DAPI: DUAL ADAPTIVE POLICY IMPROVEMENT FOR CONTACT-RICH MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Contact-rich manipulation can use action proposals ranging from sampling-based Gaussian proposals to learned Gaussian actors and generative priors. These proposals can all be improved at test time through model-based sampling and evaluation. Their performance can degrade, however, when the physical conditions at deployment differ from those assumed during planning, training, or data collection. Online model identification can mitigate this mismatch, but it creates a dual-control problem. Improving the model requires informative interactions, yet those interactions are selected by a policy that is itself affected by the current model mismatch. A task-driven policy may therefore fail to excite the uncertain parameters, leaving insufficient information for identification. The resulting uncertainty then continues to limit subsequent model-based policy improvement. We present DAPI, which augments task-driven policy improvement with an adaptive information-seeking channel. Among candidate action sequences that remain competitive for the task, DAPI favours actions whose predicted outcomes better distinguish the current physical hypotheses, and automatically removes this preference as parameter uncertainty contracts. A receding-window Stein update uses the resulting interaction data for online physical adaptation. The same candidate-reweighting mechanism applies without retraining to Gaussian sampling, frozen actors, and flow-matching proposals, with a noise-space update for the parameter-conditioned flow. Under substantial model mismatch in planar pushing, the receding-window update reduces median terminal error by 67%, whereas recursive Stein inference does not improve the nominal model. On lifting with an unknown center of mass, Gaussian sampling with online parameter adaptation alone succeeds in 0/12 rod groups and 0/24 plate cases, whereas adding DAPI's information-seeking channel with the same proposal succeeds in 12/12 and 14/24.

1 INTRODUCTION

Contact-rich manipulation can use action proposals with very different structures: Gaussian sampling for model-based control, learned Gaussian-like actors, and non-Gaussian generative priors such as flow matching. All can be improved at test time by sampling candidate action sequences and evaluating them through a physical model. Their performance can nevertheless degrade under model mismatch, because rollout models and learned priors reflect the physical conditions used during planning, training, or data collection. Classical indirect adaptive control addresses such mismatch by identifying uncertain physical parameters online and using the adapted model for control. In contact-rich manipulation, however, control and identification are coupled: better policy improvement requires a more accurate model, while improving the model requires informative interactions generated by a controller that is itself still affected by the current mismatch. A task-driven policy may therefore fail to excite the uncertain parameters, leaving identification with little useful information; the uncertainty then persists and continues to limit subsequent policy improvement. Figure 1 illustrates this failure on lifting with an unknown center of mass.

Existing methods address different parts of this problem. Robust and domain-randomized policies reduce sensitivity to physical variation, online system identification adapts models from deployment interaction, and dual-control or active-identification methods deliberately seek informative actions. Separately, test-time policy-improvement methods modify pretrained action proposals without retraining them. What remains missing is a general deployment-time adaptation mechanism for such

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.

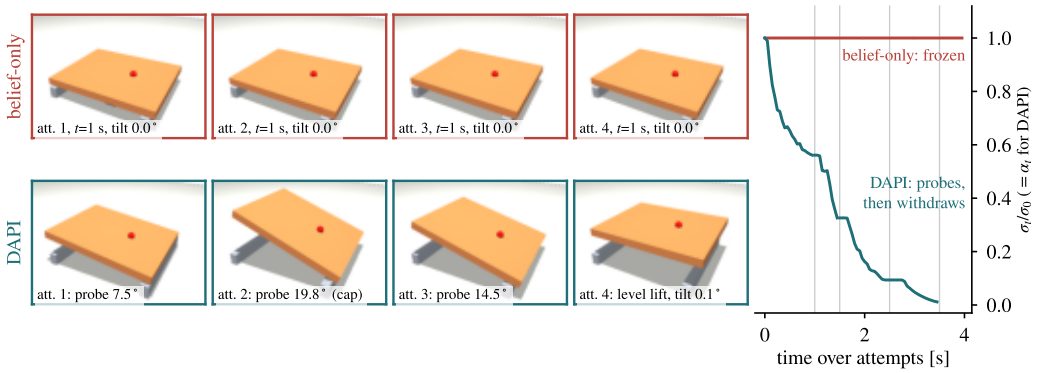


Figure 1: **A controller that is unsure where the mass is lifts nowhere, and so never learns where the mass is.** Plate lift with an unknown center of mass (red: true COM, 4.5 cm off-center; MuJoCo renders of recorded attempts). *Top:* belief-only task-driven improvement (online parameter adaptation without the information-seeking channel) repeatedly avoids lifting, so the belief does not contract. *Bottom:* DAPI selects informative but task-competitive lifts, contracts the belief, then retreats and lifts level on the fourth attempt. Right: belief spread over the concatenated attempts.

proposals that identifies the relevant physical parameters online while accounting for the dual coupling between task actions and the information they provide.

We address this question with DAPI. For every sampled action candidate, rollouts over the current parameter belief evaluate both task cost and how strongly the predicted outcome separates the physical hypotheses. An adaptive information-seeking channel then reweights only candidates that remain competitive for the task and fades as the parameter belief contracts. After execution, a receding-window Stein update uses the observed interaction to adapt the parameter belief. The same candidate-weighting principle applies across proposal structures: Gaussian sampling and actor-based proposals are updated in action space, while a parameter-conditioned flow is updated in its initial-noise space to preserve the frozen generative prior.

