < \delta \leq m_{\mathrm{v}}$, the virtual padding, ensures acceptance implies satisfaction of the unpadded geometric rows. An acceptable plan need not produce a clean executed step because of tracking error, and a clean episode need not have an acceptable full-horizon plan at every step.

Proof of Proposition 1. (i) Given $\mathcal{G}_k$, the independent failure events have probabilities $1 - r_k(z_i)$ and $1 - \nu_k$; multiplication gives eq. (10). The minimum-score plan lies in A_k iff any member does. The fresh-plan recovery clause follows by considering only the n fresh draws; their mutual independence suffices, regardless of the held plans. If every $r_k(z_i) \geq r$, the failure probability is at most $(1 - r)^m(1 - \nu_k)^n$.

(ii) The union bound gives $\Pr(\bigcup_{i=1}^N \{X_i \in A_k\} \mid \mathcal{G}_k) \leq \sum_i \Pr(X_i \in A_k \mid \mathcal{G}_k) = N\nu_k$. Under independence with a common rate it equals $1 - (1 - \nu_k)^N$. The bounds condition only on the history before this call and therefore hold in closed loop. For a success probability of at least $1/2$ the independent sampler needs $N \geq \ln 2 / \ln(1/(1 - \nu_k))$. For cold SMC, ν_k in the union bound is its mean final-particle rate, which can depend on N and differs from the unweighted model's rate.

(iii) At a fixed observation, integrate over independent held plans $z_i \sim p(\cdot \mid A)$ and independent repair noise. Theorem 1(i) gives each repair acceptance $a = \nu + V_A/\nu \geq \nu$. Together with independent fresh draws this gives $(1 - a)^m(1 - \nu)^n \leq (1 - \nu)^{m+n}$. □

Best-of-N satisfies (S) exactly with $(m, n) = (0, N)$. Plain P3 uses $(N, 0)$ and rejuvenation $(N - n, n)$ only under (S). In the implementation, resampling both changes the marginal laws and creates shared ancestors. Newcomers are generated by a weighted cold sub-chain and enter the combined population before further selection; they are not n independent final draws. Thus the recovery clause motivates rejuvenation but is not a bound for Table 1's implementation. The one-step comparison also requires the specific distribution $p(\cdot \mid A)$, not an arbitrary selection of acceptable plans.

A.5 Implementation diagnostics

Departures from exact repair. Even an exact posterior-mean network would not implement (E): a learned DDPM step draws one Gaussian around its predicted mean, whereas the exact reverse transition mixes these Gaussians over $p(x \mid y_t)$; the terminal step returns a prediction rather than a posterior draw. The network is also trained at unit noise scale but sampled at $\sigma_{\mathrm{init}} = 0.5$. Shifting and the changing observation add further departures in closed loop. The shallow-depth failure in Table 10 is consistent with retaining an infeasible continuation, but does not identify the cause or locate an optimal depth from the theory alone.

Repair at fixed states. We use 17 Avoiding states from all five model seeds, including ten two-sided decision states chosen before the measurement by a pilot side fraction in $[0.2, 0.8]$. At each state we draw 4096 cold plans, re-noise each without shifting, and denoise under the same observation for every $t_0 = 0, \ldots, 19$. Side is the sign of the final lateral position relative to the centerline. We record switching relative to $2\pi(1-\pi)$, where π is the cold side fraction, and the repaired side fraction π'. A second draw uses independent noise. For correlation, the first half of the cold sample fits two principal directions and the second half evaluates them; the maze diagnostic uses three states per maze and one direction. Routes are designated well separated when the distance between side means divided by their within-side spread is at least six, a threshold fixed before the run.

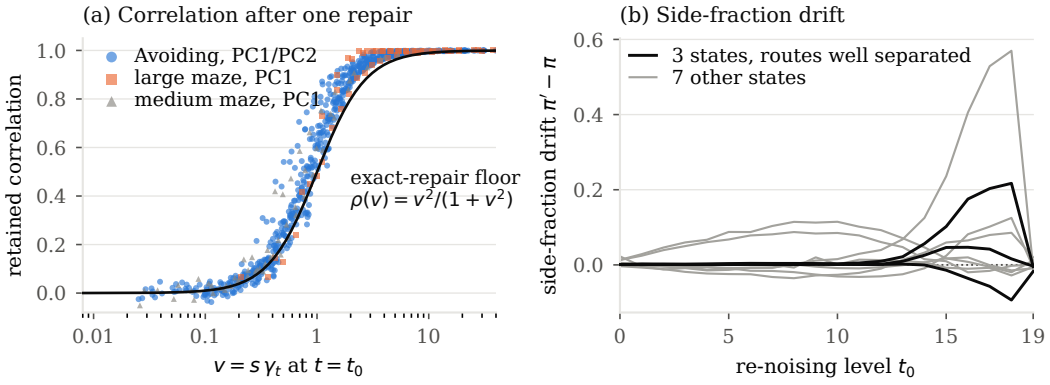


Figure 5: Learned repair at fixed states. (a) Correlation along held-out plan directions against the exact-repair floor derived in Appendix A.3. (b) Side-fraction drift at the ten decision states; (E) predicts zero. Black: the three states whose two routes are well separated.

Figure 2(b) shows the main draw's switch ratios. At the three separated-route states, ratios stay at or below 0.03 through $t_0 = 12$; at $t_0 = 19$ all ten ratios are in $[0.98, 1.03]$. Figure 5(b) shows that learned repair is not stationary: side-fraction drift reaches 0.115 at $t_0 \leq 12$ and 0.57 at $t_0 = 18$. The switch rate reaches 1.39 times the fresh-draw value. Measured correlations are consistent with the exact-repair lower bound at moderate depths, but some fall below it at larger depths. The side-fraction changes show that learned repair does not preserve the plan distribution, as (E) would require.

We also estimate κ in Corollary 1 from the side means and pooled within-side spread along their difference. At $t_0 = 10$, all ten decision states pass the estimated threshold and have minority-side retention above $1/2$ in both draws. The threshold is not reliable at larger depths. At one well-separated state, $t_0 = 17$ gives $\kappa = 1.77$ against a threshold of 1.57, but measured retention is 0.38 and 0.41 in the two draws. These moment-based estimates do not establish a two-Gaussian fit; the tested minority masses are about 0.20–0.46, not 0.01.

Acceptable plans at visited states. We log cold SMC and P3 at $N = 16$, without rejuvenation: on cluttered, 10 trials each for two model seeds and one contrast seed, $t_0 = 10$ and $\delta = 0.01$; on the narrow large maze, five trials each for two model seeds, $H = 64$, $t_0 = 16$ and $\delta = 0$. Avoiding reuses the ablation trials; the maze diagnostics use separate trials. At every visited state, 64 extra unweighted cold plans estimate ν_k independently of the planner's samples and are never executed. Four unweighted repairs of each previously acceptable held plan estimate bare repair retention. We also record the acceptable fraction among its final descendants after weighting and resampling (zero if none survive). In Table 2, $\hat{q}_k^{\mathrm{desc}}$ denotes this descendant fraction and $\hat{r}_k^{\mathrm{rep}}$ the bare-repair fraction; neither is the exact kernel probability in Proposition 1.

On the first two cluttered seeds, cold SMC loses every acceptable plan in all 20 logged episodes, at lapse states with median estimated fresh rate 0.016. At P3's own states with estimated $\nu_k < 1/N$, it holds an acceptable plan on 0.82 and 0.61 of steps, against fresh estimates 0.13 and 0.22. The table's retention columns summarize all steps. On the separate subset of hard states with estimated $\nu_k < 1/N$, median bare-repair retention is only 0.05–0.40 across the five settings, below the corre-

Table 2: Find/keep diagnostics. Lost all: cold-SMC episodes with no acceptable plan at some step; ν_k there: median estimated fresh rate at such lapses. Retention columns report medians. Held: fraction of P3 steps with an acceptable plan among states with estimated $\nu_k < 1/N$; fresh: mean plug-in estimate $1 - (1 - \nu_k)^N$ at those same states.

		clean		cold SMC		P3, all steps		P3, $\nu_k < 1/N$	
Task	Seed	cold	P3	lost all	ν_k there	$\hat{q}_k^{\mathrm{desc}}$	$\hat{r}_k^{\mathrm{rep}}$	held	fresh
Cluttered	1	0/10	10/10	10/10	0.016	1.00	0.95	0.82	0.13
Cluttered	2	0/10	9/10	10/10	0.016	1.00	0.94	0.61	0.22
Cluttered	3	8/10	9/10	7/10	0.031	1.00	0.96	0.54	0.17
Narrow maze	1	0/5	2/5	5/5	0.000	1.00	1.00	0.27	0.05
Narrow maze	2	0/5	4/5	5/5	0.000	1.00	1.00	0.30	0.07

sponding post-resampling descendant fractions. Cold SMC likewise exceeds the independent-fresh estimate at its own hard states. Assumption (S) excludes this weighting and resampling. The fresh column is an estimate, not a bound for cold SMC, and a zero rate means no successes among 64 probes, not zero model mass.

A.6 Projection and reweighting

Let F be a closed, nonempty feasible set. An exact nearest-point projection of $x \notin F$ lies in ∂F: an interior nearest point could be moved slightly toward x while remaining feasible. Consequently, if $p(\partial F) = 0$, the projected law places at least $p(F^c)$ mass on a p-null boundary. A selection target $G(x)p(dx)/p(G)$, for measurable $G \geq 0$ with $0 < p(G) < \infty$, is instead absolutely continuous with respect to p. It equals $p(\cdot \mid F)$ for $G = \mathbf{1}_F$ when $p(F) > 0$, but an arbitrary path-dependent weight need not preserve feasible route shares (Appendix A.2). Zero clearance here is relative to the projected set; padding may give positive clearance from the original constraints, and equality rows can make all of F its boundary.

This distributional distinction does not predict the switching rate of the local solver used inside the learned denoising process.

B Implementation details

Algorithm 1 gives P3 for joint state–action plans with one executed action per call. PointMaze also pins the goal coordinates and clips clean predictions to $[-1, 1]$; Avoiding does not clip them. AntMaze uses unclipped state-only plans, obtains actions from inverse dynamics, and executes and shifts by five steps per call (Appendix H). Cold SMC runs the cold branch at every control step, best-of-N skips lines 14–20, and the weight-reset ablation adds the optional reset of line 8; rejuvenation (Section 3.5) is the optional step of line 9.

Constraint residuals. In the Avoiding experiments the residuals c_j of eq. (1), aggregated by their maximum, encode geometric exclusions (the physical pillars and the added test-time regions and boundaries), action bounds, and planning-model consistency between consecutive desired positions and the action, $x_{h+1}^{\mathrm{des}} = x_h^{\mathrm{des}} + a_h$. The geometric residuals are those of the violation metric of Section 4.1, evaluated on the measured-position channel with the margins of Appendix C, after the affine map from normalized to physical coordinates.

Potential schedule. The coefficients of the potential in eq. (3) follow

$$u_t = u_{\mathrm{start}} + (u_{\mathrm{end}} - u_{\mathrm{start}}) \left(1 - \frac{t}{K-1}\right)^{1.5}, \qquad u \in \{\lambda, \beta\}, \tag{17}$$

with $(\lambda_{\mathrm{start}}, \lambda_{\mathrm{end}}) = (0.5, 15)$ and $(\beta_{\mathrm{start}}, \beta_{\mathrm{end}}) = (0.1, 15)$, so that $\Lambda_t = \beta_t \lambda_t$ rises from 0.05 at $t = K - 1$ to 225 at $t = 0$. A warm call resumes this schedule at t_0 rather than compressing it into fewer iterations, so the potential applied at a given level is the same in cold and warm calls.

Algorithm 1 Persistent particle planning (P3). Every assignment applies to all particles $i = 1, \dots, N$.

Require: model D_θ with K reverse steps, residual c, strengths $\Lambda_t = \beta_t \lambda_t$, population size N, re-noising level t_0, noise scale σ_{init}; $\epsilon^{(i)}, \xi^{(i)}$ denote fresh $\mathcal{N}(0, I)$ draws

warm $\leftarrow$ FALSE
for control step $k = 0, 1, 2, \dots$ **do**
  observe s_k
  if warm = FALSE **then** ▷ cold start from noise
    $y^{(i)} \leftarrow \sigma_{\text{init}}\, \epsilon^{(i)}$; $\log w^{(i)} \leftarrow 0$; $T \leftarrow K - 1$
  else ▷ persistence: shift, re-noise, keep the weights
    $y^{(i)} \leftarrow \sqrt{\bar{\alpha}_{t_0}}\, S(x^{(i)}) + \sigma_{\text{init}} \sqrt{1 - \bar{\alpha}_{t_0}}\, \epsilon^{(i)}$; $T \leftarrow t_0$ ▷ eqs. (5) and (6)
    optional (weight-reset ablation): $\log w^{(i)} \leftarrow 0$
    optional (rejuvenation): replace the $y^{(i)}$ of the n lowest-weight particles by fresh plans, run by this procedure with n particles from $K - 1$ to $t_0 + 1$, and set their $\log w^{(i)}$ to the log of the population's mean weight
  end if
  pin the observed coordinates of $y^{(i)}$ to s_k
  for $t = T, T - 1, \dots, 0$ **do** ▷ denoising clock
    $\hat{x}^{(i)} \leftarrow D_\theta(y^{(i)}, t, s_k)$, pinned to s_k
    $\log w^{(i)} \leftarrow \log w^{(i)} - \Lambda_t\, c(\hat{x}^{(i)})$ ▷ eqs. (3) and (4)
    $W^{(i)} \leftarrow \exp(\log w^{(i)}) \big/ \sum_j \exp(\log w^{(j)})$
    if $1 \big/ \sum_i (W^{(i)})^2 < N/2$ **then** ▷ eq. (18)
      draw ancestors $a^{(i)}$ by systematic resampling from W
      $(y^{(i)}, \hat{x}^{(i)}) \leftarrow (y^{(a^{(i)})}, \hat{x}^{(a^{(i)})})$; $\log w^{(i)} \leftarrow 0$
    else
      $\log w^{(i)} \leftarrow \log W^{(i)}$
    end if
    if $t > 0$ **then**
      $y^{(i)} \leftarrow m_t(\hat{x}^{(i)}, y^{(i)}) + \sigma_{\text{init}} e^{\ell_t/2}\, \xi^{(i)}$, pinned to s_k ▷ reverse step
    else
      $x^{(i)} \leftarrow \hat{x}^{(i)}$ ▷ final plans
    end if
  end for
  $i^\star \leftarrow \arg\min_i c(x^{(i)})$, lowest index on ties ▷ selection
  execute the first action of $x^{(i^\star)}$; keep $\{x^{(i)}, \log w^{(i)}\}_{i=1}^N$; warm $\leftarrow$ TRUE
end for

Weights and resampling. The effective sample size

$$\text{ESS}_t = \frac{1}{\sum_i (W_t^{(i)})^2} \tag{18}$$

is computed after every update, and when $\text{ESS}_t < N/2$ ancestors are drawn by systematic resampling from W_t. Resampling copies the selected ancestors and resets weights to uniform; otherwise the normalized weights are retained. P3 carries the population and its weights into the next call, continuing the weight product of Proposition 2 across the boundary. The weight-reset ablation instead starts each call with uniform weights.

Indexing of warm calls. Levels are zero-based: a cold call runs $t = K - 1, \dots, 0$ and a warm call runs $t = t_0, \dots, 0$, so a warm call performs $t_0 + 1$ denoiser evaluations, and $t_0 = 10$ with $K = 20$ corresponds to 11 iterations. Re-noising uses the cumulative signal coefficient $\bar{\alpha}_{t_0}$ of the cosine schedule at index t_0.

Noise-scale convention. All checkpoints are sampled with half the DDPM noise scale: the chain is initialized at $\sigma_{\text{init}}\, \epsilon$ with $\epsilon \sim \mathcal{N}(0, I)$, and the reverse noise injected at level t has standard deviation $\sigma_{\text{init}} e^{\ell_t/2}$, where ℓ_t is the DDPM posterior log-variance and $\sigma_{\text{init}} = 0.5$. Equation (6) uses the same

sampling noise scale, $\sigma_{\text{init}} = 0.5$, rather than unit noise scale. Using $\sigma = 1$ at the same t_0 would reduce the squared signal-to-noise ratio $\gamma_{t_0}^2$ of eq. (11) by a factor of four. All compared methods use this sampling noise scale.