Our contributions are threefold. First, we characterise a failure mode of belief-aware policy improvement in which non-exciting executed actions stall online identification; Proposition 1 formalizes this mechanism. Second, we introduce DAPI, which adds parameter-exciting interactions only within the task-competitive part of the proposal and retreats as uncertainty contracts, together with a receding-window Stein update for online parameter adaptation. Third, we apply the same candidate-reweighting mechanism without retraining to Gaussian sampling, frozen learned actors, and frozen flow-matching proposals, including a structure-preserving noise-space update for a parameter-conditioned flow. Experiments on pushing and lifting test the predicted stall and retreat regimes, while a deformable-rope study exposes the limitation of the current spread-based retreat rule.

2 Related Work

Online physical adaptation and active identification. Model mismatch can be handled before deployment by training over physical variation, as in domain randomization and robust simulation training (Tobin et al., 2017; Peng et al., 2018), or online from interaction history (Kumar et al., 2021; Nagabandi et al., 2019). Classical indirect adaptive control instead estimates uncertain physical parameters during execution and uses the adapted model for control. When control also determines which data are observed, this becomes the dual-control problem (Feldbaum, 1960; Åström & Helmersson, 1986; Mesbah, 2018). Practical approximations seek informative actions (Klenske & Hennig, 2016; Parsi et al., 2020), while belief-space methods reason about their effect on future uncertainty (Kaelbling et al., 1998; Platt et al., 2010). In manipulation, ASID (Memmel et al., 2024) learns a dedicated exploration policy before the downstream task. We instead interleave identification with task execution and seek excitation within the task-driven proposal itself.

Uncertainty-aware sampling-based control and Stein inference. Sampling-based optimal control such as MPPI (Williams et al., 2017) improves action proposals from sampled trajectories, while Stein variational MPC supports richer control distributions (Lambert et al., 2021). Candidate actions can be evaluated across model ensembles (Abraham et al., 2020), and DuSt-MPC (Barcelos et al., 2021) combines online Stein inference over dynamics parameters with Stein MPC. Recent Stein variational control for contact-rich manipulation likewise accounts for parametric uncertainty during control (Sathyanarayan et al., 2026). These methods propagate parameter uncertainty into planning, but task-driven action selection can still generate interactions that are uninformative for the parameter update. We therefore combine SVGD-based online adaptation with explicitly information-aware policy improvement.

Test-time improvement of learned action priors. Diffusion and flow policies (Chi et al., 2023; Lipman et al., 2023) can be guided toward task objectives (Janner et al., 2022; Chung et al., 2023), improved with learned value functions (Park et al., 2025), or optimized in latent or noise space (Wagenmaker et al., 2025). In our setting, learned actors and generative policies remain frozen and serve as action priors inside the same model-based policy-improvement loop as Gaussian sampling. DAPI changes how their sampled candidates are improved using the online physical belief: Gaussian and actor-based proposals are updated in action space, while a parameter-conditioned flow is updated in its initial-noise space.

Differentiable physical models. Our policy-improvement and adaptation loops require batched rollouts across action candidates and parameter hypotheses, together with parameter gradients. For rigid manipulation we use ContactSDF (Yang & Jin, 2025), building on differentiable quasi-dynamic contact models (Pang et al., 2023); for the deformable study we use a reduced geometric-variable-strain model following Renda et al. (2018); Boyer et al. (2021).

3 Problem Setting

The physical system evolves as

$$\boldsymbol{x}_{t+1} = f^\star(\boldsymbol{x}_t, \boldsymbol{u}_t; \boldsymbol{\theta}^\star) + \boldsymbol{w}_t, \tag{1}$$

where $f^\star : \mathbb{R}^n \times \mathbb{R}^m \times \Theta \to \mathbb{R}^n$ denotes the physical dynamics, $\boldsymbol{x}_t \in \mathbb{R}^n$ is the observed state, $\boldsymbol{u}_t \in \mathbb{R}^m$ the control input, $\boldsymbol{w}_t \in \mathbb{R}^n$ process noise, and $\boldsymbol{\theta}^\star \in \Theta \subset \mathbb{R}^d$ the task-relevant physical parameters unknown to the controller. For model-based policy improvement, the controller uses a differentiable parameterized surrogate $F_{\mathrm{p}}(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\theta})$ of $f^\star$. The same surrogate is used for online parameter adaptation from observed interactions.

At state $\boldsymbol{x}_t$, an *action proposal* π generates K candidate action sequences over horizon H,

$$\boldsymbol{U} = (\boldsymbol{u}_0, \dots, \boldsymbol{u}_{H-1}) \in \mathbb{R}^{H\times m}, \qquad \mathcal{B}_t = \{\boldsymbol{U}^n\}_{n=1}^K.$$

When the proposal is learned, its parameters remain frozen at test time. The task cost $J(\boldsymbol{U}; \boldsymbol{x}_t, \boldsymbol{\theta})$ is obtained by rolling $\boldsymbol{U}$ through F_{p} under parameter hypothesis $\boldsymbol{\theta}$. Candidates carry proposal weights ρ_n; we use $\rho_n = 1/K$. We consider Gaussian sampling, a frozen Gaussian actor, and frozen flow-matching policies. Their proposal-specific representations and test-time updates are described in Section 4.4.