Action selection. The executed plan is the population member with minimal c after the final denoising step, with ties resolved by the lowest particle index.

RDM baseline. RDM (Zhou et al., 2023) executes one plan. After every environment step it writes the observed state into the plan and scores the partially executed plan by the KL terms of the model's variational bound at small noise levels. It regenerates the plan from noise every H steps or when the normalized score falls below l_s, repairs it by re-noising and denoising under the current state (dropping the executed transitions) when the score falls below $l_f > l_s$, and otherwise keeps executing it, with actions from inverse dynamics towards the plan's next state. Every run uses the unguided row's sampler, model seeds, trials and step caps. RDM leaves three details open: we normalize the score as $\ell = \min(1, L_{\text{gen}}/L)$, so that a fresh plan scores one; we use the two smallest nondegenerate KL levels of our chains; and we write states only into the plan, because the action channels of the Avoiding checkpoints are not consistent with their planned setpoints. As published, RDM does not see the added constraints: its paths do not depend on them, because they never change the dynamics its likelihood monitors.

RDM with constraints. We add constraint projection during repair and regeneration, and trigger replanning when a retained plan violates an added constraint. RDM's step "constrain the plan with the conditions", used to pin the current state, also applies Projection's projector and margins. A kept plan is checked against the constraints padded by the selection margin, following RDM's rule that an infeasible plan must be replanned. A violation triggers a regeneration. On cluttered, the projector's SLSQP failed on 71% of repairs and 15% of regenerations in the same episodes. We use the thresholds $(0.5, 0.98)$ with repair depth 9 on Avoiding, repair at every step with depth 5 on the medium maze, and $(0.9, 0.99)$ with depth 25 on AntMaze. On the large maze we use RDM's published thresholds and plan 10 trials per seed. The narrow result includes 50 completed trials.

DiRecT baseline. We port the official DiRecT code (Giaretta et al., 2026) to our checkpoints and run it with the unguided row's sampler, model seeds, trials and step caps, replanning at every control step (every five on AntMaze). In the last half of the denoising steps (10 of 20; 25 of 50 on AntMaze) it takes the reverse step, solves $x^\star = \arg\min_{x \in S} \frac{1}{2}\lambda\bar{\alpha}_{t-1}\beta_t^{-1}\|x - \hat{x}_0\|^2 + C(x)$ with IPOPT ($\lambda = 1$), and adds $\sqrt{\bar{\alpha}_{t-1}}(x^\star - \hat{x}_0)$ to the noisy plan; the last step returns $x^\star$. As in the official code, $\hat{x}_0$ is the clean estimate at the current plan on Avoiding and at the uncorrected step on the mazes, and S includes linear dynamics fitted by least squares to the model's training transitions (none on AntMaze, whose plans have no actions). On Avoiding, S holds the added constraints and the pillars padded by DiRecT's 0.025 m and $C = 0$; on cluttered this padding closes the corridor, so we use Projection's 0.01. On the mazes, S holds the obstacles with DiRecT's margin (0.4 added to the superellipse power sum) and actions within their data range, and C is the squared path length; on AntMaze, S is the planning set of the other rows. Constraints apply to every planned state but the current one. IPOPT did not converge on under 0.3% of solves (5.0% on AntMaze), where the last iterate is used, as in the official code.

SafeDiffuser baseline. We port the official SafeDiffuser code (Xiao et al., 2025) to our checkpoints and run it with the unguided row's sampler, model seeds, trials and step caps. At every denoising step each planned position moves by the step closest to the unguided update that satisfies $\nabla b \cdot u + b \geq 0$, with b the constraint's power sum minus 1.01 (the official robust term) or, for a boundary, its signed distance; the other channels follow the unguided update, and the QPs are solved with qpth as in the official code. Of its variants, RoS imposes this at every step, ReS relaxes it (penalty weight 100) except in the last steps, and TVS grows the barrier over the last steps; we map the official 256-step schedules to our chains by their fraction, and use RoS on both-hard and the medium maze and ReS elsewhere. As in its official evaluation, the executed action tracks the plan's next waypoint, $(p_1 - p) + (v_1 - v)$ on PointMaze; on Avoiding the tracked waypoints are the commanded setpoints, and on AntMaze the inverse-dynamics head acts on the plan. Avoiding and AntMaze, where SafeDiffuser defines no setting, use the other rows' padded constraints, on AntMaze only the added disc (the official code has no wall barriers); the mazes use the obstacles

with the robust term only. A waypoint QP that is infeasible, or that asks for a move wider than the normalized workspace (a vanishing barrier gradient inside a large power-4 obstacle), keeps the unguided update; this affected at most 4.9% of waypoint QPs (right-hard) and under 0.3% on the mazes. Relaxation and fallback updates do not guarantee barrier satisfaction. The position corrections also leave velocities and actions unchanged, so a tracked trajectory can violate constraints even when its planned positions do not.

C Experimental protocol

Models, horizons and re-noising. For each task and model seed, all methods share the same pretrained model. Cold calls use $K = 20$ reverse iterations, or $K = 50$ on AntMaze. The standard execution interval is one action, or five on AntMaze; RDM instead checks its likelihood after each action and replans adaptively (Appendix B). The planning horizon is $H = 8$ on Avoiding and the medium maze, $H = 32$ on AntMaze and $H = 64$ on the large maze. Population methods use $N = 16$ particles, $N = 32$ in the Avoiding columns and $N = 4$ in the AntMaze column of Table 1 ($N = 32$ on AntMaze: Appendix H). P3 re-noises to $t_0 = 10$ on Avoiding, 16 on the large maze, 4 on the medium maze and 40 of 50 on AntMaze; in Table 1 it rejuvenates $n = 8$ particles per step on Avoiding only.

Constraints and margins. Avoiding aggregates its residuals by their maximum and the mazes sum the obstacle penetration of the plan's positions. On Avoiding, potentials and selectors pad the added constraints by 0.01, or 0.04 for best-of-N, SMC and P3 in Table 1, and the pillars by 0.025; Projection pads its constraints by 0.025 (0.01 for the cluttered regions), with approximate dynamics rows that couple its corrections to the actions. On cluttered, two radius-0.08 discs have centers 0.20 apart; padding each by 0.025 closes the passage between them, so Projection uses 0.01 instead. On the mazes the sampling methods pad the obstacles by 0.01 on the large layout and by 0.3 on the medium layout. Projection pads them by the ball radius 0.1; on AntMaze every constraint-aware planner keeps a 1.0 clearance from the disc. On Avoiding and PointMaze, grading uses the unpadded constraints on the measured state before each planning call; the terminal post-action state is not included. AntMaze grades the recorded center path after every executed action, including the terminal state.

Trials. Avoiding and PointMaze use five model seeds and 50 trials per seed, method and configuration (100 for the Avoiding population rows of Table 1). Computational cost limits Projection on the large maze to 5–10 trials per seed. AntMaze uses six evaluation seeds of 20 trials, paired across methods on their environment seeds. Episodes last at most 200 control steps on Avoiding, 1200 on the large maze and 800 on the medium maze. Methods are paired by trial seed, which fixes each episode's start state.

Timing. The Avoiding ablation latencies were measured separately from the success-rate runs, with one process on an idle CPU, one model seed and ten trials per setting. Population-size and noise-depth sweeps use separate timing runs. Appendix D.1 compares all methods on one GPU.

D Full results

Tables 3 and 4 give task success and accumulated violation for the runs in Table 1. Tables 5 and 7 report plan switches and episode lengths.

Avoiding. Table 7 compares plan-side switching under Projection, SMC and P3. At the Table 1 setting, Projection's SLSQP solver fails on 1–9% of calls. Figure 6 shows executed paths on cluttered and on the medium maze.

Margin sensitivity. Table 6 varies each planner's Avoiding margin without changing its other settings or the unpadded grading. The base margin is 0.025 on both-hard, left-hard and right-hard, and 0.01 on cluttered. P3 at the base margin retains its advantage over Projection on cluttered and right-hard. On both-hard, increasing Projection's margin to 0.04 improves its clean success from

Table 3: Task success for the runs in Table 1, irrespective of constraint violations. Means and standard deviations are across model seeds (evaluation seeds on AntMaze); trial counts are as in Table 4.

	Avoiding				Large maze		Medium	AntMaze
Method	Both	Cluttered	Left	Right	Broad	Narrow	Blocked	Turn
Task success ↑								
Unguided	0.80 ± 0.11	0.79 ± 0.11	0.80 ± 0.11	0.79 ± 0.11	**1.00** ± 0.00	**1.00** ± 0.00	**0.76** ± 0.17	0.82 ± 0.08
RDM + projection	0.77 ± 0.10	0.85 ± 0.05	0.79 ± 0.10	0.63 ± 0.10	0.88 ± 0.12	0.98 ± 0.04	0.58 ± 0.10	0.84 ± 0.06
Projection	0.87 ± 0.05	0.97 ± 0.02	0.95 ± 0.02	0.88 ± 0.08	**1.00** ± 0.00	**1.00** ± 0.00	0.26 ± 0.13	0.83 ± 0.08
DiRecT	**1.00** ± 0.00	0.72 ± 0.12	0.88 ± 0.03	0.93 ± 0.07	**1.00** ± 0.00	**1.00** ± 0.00	0.12 ± 0.11	0.85 ± 0.09
SafeDiffuser	0.50 ± 0.21	0.53 ± 0.29	0.66 ± 0.32	0.44 ± 0.36	0.41 ± 0.43	0.52 ± 0.36	0.53 ± 0.10	0.78 ± 0.06
Best-of-N	0.99 ± 0.01	0.65 ± 0.25	**1.00** ± 0.00	0.95 ± 0.05	**1.00** ± 0.00	**1.00** ± 0.00	0.55 ± 0.11	0.85 ± 0.05
SMC	0.99 ± 0.00	0.82 ± 0.16	**1.00** ± 0.00	0.97 ± 0.04	**1.00** ± 0.00	**1.00** ± 0.00	**0.76** ± 0.13	0.86 ± 0.13
Persistent best-of-N	0.98 ± 0.03	0.96 ± 0.04	**1.00** ± 0.00	0.92 ± 0.13	**1.00** ± 0.00	**1.00** ± 0.00	0.35 ± 0.11	0.83 ± 0.09
P3 (ours)	**1.00** ± 0.00	**1.00** ± 0.00	**1.00** ± 0.00	**0.98** ± 0.04	**1.00** ± 0.00	**1.00** ± 0.00	0.45 ± 0.09	**0.89** ± 0.07

Table 4: Accumulated violation, mean ± standard deviation across five model seeds (clean success is in Table 1). Avoiding: 50 trials per seed (100 for the population rows); best-of-N, SMC and P3 use $N = 32$ and margin 0.04, P3 with eight fresh particles per step (Section 4.1); the last Avoiding block is P3 at the same setting without rejuvenation. Mazes: large maze ($H = 64$), medium maze ($H = 8$) and AntMaze U ($H = 32$), 50 trials per seed (AntMaze: six evaluation seeds of 20 paired trials; large-maze Projection: 5–10); population methods use $N = 16$, except on AntMaze ($N = 4$). Unguided, RDM + projection and Projection use one sample; persistent best-of-N uses P3's setting without weighting or resampling (Appendix E.4); RDM + projection uses 10 trials per seed on the large maze (Appendix B).

	Avoiding			
Method	Both-hard	Cluttered	Left-hard	Right-hard
Accumulated violation ↓				
Unguided	2.717 ± 0.788	2.463 ± 0.582	1.484 ± 2.506	6.992 ± 3.522
RDM + projection	0.295 ± 0.086	0.594 ± 0.274	0.646 ± 0.361	1.030 ± 0.987
Projection	0.035 ± 0.040	0.016 ± 0.015	0.025 ± 0.019	0.215 ± 0.188
DiRecT	**0.000** ± 0.000	0.021 ± 0.014	0.004 ± 0.001	0.003 ± 0.004
SafeDiffuser	0.009 ± 0.004	0.848 ± 1.637	**0.000** ± 0.000	4.633 ± 5.890
Best-of-N	0.001 ± 0.001	0.014 ± 0.013	0.004 ± 0.002	0.014 ± 0.027
SMC	**0.000** ± 0.001	0.013 ± 0.017	0.004 ± 0.004	**0.000** ± 0.000
Persistent best-of-N	0.001 ± 0.001	0.002 ± 0.002	0.006 ± 0.005	**0.000** ± 0.000
P3 (ours)	**0.000** ± 0.000	**0.000** ± 0.000	0.001 ± 0.001	**0.000** ± 0.000
P3 without rejuvenation				
Clean success ↑	0.88 ± 0.12	0.90 ± 0.10	0.94 ± 0.05	1.00 ± 0.00
Accumulated violation ↓	0.090 ± 0.118	0.055 ± 0.053	0.002 ± 0.001	0.000 ± 0.000

	Large maze		Medium maze	AntMaze U
Method	Broad	Narrow	Blocked	Turn
Accumulated violation ↓				
Unguided	5.684 ± 2.430	14.442 ± 5.039	18.396 ± 0.786	0.188 ± 0.097
RDM + projection	2.319 ± 2.273	1.530 ± 0.754	6.505 ± 1.336	0.177 ± 0.062
Projection	3.977 ± 3.019	1.447 ± 1.979	4.005 ± 1.992	0.148 ± 0.052
DiRecT	**0.000** ± 0.000	**0.001** ± 0.002	**0.000** ± 0.000	0.145 ± 0.035
SafeDiffuser	0.404 ± 0.468	12.062 ± 7.333	15.544 ± 1.378	0.174 ± 0.071
Best-of-N	0.389 ± 0.605	3.589 ± 3.935	14.999 ± 1.515	0.131 ± 0.026
SMC	0.528 ± 1.021	3.506 ± 4.444	18.272 ± 1.060	0.126 ± 0.024
Persistent best-of-N	0.169 ± 0.109	2.525 ± 2.994	4.229 ± 3.323	0.173 ± 0.042
P3 (ours)	0.031 ± 0.021	0.246 ± 0.454	7.092 ± 3.988	**0.091** ± 0.029

0.834 to 0.942, leaving a smaller gap to P3's 0.986. The same increase harms Projection on left-hard and right-hard, where its SLSQP fallback rate rises from 0.033 to 0.233 and from 0.040 to 0.339, respectively. Projection is not tested with the larger margin on cluttered because the padded

Table 5: Large maze: plan switches (changes of the selected plan's end cell) per episode and episode length for the runs of Table 4, mean ± standard deviation across five model seeds.

	Plan switches ↓		Episode length (steps) ↓	
Method	Broad	Narrow	Broad	Narrow
Unguided	72.6 ± 15.9	72.5 ± 15.9	$\mathbf{529} \pm 39$	528 ± 38
Projection	58.6 ± 17.2	37.8 ± 10.5	535 ± 36	432 ± 60
Best-of-N	68.3 ± 18.6	44.9 ± 11.0	586 ± 39	426 ± 5
SMC	69.1 ± 21.8	43.2 ± 13.8	615 ± 50	421 ± 16
P3 (ours)	$\mathbf{44.5} \pm 11.7$	$\mathbf{28.2} \pm 4.8$	620 ± 84	$\mathbf{408} \pm 6$

Table 6: Avoiding margin sensitivity: clean success, mean ± standard deviation across five model seeds and 100 trials per seed. Base margin is 0.025, except 0.01 on cluttered. P3 uses $N = 32$ and eight fresh particles per step throughout.

	Projection		P3	
Case	Base	0.04	Base	0.04
Both-hard	0.834 ± 0.063	0.942 ± 0.044	0.944 ± 0.036	0.986 ± 0.015
Cluttered	0.446 ± 0.270	n/a	0.980 ± 0.009	0.994 ± 0.008
Left-hard	0.824 ± 0.112	0.364 ± 0.240	0.892 ± 0.103	0.936 ± 0.037
Right-hard	0.796 ± 0.095	0.108 ± 0.135	0.966 ± 0.043	0.978 ± 0.044

discs block its narrow passage (Appendix C). P3 can use 0.04 there because its margin enters a soft potential and selector, not a hard projection.

Persistence at matched settings. Table 7 compares P3 with SMC at the same N and margin. P3 switches plan side less often in every configuration, by up to 2.5 times. Accumulated violations are reported in Table 4. Figure 7 shows single episodes: on both-hard SMC keeps changing its planned continuation near the upper obstacles, and the plotted P3 episode on cluttered has smaller realization errors. Realization error is the distance between a planned endpoint and the position reached $H - 1$ steps later.