The controller maintains a weighted particle belief over the uncertain physical parameters, $q_t = \sum_{i=1}^N a_{i,t}\delta_{\boldsymbol{\theta}_{i,t}}$, where $\delta_{\boldsymbol{\theta}_{i,t}}$ denotes a Dirac point mass at the i-th parameter hypothesis $\boldsymbol{\theta}_{i,t}$ and $a_{i,t}$ is its normalized weight. Model-based policy improvement evaluates every candidate $\boldsymbol{U}^n$ across the current parameter hypotheses using the same surrogate F_{p}. After the first control interval is executed, the resulting interaction updates the parameter belief.

We denote model-based proposal improvement and action selection by C and the online parameter update by T:

$$\boldsymbol{U}_t = \mathsf{C}(\boldsymbol{x}_t, q_t, \mathcal{B}_t), \qquad q_{t+1} = \mathsf{T}(q_t, \boldsymbol{x}_t, \boldsymbol{U}_t, \boldsymbol{x}_{t+1}).$$

The objective is task performance under the unknown $\boldsymbol{\theta}^\star$, not parameter identification by itself. However, the data available to T are generated by C, which is itself operating under the current model uncertainty. Task-driven improvement can therefore select interactions that provide little information about the uncertain parameters, allowing the uncertainty to persist and continue limiting subsequent policy improvement. Section 4 develops DAPI for this coupled setting.

4 DUAL ADAPTIVE POLICY IMPROVEMENT

DAPI couples online parameter adaptation with model-based policy improvement. At each control step, it updates the parameter belief from recent executed interactions (Section 4.1), performs task-driven policy improvement under that belief (Section 4.2), adds an adaptive information-seeking channel within task-competitive candidates (Section 4.3), and applies the resulting candidate weights in the representation of the action proposal (Section 4.4). Algorithm 1 summarizes the complete loop.

4.1 ONLINE PARAMETER ADAPTATION FROM A RECEDING INTERACTION WINDOW

The belief is represented by N weighted parameter particles and updated from recent executed interactions. Let $\mathcal{W}_t$ denote the current interaction window, and let $\boldsymbol{y}_{k+1}$ denote the measured observation components used for identification. For a parameter hypothesis $\boldsymbol{\theta}$, the surrogate F_{p} is evaluated over the recorded interaction window under the executed controls, yielding predicted observations $\hat{\boldsymbol{y}}^{\boldsymbol{\theta}}_{k+1}$. We write the corresponding window prediction error as

$$\mathcal{E}_t(\boldsymbol{\theta}) = \sum_{k \in \mathcal{W}_t} \omega_{k,t} \, \tfrac{1}{2} \left\| D^{-1} \left(\hat{\boldsymbol{y}}^{\boldsymbol{\theta}}_{k+1} - \boldsymbol{y}_{k+1} \right) \right\|^2, \tag{2}$$

where D is a diagonal matrix of observation scales and $\omega_{k,t}$ denotes the window weighting. We use uniform weights unless otherwise stated.

The interaction data update the particle representation of the current physical uncertainty, but the role of particle weights differs across tasks. The common operation is Stein transport toward the window-conditioned target $p_t(\boldsymbol{\theta}) \propto p_0(\boldsymbol{\theta}) \exp[-\beta \mathcal{E}_t(\boldsymbol{\theta})]$, using Stein variational gradient descent (SVGD) (Liu & Wang, 2016),

$$\boldsymbol{\theta}_i \leftarrow \boldsymbol{\theta}_i + \epsilon_t \sum_j a_j \Big[k(\boldsymbol{\theta}_j, \boldsymbol{\theta}_i) \big(\nabla \log p_0(\boldsymbol{\theta}_j) - \beta \nabla \mathcal{E}_t(\boldsymbol{\theta}_j) \big) + \nabla_{\boldsymbol{\theta}_j} k(\boldsymbol{\theta}_j, \boldsymbol{\theta}_i) \Big], \tag{3}$$

where p_0 is the parameter prior, β controls the influence of the prediction error, k is the Stein kernel, and ϵ_t is the update step size. The gradient $\nabla \mathcal{E}_t$ is obtained by automatic differentiation through F_{p}. In equation 3, a_j denotes the weight used in the Stein transport: lifting uses the current recursively updated particle weights, whereas pushing and rope use equal transport weights. The weighted belief used for policy improvement is then constructed according to the task-specific rule described in Appendix B.

Unlike sequential Stein inference that incorporates each transition only when it arrives, the receding-window update repeatedly evaluates recent executed interactions under the current parameter hypotheses. This reduces the persistent influence of stale residuals when the surrogate does not exactly match the physical system.