Large maze. DiRecT has the lowest mean penetration on both presets (Table 4). Among the population methods, P3 has the lowest mean penetration, including 0.031 on broad against 0.389 for best-of-N, and the fewest endpoint-cell switches. On narrow, its mean penetration is less than one tenth that of the fresh populations. Appendix F gives further large-maze details.

AntMaze. At $N = 4$, P3 enters the disc less often than SMC and Projection (29% of episodes, against 38% and 48%), has the lowest mean penetration and reaches the goal as often as SMC (0.89 and 0.86; Tables 3 and 4).

D.1 Planning time

Table 8 times every method of Table 1 on one configuration per task. All cells ran one after another on an otherwise idle machine with one RTX 4070 Ti GPU and an i7-13700F CPU. The diffusion model runs on the GPU and denoises all particles of a population in one batch. The solvers of Projection, DiRecT and SafeDiffuser run on the CPU. Each cell runs three episodes of 50 control steps with one model seed (two episodes of ten steps for Projection and RDM + projection on narrow), and the first step of every episode, which includes the cold start, is not timed. On AntMaze, a control step is one replan of five environment steps. RDM + projection keeps its plan on most steps and repairs or regenerates it on the others, so its step time varies widely. On narrow it kept its plan at every timed step.

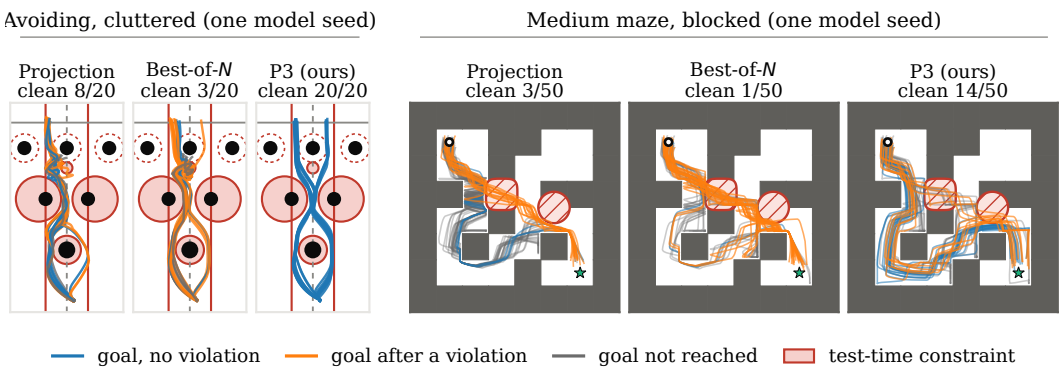


Figure 6: Executed paths on cluttered (twenty trials per method on the model seed whose Projection clean success is closest to its five-seed mean) and on the medium maze (the 50 trials per method of one model seed, within 0.07 of the five-seed mean clean success for every method). Cluttered uses $N = 16$, $t_0 = 10$, potential margin 0.01 and no rejuvenation, rather than the Table 1 setting; the medium-maze runs use that table's setting. The plots show side changes around the exclusion regions on cluttered (left) and trajectories through or around the blocked corridor on the medium maze (right).

Table 7: Projection, SMC and P3 on Avoiding at their Table 1 settings. SMC and P3 use $N = 32$, margin 0.04 and 100 trials per seed; Projection uses one plan and 50 trials per seed. Entries are means ± standard deviations across five model seeds.

Constraint configuration	Projection	SMC	P3 (ours)
Plan-side switches per episode ↓			
Both-hard	5.34 ± 2.70	4.49 ± 1.56	**1.79** ± 0.90
Cluttered	12.42 ± 2.56	10.63 ± 2.92	**7.72** ± 0.91
Left-hard	7.28 ± 3.36	2.87 ± 1.02	**2.35** ± 0.62
Right-hard	11.10 ± 4.09	3.31 ± 0.95	**2.63** ± 0.46

E Ablations

The population-size, re-noising-depth and weight-retention sweeps use cluttered, five model seeds, 50 trials per seed, at most 200 control steps, the unchanged learned transition, accumulated weighting with ESS resampling, and final selection by minimum violation. Along each swept axis the remaining settings are held at $N = 16$ particles, re-noising level $t_0 = 10$, weights carried across control steps, virtual potential margin 0.01, and no rejuvenation. Entries are means ± standard deviations across the five seeds. Appendix E.4 separately compares persistence with and without SMC across all tasks, with its own trial counts.

E.1 Population size

We vary N for fresh and retained populations under the same constraint potential. At $N = 1$, clean success is 0.00 for cold SMC and 0.06 for P3. Proposition 2(a) shows that, with one particle, constraint weighting cannot change the executed plan. With two particles the effective sample size never falls below $N/2$, so resampling never fires and the constraints act only through the final choice between two unweighted lineages. P3 at $N = 8$ reaches 0.92 clean success against 0.46 for cold SMC at $N = 32$, using $8 \times 11 = 88$ denoiser evaluations per warm replanning call, compared with $32 \times 20 = 640$ for cold SMC at $N = 32$, and takes 60 rather than 134 ms in the serial CPU measurement. The reduction is 7.3-fold in denoiser evaluations and 2.2-fold in measured latency. Proposition 1 compares finding acceptable plans with retaining them under independent final plans. Its $N\nu$ bound also holds for dependent plans, but ν must then be their mean final-particle acceptance rate, not the unweighted model's rate. The $N = 1$ result concerns this unchanged learned transition

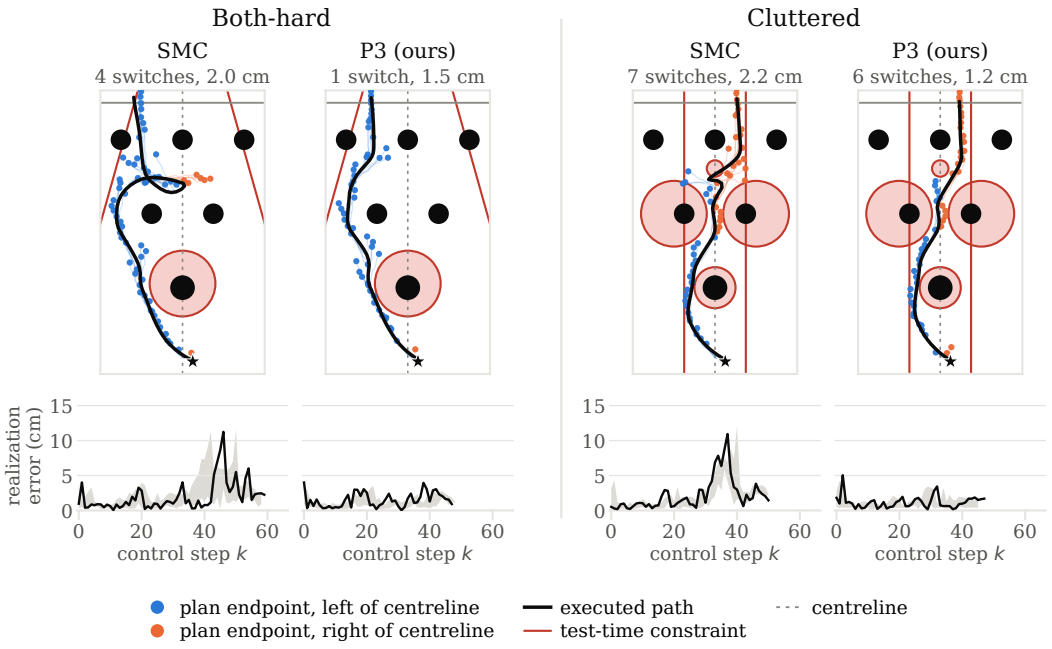


Figure 7: One example episode of SMC and P3 on both-hard and cluttered. These runs use $N = 16$, $t_0 = 10$, potential margin 0.01 and no rejuvenation, with eight episodes per method and configuration. Top: the executed path, with each selected plan's endpoint coloured by its side of the centerline, so a colour change is a plan-side switch. Bottom: realization error, the distance between each plan's endpoint and the position reached $H-1$ steps later (black), with the 10–90% band of the other seven episodes of the same method and configuration.