4.2 POLICY IMPROVEMENT UNDER PARAMETER UNCERTAINTY

Task-driven improvement. For each candidate $\boldsymbol{U}^n \in \mathcal{B}_t$, we evaluate its expected task cost under the current parameter belief,

$$S_n = \sum_{i=1}^{N} a_i \, J(\boldsymbol{U}^n; \boldsymbol{x}_t, \boldsymbol{\theta}_i), \tag{4}$$

where the time index is suppressed within a control step. The $N \times K$ candidate–parameter rollouts are evaluated in batch through F_{p}.

We then perform KL-regularized policy improvement over the candidate weights,

$$w^{\mathrm{T}} = \arg\min_{w \in \Delta_K} \; \mathbb{E}_w[S] + \lambda \, D_{\mathrm{KL}}(w \,\|\, \rho) \quad \Longrightarrow \quad w^{\mathrm{T}}_n \propto \rho_n e^{-S_n/\lambda}, \tag{5}$$

where Δ_K is the probability simplex over the K candidates, ρ denotes their proposal weights, and $\lambda > 0$ is the temperature. This is the usual MPPI-style soft reweighting (Williams et al., 2017), with task cost averaged over the current physical belief. How these candidate weights modify each proposal class is described in Section 4.4.

Loss of excitation. Online parameter adaptation requires interactions whose predicted observations vary with the uncertain parameters. Let $\hat{\boldsymbol{y}}_{1:H}(\boldsymbol{U};\boldsymbol{\theta})$ denote the observation trajectory predicted by F_{p} when candidate $\boldsymbol{U}$ is applied under parameter $\boldsymbol{\theta}$. We call an interaction *non-exciting* for the current belief if these predictions are locally invariant to $\boldsymbol{\theta}$ over a neighbourhood $\mathcal{V}$ containing the supported parameter hypotheses. Such an interaction cannot distinguish the hypotheses because changing the parameter produces the same predicted observations.

This creates a failure mode for task-driven policy improvement. A candidate may have low task cost under the belief while its executed prefix is non-exciting. Because only the first control interval is executed before replanning, repeatedly selecting such candidates can prevent the interaction needed for online identification.

Proposition 1 (Task-driven policy improvement can stall identification)**.** *Let $\mathcal{V}$ contain the support of the current parameter belief. If the executed interaction is non-exciting over an identification window, then $\mathcal{E}_t(\boldsymbol{\theta})$ is constant on $\mathcal{V}$ and $\nabla_{\boldsymbol{\theta}}\mathcal{E}_t(\boldsymbol{\theta}) = 0$, so the executed data provide no parameter-discriminating gradient. Moreover, if the belief-averaged task cost has a unique minimizer $\boldsymbol{U}^* \in \mathcal{B}_t$ with a non-exciting executed prefix, then as $\lambda \to 0$ the task-driven weights concentrate on $\boldsymbol{U}^*$. If such a non-exciting task minimizer recurs under receding-horizon execution, identification can stall.*

A proof is given in Appendix A.

4.3 Task-competitive excitation with adaptive retreat

The previous subsection shows that task-driven policy improvement can select non-exciting interactions. DAPI therefore adds an information-seeking candidate reweighting, but restricts it to candidates that are already competitive for the task. The contribution of this information-seeking update is reduced as the parameter belief contracts.

Excitation score. For candidate $\boldsymbol{U}$, let $\hat{y}_{h,d}(\boldsymbol{U};\boldsymbol{\theta}_i)$ denote the d-th predicted observation at rollout step h under parameter hypothesis $\boldsymbol{\theta}_i$. Its belief-weighted predictive mean and variance are

$$\bar{y}_{h,d}(\boldsymbol{U}) = \sum_i a_i\, \hat{y}_{h,d}(\boldsymbol{U};\boldsymbol{\theta}_i), \qquad V_{h,d}(\boldsymbol{U}) = \sum_i a_i \big(\hat{y}_{h,d}(\boldsymbol{U};\boldsymbol{\theta}_i) - \bar{y}_{h,d}(\boldsymbol{U})\big)^2. \tag{6}$$

We score the candidate by

$$\mathcal{I}(\boldsymbol{U}; q_t) = \frac{1}{2}\sum_{h=0}^{H-1}\sum_d \log\left(1 + \frac{V_{h,d}(\boldsymbol{U})}{\sigma_d^2}\right), \tag{7}$$

where σ_d is the observation-noise scale for coordinate d, i.e., the corresponding diagonal entry of D in equation 2. This is a Gaussian moment-matched information proxy: it vanishes when the current parameter hypotheses predict identical observations and increases as their predicted outcomes separate relative to the observation noise. It uses the same particle rollouts as the belief-averaged task-cost evaluation.