Table 8: Wall-clock time per control step in milliseconds, median and mean over the timed steps, on one RTX 4070 Ti. Each method uses its Table 1 setting, except that the AntMaze population rows are timed at $N = 32$. P3 rejuvenates only on Avoiding, and the last row times it there without rejuvenation.

	Cluttered		Narrow		Medium maze		AntMaze	
Method	Median	Mean	Median	Mean	Median	Mean	Median	Mean
Unguided	121	120	107	108	86	86	213	213
RDM + projection	4.8	41	5.5	6.4	44	51	653	621
Projection	142	149	27 238	26 143	101	101	214	236
DiRecT	100	101	467	688	139	138	399	2 048
SafeDiffuser	235	236	328	331	210	210	493	486
Best-of-N	120	121	109	109	82	82	200	203
SMC	112	112	118	118	86	87	224	225
Persistent best-of-N	108	109	89	90	22	22	171	174
P3 (ours)	117	118	93	93	24	24	186	187
P3 without rejuvenation	65	65	–	–	–	–	–	–

and does not imply that all single-trajectory warm-start methods behave the same way. The sweep shows no consistent trend in switching with N relative to the between-seed variability.

E.2 Re-noising depth

Clean success is highest at intermediate depths, 0.97–0.99 for $t_0 = 4$–12, compared with 0.84 at $t_0 = 2$ and 0.40 at $t_0 = 19$. The shallowest setting also has the largest accumulated violation

Table 9: Population size on cluttered constraints, at matched N for SMC and P3 ($t_0 = 10$). Violation is the accumulated residual along the executed path and switches are plan-side changes per episode; lower is preferable for both. Evals is N times the number of reverse iterations: per planning call for SMC and per warm replanning call for P3. P3's first call uses the full 20 iterations. s/step is wall-clock measured under the protocol in Appendix C, in a separate run from Table 10. Denoiser evaluations grow linearly in N, whereas measured latency changes little between $N = 2$ and $N = 16$.

	Task succ. ↑	Clean succ. ↑	Violation ↓	Switches ↓	Evals ↓	s/step
SMC						
$N = 1$	0.75 ± 0.11	0.00 ± 0.00	2.3805 ± 0.5500	7.46 ± 3.34	20	0.060
$N = 2$	0.62 ± 0.07	0.02 ± 0.02	0.8752 ± 0.2505	11.22 ± 4.76	40	0.117
$N = 4$	0.77 ± 0.18	0.16 ± 0.20	0.1440 ± 0.1234	12.92 ± 4.03	80	0.114
$N = 8$	0.87 ± 0.10	0.31 ± 0.31	0.0445 ± 0.0400	12.88 ± 4.71	160	0.105
$N = 16$	0.90 ± 0.08	0.35 ± 0.33	0.0323 ± 0.0288	11.42 ± 5.21	320	0.115
$N = 32$	0.97 ± 0.02	0.46 ± 0.42	0.0212 ± 0.0200	8.68 ± 3.09	640	0.134
P3 (ours), $t_0 = 10$						
$N = 1$	0.65 ± 0.18	0.06 ± 0.04	1.5026 ± 0.9990	3.08 ± 1.20	11	0.034
$N = 2$	0.69 ± 0.19	0.29 ± 0.18	0.6528 ± 0.7916	4.64 ± 1.32	22	0.067
$N = 4$	0.94 ± 0.06	0.76 ± 0.14	0.0413 ± 0.0203	4.56 ± 0.45	44	0.064
$N = 8$	0.99 ± 0.02	0.92 ± 0.07	0.0002 ± 0.0002	4.52 ± 0.58	88	0.060
$N = 16$	1.00 ± 0.01	0.99 ± 0.02	0.0035 ± 0.0070	4.46 ± 0.59	176	0.067
$N = 32$	1.00 ± 0.00	0.99 ± 0.02	0.0002 ± 0.0002	4.24 ± 0.92	352	0.076

Table 10: Re-noising depth on cluttered constraints at $N = 16$ with carried weights. The SMC row is the cold-start control. $t_0 = 19 = K-1$ re-noises to the start of the reverse chain. The accumulated weights are retained, but the re-noised input depends only weakly on the previous trajectory. Iter. counts reverse diffusion iterations per warm replanning call for P3 and per planning call for SMC; P3's first call uses all 20 iterations. Latency uses the separate serial CPU protocol of Appendix C. Cold SMC and $t_0 = 19$ have the same number of denoising iterations and latencies within 1%.

	Task succ. ↑	Clean succ. ↑	Violation ↓	Switches ↓	Iter.	s/step
SMC	0.90 ± 0.08	0.35 ± 0.33	0.0323 ± 0.0288	11.42 ± 5.21	20	0.114
$t_0 = 2$	0.96 ± 0.03	0.84 ± 0.13	0.1201 ± 0.1122	3.18 ± 0.56	3	0.020
$t_0 = 4$	0.99 ± 0.01	0.98 ± 0.02	0.0106 ± 0.0211	3.82 ± 0.48	5	0.031
$t_0 = 6$	1.00 ± 0.00	0.98 ± 0.01	0.0001 ± 0.0001	3.80 ± 0.41	7	0.043
$t_0 = 8$	1.00 ± 0.00	0.99 ± 0.01	0.0000 ± 0.0000	4.12 ± 0.57	9	0.053
$t_0 = 10$	1.00 ± 0.01	0.99 ± 0.02	0.0035 ± 0.0070	4.46 ± 0.59	11	0.063
$t_0 = 12$	1.00 ± 0.01	0.97 ± 0.02	0.0001 ± 0.0002	4.86 ± 0.85	13	0.075
$t_0 = 14$	0.99 ± 0.02	0.90 ± 0.14	0.0009 ± 0.0015	5.20 ± 0.82	15	0.085
$t_0 = 16$	0.98 ± 0.02	0.82 ± 0.17	0.0022 ± 0.0027	5.62 ± 0.60	17	0.098
$t_0 = 19$	0.96 ± 0.03	0.40 ± 0.40	0.0305 ± 0.0294	11.36 ± 5.06	20	0.113

among the persistent configurations. At $t_0 = 19$, the re-noised input depends only weakly on the retained plan (Appendix A.2). Plan-side switching increases overall, from 3.18 at $t_0 = 2$ to 11.36 at $t_0 = 19$, against 11.42 for cold SMC. This agrees qualitatively with the rise toward the fresh-draw rate in Theorem 1; the theorem does not predict the closed-loop values because the implemented repair is inexact (Appendix A.5). At $t_0 = 6$ and $t_0 = 10$, both within the high-success range, serial latency on one idle CPU is 43 and 63 ms per step, compared with 114 ms for cold SMC.

E.3 Weight retention

Table 11 reports reset weights, to compare with the carried weights in Table 10. At $t_0 = 10$, carrying versus resetting weights gives mean clean success of 0.99 versus 0.98. Carrying weights reduces accumulated violation at shallow depths, including 0.1201 versus 0.2293 at $t_0 = 2$. Carried weights also reduce plan-side switching slightly at every depth below $t_0 = 19$.

Table 11: Resetting weights to uniform at every control step, at $N = 16$ on cluttered. The carried-weight comparison is in Table 10. Both variants retain trajectories and differ only in the weights entering the next step.

	Clean succ. ↑	Violation ↓	Switches ↓
$t_0 = 2$	0.84 ± 0.13	0.2293 ± 0.2035	3.56 ± 0.61
$t_0 = 4$	0.94 ± 0.06	0.0559 ± 0.0888	4.04 ± 0.27
$t_0 = 6$	0.98 ± 0.01	0.0079 ± 0.0158	4.12 ± 0.31
$t_0 = 8$	0.97 ± 0.02	0.0002 ± 0.0002	4.58 ± 0.48
$t_0 = 10$	0.98 ± 0.03	0.0037 ± 0.0072	4.76 ± 0.62
$t_0 = 12$	0.98 ± 0.01	0.0034 ± 0.0066	4.90 ± 0.52
$t_0 = 14$	0.91 ± 0.12	0.0009 ± 0.0013	5.38 ± 0.56
$t_0 = 16$	0.82 ± 0.22	0.0028 ± 0.0040	5.66 ± 0.70
$t_0 = 19$	0.43 ± 0.41	0.0295 ± 0.0269	10.94 ± 5.23

Table 12: Persistent best-of-N against P3 at the ablation setting ($N = 16$, $t_0 = 10$, margin 0.01, no rejuvenation), five model seeds with 100 trials each.