Excitation among task-competitive candidates. Writing the ordered task costs as $S_{[1]} \le \cdots \le S_{[K]}$ and $k_\tau = \lceil \tau K \rceil$, we define the task-competitive set as the lowest-cost fraction τ of the sampled candidates,

$$\mathcal{A}_t = \{n : S_n \le S_{[k_\tau]}\}, \qquad k_\tau = \lceil \tau K \rceil. \tag{8}$$

For $n \in \mathcal{A}_t$, let $\mathcal{I}_n = \mathcal{I}(\boldsymbol{U}^n; q_t)$ and min–max normalize these values to $r_n \in [0,1]$, with $r_n = 0$ when all scores in $\mathcal{A}_t$ are equal. We set $\kappa_t = \max\big(S_{[k_\tau]} - \min_m S_m,\ \lambda\big)$, so that the information term is scaled to the task-cost variation inside $\mathcal{A}_t$, with λ as a floor. The information-seeking weights are

$$w_n^{\mathrm{I}} \propto \mathbf{1}[n \in \mathcal{A}_t]\, \rho_n \exp\left(-\frac{S_n - \kappa_t r_n}{\lambda}\right), \tag{9}$$

and the final candidate weights interpolate between the task-driven and information-seeking updates,

$$w^\alpha = (1-\alpha_t) w^{\mathrm{T}} + \alpha_t w^{\mathrm{I}}. \tag{10}$$

Thus the information-seeking channel redistributes weight only within the task-competitive part of the current proposal. By equation 10, α_t directly controls its contribution, while candidates outside $\mathcal{A}_t$ receive no additional information-seeking weight. Additional properties are given in Appendix A.

Adaptive retreat. The information-seeking channel should weaken as uncertainty about the physical parameters is reduced. Let $\boldsymbol{z}_{i,t} = \mathcal{N}_\theta(\boldsymbol{\theta}_{i,t})$ denote normalized particle coordinates and $\bar{\boldsymbol{z}}_t = \sum_i a_{i,t}\boldsymbol{z}_{i,t}$. We measure the current belief spread by $\sigma_t^2 = \sum_i a_{i,t}\|\boldsymbol{z}_{i,t} - \bar{\boldsymbol{z}}_t\|^2$ and set

$$\alpha_t = \alpha_0 \min\left(1, \frac{\sigma_t}{\sigma_0}\right), \tag{11}$$

where σ_0 is the initial belief spread and $\alpha_0 \in [0, 1]$ the maximum contribution of the information-seeking channel. If the predicted observations are Lipschitz in the normalized parameter coordinates, then $\mathcal{I}(\boldsymbol{U}; q_t) = O(\sigma_t^2)$ (Appendix A). Hence, as the belief contracts, α_t decreases and w^α approaches the task-driven weights w^{T}.

This retreat rule depends only on belief spread. If the remaining uncertainty cannot be reduced by the available interactions, σ_t may remain large and the information-seeking channel may remain active. We return to this limitation in Section 6.

4.4 Proposal-specific policy improvement

The candidate weights w^α are shared across proposal classes. What changes is the representation in which the weighted update is applied.

Gaussian sampling and Gaussian actor. For Gaussian proposals, an $H \times m$ action sequence is identified with its vectorization in $\mathbb{R}^{Hm}$. Candidate sequences are sampled as

$$\boldsymbol{U}^n = \boldsymbol{m} + \boldsymbol{\Sigma}^{1/2}\boldsymbol{\epsilon}^n, \qquad \boldsymbol{\epsilon}^n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{Hm}), \tag{12}$$

with fixed diagonal covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{Hm \times Hm}$. For sampling-based MPC, $\boldsymbol{m}$ is initialized by shifting the previous action sequence by one step. For the frozen Gaussian actor, the actor is rolled out through F_{p} from $\boldsymbol{x}_t$ to provide the initial mean sequence $\boldsymbol{m}$.

After candidate evaluation, both proposals use the same action-space update, $\boldsymbol{m} \leftarrow \sum_n w_n^\alpha \boldsymbol{U}^n$. If another policy-improvement iteration is performed, candidates are resampled around the updated mean using equation 12.

Randomized flow. A frozen flow-matching policy (Lipman et al., 2023) defines a generator

$$\boldsymbol{U} = G_\phi(\boldsymbol{\xi}; \boldsymbol{c}), \qquad \boldsymbol{\xi} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{Hm}),$$

where G_ϕ is obtained by integrating the learned flow and its parameters ϕ remain frozen at test time. For the randomized flow, $\boldsymbol{c} = \boldsymbol{x}_t$. The flow generates the initial candidate set, after which the candidates are reweighted under the current parameter belief and refined using the same Gaussian action-space iterations as above. The final improved sequence therefore need not remain an exact output of G_ϕ.

Parameter-conditioned flow. For the parameter-conditioned flow, we preserve the frozen generative structure and instead update its initial-noise representation. The conditioning parameter is the highest-weight current hypothesis,

$$j = \arg\max_i a_i, \qquad \boldsymbol{c} = (\boldsymbol{x}_t, \boldsymbol{\theta}_j).$$

Candidates are generated from

$$\boldsymbol{\xi}^n = \bar{\boldsymbol{\xi}} + \sigma_\xi \boldsymbol{\epsilon}^n, \qquad \boldsymbol{U}^n = G_\phi(\boldsymbol{\xi}^n; \boldsymbol{c}),$$

but each $\boldsymbol{U}^n$ is still evaluated across the full parameter belief when computing S_n and $\mathcal{I}_n$. The weighted update is performed in noise space,

$$\bar{\boldsymbol{\xi}} \leftarrow \sum_n w_n^\alpha \boldsymbol{\xi}^n,$$

and the executed sequence is $G_\phi(\bar{\boldsymbol{\xi}}; \boldsymbol{c})$. The noise mean $\bar{\boldsymbol{\xi}}$ is carried between control steps and reset when the selected conditioning hypothesis changes. Thus the improved sequence remains an output of the frozen flow map while its policy-improvement weights are determined by the same task and information criteria as for the action-space proposals. Related work has also used noise-space steering of frozen generative policies (Wagenmaker et al., 2025); additional training details are given in Appendix D.