	Method	Both-hard	Cluttered	Left-hard	Right-hard
Clean	Persistent best-of-N	0.54 ± 0.30	0.87 ± 0.18	0.46 ± 0.12	0.77 ± 0.18
	P3	0.78 ± 0.13	0.99 ± 0.01	0.81 ± 0.14	0.95 ± 0.05
Violation	Persistent best-of-N	0.6503 ± 0.8046	0.0009 ± 0.0012	0.0245 ± 0.0113	0.0128 ± 0.0216
	P3	0.2855 ± 0.3524	0.0045 ± 0.0088	0.0065 ± 0.0059	0.0004 ± 0.0005
Switches	Persistent best-of-N	1.48 ± 0.53	4.80 ± 0.70	3.55 ± 1.41	4.54 ± 1.12
	P3	1.39 ± 0.44	4.44 ± 0.54	3.68 ± 1.03	4.45 ± 0.59

Carried weights without trajectory memory. At $t_0 = 19$, the re-noising reaches the start of the reverse chain, where $\gamma_{t_0} \approx 0.005$. The re-noised input therefore depends only weakly on the previous trajectory, while the accumulated weights are retained. Cold SMC, carried weights and reset weights have mean clean success 0.35, 0.40 and 0.43, respectively, and similar observed switch rates. These clean-success rates are below the 0.97–0.99 observed at $t_0 = 4$–12: retaining weights at $t_0 = 19$ does not reproduce those intermediate-depth results. The near-equality of the three switch rates is consistent with the fresh-draw limit of Theorem 1(ii).

E.4 Persistence without SMC weighting and resampling

Resetting the carried weights leaves the weighting and resampling inside every planning call in place, so it cannot show whether this selection still matters once plans are kept. Persistent best-of-N removes it: at every control step all N plans are shifted and re-noised as in Section 3.3, denoised with the unweighted reverse kernel, and the plan with the smallest violation is executed. Each column runs it at that column's P3 setting: on Avoiding $N = 32$, $t_0 = 10$, potential margin 0.04 and eight fresh draws per step, which replace randomly chosen plans because persistent best-of-N has no weights to rank them by; on the large maze $N = 16$ and $t_0 = 16$; on the medium maze $N = 16$, $t_0 = 4$ and margin 0.3; on AntMaze $N = 4$ and $t_0 = 40$ of $K = 50$, shifted by the five executed steps. Every comparison pairs methods by trial seed, with the same initial environment state.

Tables 1, 3 and 4 report clean success, task success and violation for P3 and persistent best-of-N, which uses neither SMC weighting nor resampling. On cluttered, persistent best-of-N is clean on 0.87 of trials, compared with 0.34 for fresh best-of-N and 0.48 for cold SMC. On the medium maze, persistent best-of-N and P3 have mean clean success 0.20 and 0.21; fresh populations stay at or below 0.01. P3 has higher observed clean success than persistent best-of-N on the other tasks, with the largest large-maze difference on narrow (0.85 against 0.56). Without rejuvenation and at the smaller population of the ablations (Table 12), the clean-success difference is larger on both-hard, left-hard and right-hard. On both-hard and left-hard, persistent best-of-N reaches 0.54 and 0.46, compared with 0.78 and 0.81 for P3. Its observed plan-side switch rates remain close to P3's.

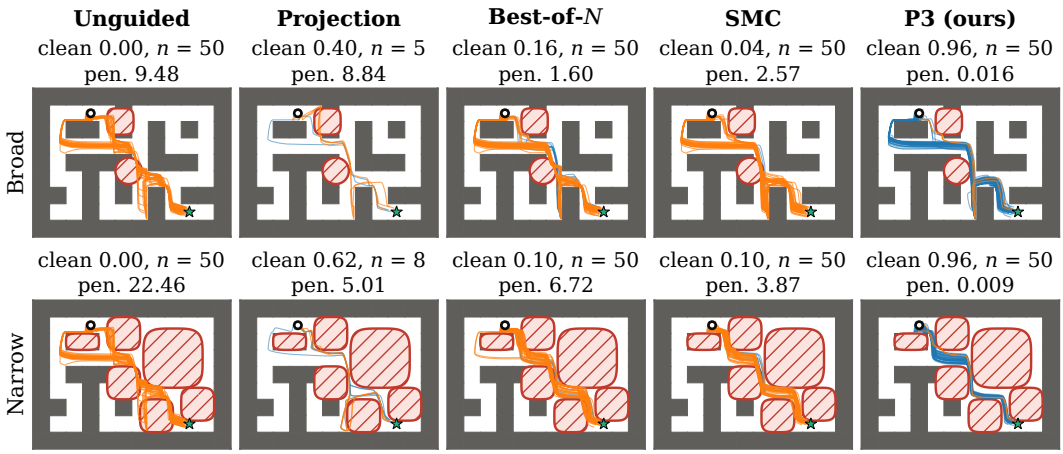


Figure 8: Executed paths on the large maze for one model seed, broad (top) and narrow (bottom). Blue: goal reached cleanly; orange: after a violation. Titles give clean success, mean penetration and trial count n.

F LARGE MAZE

The main text reports both presets at $H = 64$; this appendix gives their protocol and further results.

Configurations. The large maze is the D4RL maze2d-large layout of Gymnasium-Robotics (12×9 cells), with the start cell $(1, 3)$ and the goal cell $(7, 9)$ fixed. The *broad* and *narrow* presets are the superellipse obstacle configurations that DiRecT (Giaretta et al., 2026) defines for this layout, with two and six obstacles (Figure 3); the walls are enforced by the simulator and are not part of the constraint set. Episodes last at most 1200 control steps. The sampling methods pad the obstacles by 0.01 in their potentials and selectors, and Projection by the ball radius 0.1.

Planning model. The maze model is a joint state–action diffusion model over 64-step segments (forces, positions, velocities and the goal position pinned at every step), a temporal U-Net of base width 64 trained for 60k steps on 250 self-collected episodes of the large maze with random start and goal cells; five independently initialised models are used. Sampling clips the predicted clean plan to the normaliser's range at every reverse step. Unguided, the models reach the fixed goal cell on every trial, but their routes graze the test-time obstacles: clean success is 0.06 on broad and 0.02 on narrow (Table 1).

Potential. The maze potential is the summed penetration depth of the plan's positions after the pinned start (63 at $H = 64$) into the test-time obstacles, padded by 0.01. Unlike Avoiding's maximum over geometric, dynamics and bound residuals, this score measures only obstacle penetration. All population methods use the same score; dynamics rows remain in the projector's constraint set. SMC and P3 use $\lambda_{\text{end}} = \beta_{\text{end}} = 5$.

Executed paths. Figure 8 overlays the executed paths of every trial for one model seed. All routes are in the same homotopy class, so the constraint concerns where inside the corridor the ball travels. The plotted P3 trajectories pass farther from the obstacles at corridor corners.

Projection baseline. Projection projects the predicted clean plan onto the padded constraint set at each of the last 11 of the 20 reverse steps, with SLSQP over 512 variables. On one GPU it takes 27 s per control step on narrow, against 0.09–0.12 s for the population methods (Appendix D.1). Its computational cost limited the campaign to 5–10 trials per seed.

Blocked-corridor preset. A further preset replaced the broad route obstacle by a block that closes the direct corridor for the ball, leaving a clean route of 13.8 maze units against 9.0 for the direct

one. Every sampling method, P3 included, drove through the block on all 25 trials of one model seed. The model was trained on shortest-path demonstrations with random endpoints, and none of the methods found the detour with $N = 16$ in these trials. These trials do not establish that the detour has zero model probability. A population of N fresh plans finds a route of prior mass p with probability at most Np, so low-probability routes can remain undiscovered. The medium-maze experiment of Appendix G instead trains on demonstrations of both routes and uses horizon $H = 8$.