Algorithm 1 DAPI and the two realizations of its policy-improvement step. The belief update is shared; the proposal determines whether the weighted update is performed in action space (1a) or in the initial-noise space of the parameter-conditioned flow (1b). Setting $\alpha_t \equiv 0$ gives belief-aware task-driven improvement.

DAPI *input:* proposal π, F_{p}, initial belief q_0, λ, τ, α_0

1: **for** $t = 0, 1, \ldots$ **do**
2: $\quad \boldsymbol{U}_t \leftarrow \textsc{Improve}(\boldsymbol{x}_t, q_t, \alpha_t)$ ▷ Algorithm 1a or 1b
3: $\quad$ execute the first control interval of $\boldsymbol{U}_t$, observe $\boldsymbol{x}_{t+1}$
4: $\quad$ update q_{t+1} from recent interaction using the task-specific particle weighting and Stein transport of Section 4.1; α_{t+1} by equation 11

1a IMPROVE: action-space proposal

1: initialise $\mathcal{B}_t$ from π: Gaussian sampling, actor prior, or randomized flow
2: **repeat** M times:
3: $\quad$ roll out F_{p} for all (i, n): $S_n, \mathcal{I}_n$
4: $\quad w^{\alpha}$ by equation 5, equation 8, equation 10
5: $\quad \boldsymbol{m} \leftarrow \sum_n w_n^{\alpha} \boldsymbol{U}^n$
6: $\quad \boldsymbol{U}^n = \boldsymbol{m} + \boldsymbol{\Sigma}^{1/2} \boldsymbol{\epsilon}^n, n \le K$ ▷ if another iteration
7: **return** $\boldsymbol{m}$

1b IMPROVE: parameter-conditioned flow

1: $j \leftarrow \arg\max_i a_i$, $\boldsymbol{c} = (\boldsymbol{x}_t, \boldsymbol{\theta}_j)$; **if** j changed: $\bar{\boldsymbol{\xi}} \leftarrow \mathbf{0}$
2: **repeat** M times: ▷ noise-space update
3: $\quad \boldsymbol{\xi}^n = \bar{\boldsymbol{\xi}} + \sigma_\xi \boldsymbol{\epsilon}^n, \quad \boldsymbol{U}^n = G_\phi(\boldsymbol{\xi}^n; \boldsymbol{c}), n \le K$
4: $\quad$ roll out F_{p} for all (i, n): $S_n, \mathcal{I}_n$
5: $\quad w^{\alpha}$ by equation 5, equation 8, equation 10
6: $\quad \bar{\boldsymbol{\xi}} \leftarrow \sum_n w_n^{\alpha} \boldsymbol{\xi}^n$ ▷ kept for $t+1$
7: **return** $G_\phi(\bar{\boldsymbol{\xi}}; \boldsymbol{c})$

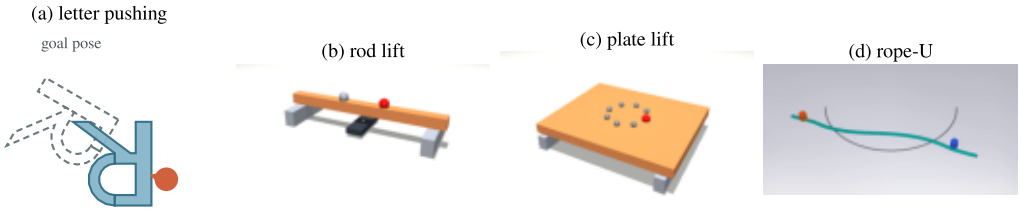


Figure 2: **Tasks.** (a) Letter-shaped object (R shown) and circular pusher; dashed: goal pose. (b) Rod on two pedestals with the lifting bar below; red: true center of mass, grey: its mirror hypothesis. (c) Plate; red: true center of mass, grey: the other seven load directions. (d) Rope-U shaping task.

5 EXPERIMENTS

We evaluate two questions. (Q1) Under model mismatch, does the receding-window Stein update outperform recursive Stein inference when the controller is fixed? (Q2) Can DAPI overcome identification stall when task-driven interaction is non-exciting, while retreating when ordinary task motion is already informative? Pushing evaluates Q1 and the informative-motion regime of Q2, lifting evaluates the non-exciting regime, and rope shaping tests the limitation of the spread-based retreat rule.

5.1 TASKS AND PROTOCOL

Pushing (Q1; Figure 2(a)): a sphere pushes a box or a letter-shaped object (V, D, R of the public Letters-Planar-Pushing geometry, extruded) to a goal pose on a MuJoCo plant whose object mass, ground friction and contact friction are changed together (mild, strong, extreme mismatch). The controller uses a planar ContactSDF surrogate calibrated on the matched plant, with uncertainty over its effective translational and rotational mobility and contact-friction parameters.