G MEDIUM MAZE

Configuration. The medium maze is the D4RL maze2d-medium layout of Gymnasium-Robotics (8×8 cells). Between the start cell $(1, 1)$ and the goal cell $(6, 6)$ the layout offers two routes: a direct corridor through row 3 of ten cells, and a detour through the lower-left rooms of thirteen cells. The *blocked* preset places a power-4 superellipse of semi-axis 0.6 on the first cell of the direct corridor, which closes it for the ball, and a disc of radius 0.6 over the row-4 corridor before the goal, which leaves a passage 0.2 units wide along the corridor's lower wall (Figure 3). The preset is ours; DiRecT publishes obstacles for the large layout only. As on the large maze the walls are enforced by the simulator and are not part of the constraint set.

Data and model. The planning model is trained on 200 scripted demonstrations of this single start–goal pair, 100 through each route, generated by the same waypoint-following PD controller and action noise as the large-maze data (Figure 9, left). The model is the Avoiding architecture at $H = 8$, $K = 20$, base width 32, trained for 30k steps. We omit goal conditioning because every demonstration and evaluation episode uses the same goal. Five independently initialised models are used.

Behavior of the unguided model. Half the demonstrations use the detour, but only about 1% of the unguided per-step rollouts enter it across the five evaluation seeds. Figure 9 (right) shows rollouts that follow the direct corridor. An 8-step plan covers about 0.07 maze units. At the branch cell, the sampled continuation depends strongly on the conditioning velocity: at rest, 99% of 256 sampled plans head toward the direct corridor; with the demonstrations' downward entry velocity, 100% head toward the detour. P3 enters the detour in 42% of the evaluation episodes.

Horizon. Table 1 uses $H = 8$. At this horizon, projected plans often stop in front of the block, and the agent can remain there across repeated replanning calls.

Protocol and settings. 50 trials per seed and method, replanning after every action, at most 800 control steps, with methods paired by trial seed. Unguided sampling uses one particle; Projection pads the obstacles by the ball radius 0.1; best-of-N, cold SMC and P3 use $N = 16$, $\lambda = \beta = 5$ and pad the obstacles by 0.3 in the potential and the selector. P3 uses $t_0 = 4$.

Results. Projection enters the detour in 0.59 of episodes, compared with 0.42 for P3, but reaches the goal in only 0.26. Its mean accumulated penetration is 4.0, compared with 7.1 for P3. Figure 6 (right) includes trajectories that stop near the block, enter the blocked corridor, or take the detour.

H ANTMAZE

Model and closed-loop protocol. We use the $H = 32$, $K = 50$ goal-conditioned inverse-dynamics checkpoint trained on `antmaze-umaze-v1`. The diffusion model predicts a 29-dimensional state trajectory; a two-hidden-layer inverse-dynamics network with Mish activations maps adjacent predicted states to the eight Ant torques. We load the EMA weights, do not clip denoised states, execute five actions from each plan, and then replan from the measured state. Unguided sampling, Projection, best-of-N and SMC begin each replan from fresh Gaussian noise, and RDM + projection keeps one plan (Appendix B). P3 carries its population and log-weights across replans: it drops the five executed states of every plan, repeats the last one to restore the horizon, and re-noises to $t_0 = 40$, where $\gamma_{t_0} = 0.58$.

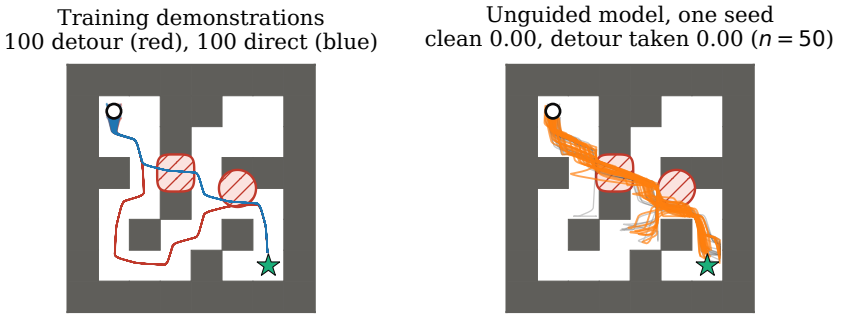


Figure 9: Medium maze, blocked. Left: the 200 training demonstrations, half through the detour (red) and half through the direct corridor (blue); the obstacles exist only at test time. Right: the unguided $H = 8$ model takes the direct corridor on every trial in the plotted seed (orange: goal after a violation; grey: failure).

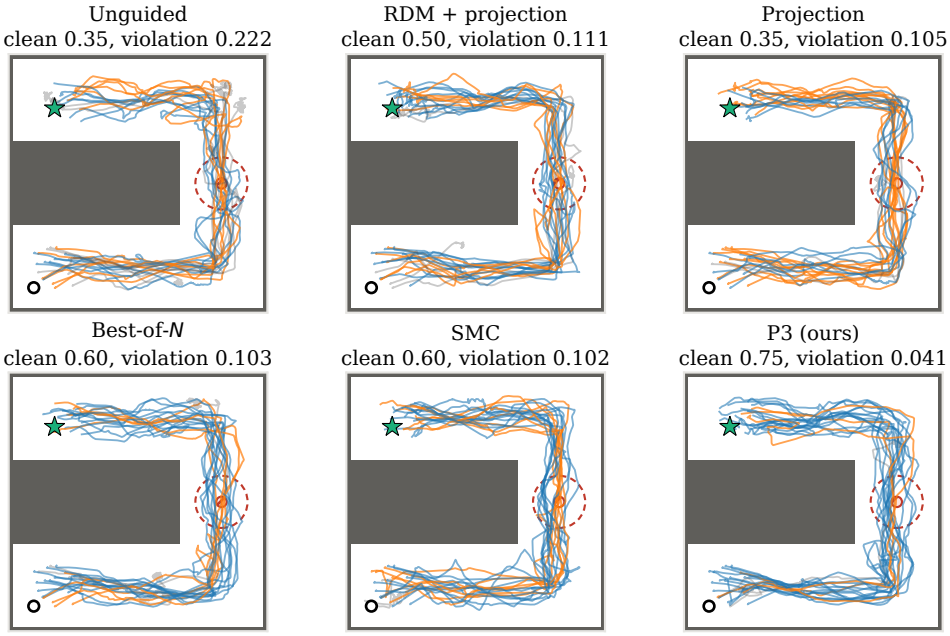


Figure 10: Executed AntMaze paths for one block of 20 trials, population methods at $N = 32$. Pooled results over all six blocks are in Table 1 for the one-sample methods and in Table 13 for SMC and P3. Blue: clean success; orange: goal after entering the obstacle; grey: failure. Hatched disc: graded region; dashed circle: planning clearance.

Constraint and methods. The inference-only obstacle is a disc centered at $(4, 0)$ in the vertical corridor. Its physical radius 0.25 grades the executed center path, while every constraint-aware method plans with an additional 1.0 clearance margin (Figure 3). It is absent from both the training data and MuJoCo collision geometry. Unguided uses one constraint-unaware plan; RDM + projection keeps one plan, using the same projector and margins as Projection, and repairs or regenerates it when the model's likelihood drops or the plan enters the padded disc. Projection projects one sample during the last 10% of the reverse chain. In Table 1, best-of-N, SMC, persistent best-of-N and P3 use $N = 4$; best-of-N executes the least-violating of N independent plans. SMC and P3 use $\lambda_{\text{end}} = \beta_{\text{end}} = 10$, accumulated weighting, ESS resampling and minimum-violation final selection, and differ only in P3's persistence; P3 uses no rejuvenation.

Population size. Table 13 gives SMC and P3 at $N = 4$, 8, 16 and 32 on the same 120 episodes. P3's clean success stays between 0.57 and 0.62 at every N. SMC is lower at $N = 4$ and 8 than at $N = 32$. With $N = 32$, P3 and SMC are close (0.61 and 0.60), and P3 enters the disc in 23% of episodes against 30% for SMC.

Table 13: AntMaze clean success of SMC and P3 by population size N, mean $\pm$ standard deviation across six evaluation seeds of 20 trials. Table 1 reports $N = 4$.

N	SMC	P3 (ours)
4	0.53 ± 0.11	0.62 ± 0.10
8	0.50 ± 0.12	0.57 ± 0.12
16	0.58 ± 0.09	0.57 ± 0.10
32	0.60 ± 0.08	0.61 ± 0.10

Tracking and constraint violations. Around the turn, the executed state trajectory can differ from the projected, geometrically feasible plan. Population sampling selects among model-generated trajectories without projecting their states, but this does not guarantee clean execution either (Figure 10).