Rod lift (Figure 2(b)) and *plate lift* (Figure 2(c)) (Q2): a bar under the object must lift it level off two pedestals with the center of mass unknown along the rod or in the plane of the plate. The plant is simulated with MJX (MuJoCo dynamics), while ContactSDF is the parameterised surrogate F_{p} used for both model-based policy improvement and online identification of the center of mass (Table 1).

Table 1: Tasks, uncertain parameters, execution budgets, and units of evaluation.

Task	Unknown $\boldsymbol{\theta}$ (d)	Plant / Surrogate	Execution budget	Unit of evaluation
PushBox	mobility, contact friction (3)	MuJoCo / ContactSDF	56 steps	2 motions $\times$ 3 mismatches $\times$ 3 seeds
PushLetter	mobility, contact friction (3)	MuJoCo / ContactSDF	56 steps	3 letters $\times$ 3 mismatches $\times$ 3 seeds
Rod lift	COM along rod, ± 4 cm (1)	MJX / ContactSDF	6 s per attempt, ≤ 5 attempts	12 groups (2 COM $\times$ 6 seeds)
Plate lift	planar COM, 4.5 cm (2)	MJX / ContactSDF	1 s per attempt, ≤ 10 attempts	8 directions $\times$ 3 seeds

Table 2: **Online adaptation under model mismatch.** Terminal error [mm], median / mean / worst; wins: cases better than the nominal model. Belief-only rows isolate the belief update; the DAPI row is included to evaluate adaptive retreat when ordinary task motion is already informative (Section 5.4).

	PushBox (18 cases)		PushLetter (27 cases)	
Belief	error	wins	error	wins
Nominal model (no update)	29.9 / 32.0 / 91.2	–	30.0 / 28.6 / 40.2	–
Recursive (DuSt dynamics inference)	33.6 / 40.7 / 97.5	6	38.0 / 39.0 / 71.1	3
Recursive, with DuSt Stein MPC	37.5 / 46.1 / 102.3	5	46.3 / 42.0 / 68.9	5
Receding window (ours), belief-only	**9.8 / 14.4 / 55.2**	**15**	24.5 / 23.5 / 43.7	19
Receding window (ours), DAPI	10.0 / 26.3 / 128.5	14	**21.5 / 20.4 / 40.7**	**21**

5.2 Receding-window belief versus recursive Stein inference (Q1)

The primary comparison in Table 2 keeps the MPPI planner, surrogate, seeds and evaluation cases fixed and compares the receding-window update of Section 4.1 (window of 18 transitions, 6 SVGD steps per control step) with the recursive dynamics inference of DuSt-MPC (Barcelos et al., 2021) (our JAX reimplementation, 50 particles). The information-seeking channel is disabled in this comparison so that the parameter-update effect is isolated. We additionally report recursive inference together with the Stein MPC planner of DuSt-MPC as a system-level baseline. The receding-window belief lowers the terminal error below the nominal model in 15 of 18 box cases and 19 of 27 letter cases, and below recursive inference in 17 of 18 and 26 of 27, respectively. Recursive inference, with or without the Stein MPC planner, is worse than the nominal model in most cases. The gain over the nominal model is large on the box and smaller on the letters, whose rotational inertia and area-centroid offset are captured less accurately by the planar surrogate (Appendix E.2).

5.3 Identification stall and the information-seeking update (Q2)

Within each task in Table 3, the belief-only and DAPI columns share the proposal, surrogate, belief update, execution budget and seeds, and differ only in the information-seeking channel: *belief-only* sets $\alpha_t \equiv 0$, whereas DAPI uses equation 11 with $\alpha_0 = 1$ and $\tau = 0.1$. Each lifting group consists of repeated attempts on the same object with the parameter belief carried between attempts. Rod and plate attempts last 6 s and 1 s (see Appendix B), respectively; success criteria, stopping rules and additional task details are given in Appendix D.

With belief-averaged task-driven improvement, Gaussian sampling does not initiate informative lifting in any rod or plate case, consistent with the failure mechanism of Proposition 1. Adding the information-seeking channel raises its success from 0/12 to 12/12 on the rod and from 0/24 to 14/24 on the plate. The same mechanism also improves the frozen actor and randomised flow proposals, while the conditioned flow already provides a stronger belief-only baseline and benefits mainly on the plate (Table 3). Successful groups share the same qualitative sequence (Figures 3, 4 and 7): informative motion contracts the center-of-mass belief, α_t decreases, and a level lift follows. Additional diagnostics are given in Appendix E.1.

5.4 Retreat when task motion is already informative

PushBox and PushLetter instantiate the complementary Q2 regime: ordinary task motion already produces observations that discriminate the uncertain physical parameters. The information-seeking channel is therefore typically needed only transiently: ordinary task interaction contracts the belief and rapidly reduces α_t (Figure 3). On PushBox, α_t falls below 0.1 after a median of 5 control steps

Table 3: **Lifting with an unknown center of mass.** Successful groups (rod, 6 s per attempt, ≤ 5 attempts) and direction–seed pairs (plate, 1 s per attempt, ≤ 10 attempts). The two columns of a task differ only in the information-seeking channel.

	Rod lift (of 12)		Plate lift (of 24)	
Proposal	belief-only	DAPI	belief-only	DAPI
Gaussian sampling (MPPI)	0	**12**	0	**14**
Gaussian actor (RL, frozen)	2	**7**	0	**7**
Flow policy, randomised (frozen)	2	**12**	14	**17**
Flow policy, conditioned (frozen, Algorithm 1b)	10	10	13	**18**

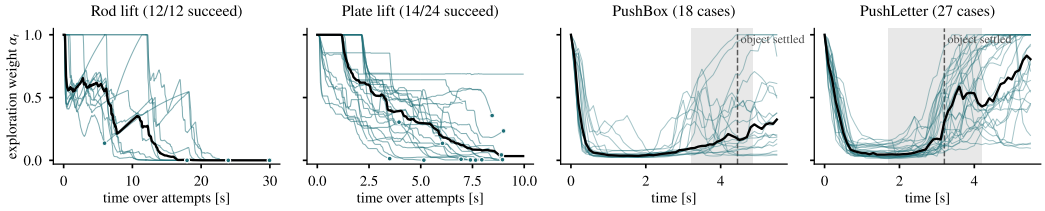


Figure 3: **Information-seeking weight α_t of DAPI in every case** (Gaussian sampling proposal). Left, excitation required: lifting, over the concatenated attempts of each group (rod 6 s, plate 1 s attempts); dots mark the end of the successful attempt. Right, task motion already informative: pushing, one line per case; dashed line and grey band: median and interquartile range of the time at which the object has settled. Black: median over cases (lifting: a finished group keeps its last value). Other proposals: Figure 6; rope: Appendix E.3.

in all 18 cases; on PushLetter it does so after a median of 7 steps in 24 of 27 cases. The median terminal error of DAPI remains close to belief-only adaptation on PushBox (10.0 versus 9.8 mm) and improves on PushLetter (21.5 versus 24.5 mm). After the object settles, the recent window can contain less parameter-discriminating motion and the windowed belief may broaden again, causing α_t to increase. This reflects that the current retreat rule tracks belief spread rather than the instantaneous informativeness or identifiability of the available actions; we return to this limitation in Section 6.

6 Discussion and Limitations

The experiments expose two task regimes of the same adaptive-control loop. In lifting, belief-averaged optimization suppresses the interaction needed for identification and the same cautious decision repeats; DAPI moves the proposal toward an informative but task-competitive interaction and returns to the task objective as the belief contracts. In pushing, ordinary task motion is already informative, so the information-seeking channel retreats rapidly. The parameter-conditioned flow reveals an additional proposal-dependent effect: conditioning on the most probable hypothesis already induces informative lifts in many cases, reducing the need for explicit excitation.

The rope study exposes the main limitation of the present retreat rule. Its nine-dimensional belief remains broad even though receding-window adaptation improves the task, so spread alone cannot distinguish reducible uncertainty from uncertainty that cannot be reduced by actions supported by the current proposal (Appendix E.3). An action-dependent identifiability measure is therefore a natural extension. DAPI is limited by the support of the underlying action proposal: it can reweight and refine proposed actions, but cannot recover behaviours that the proposal does not generate. Future work will extend the retreat mechanism with an action-dependent identifiability criterion and evaluate the resulting adaptive policy-improvement framework on real contact-rich manipulation systems.

AI USE STATEMENT

The research question and core method were proposed by the authors. Generative AI tools were used to assist with method and experiment implementation, experiment design and method variants, mathematical formalization and proofs, result analysis and interpretation, and drafting and editing the manuscript. They were not used to generate research data; all reported results come from executed simulations under fixed evaluation protocols. The authors reviewed the AI-assisted code, analyses, mathematical claims, and manuscript content and take responsibility for the final paper.

REPRODUCIBILITY STATEMENT

All runs of the final evaluation protocols are reported. The appendix provides the experimental settings, adaptation details, surrogate models, task definitions, learned-proposal training details, and computational cost. Code, protocols, and raw episode summaries will be released with the paper.

REFERENCES

Ian Abraham, Ankur Handa, Nathan Ratliff, Kendall Lowrey, Todd D. Murphey, and Dieter Fox. Model-based generalization under parameter uncertainty using path integral control. *IEEE Robotics and Automation Letters*, 5(2):2864–2871, 2020.

Karl Johan Åström and Anders Helmersson. Dual control of an integrator with unknown gain. *Computers & Mathematics with Applications*, 12(6):653–662, 1986.

Lucas Barcelos, Alexander Lambert, Rafael Oliveira, Paulo Borges, Byron Boots, and Fabio Ramos. Dual online Stein variational inference for control and dynamics. In *Robotics: Science and Systems (RSS)*, 2021.

Frédéric Boyer, Vincent Lebastard, Fabien Candelier, and Federico Renda. Dynamics of continuum and soft robots: A strain parameterization based approach. *IEEE Transactions on Robotics*, 37(3): 847–863, 2021.

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 (RSS)*, 2023.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *International Conference on Learning Representations (ICLR)*, 2023.

A. A. Feldbaum. Dual control theory. I. *Automation and Remote Control*, 21(9):874–880, 1960.

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

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

Edgar D. Klenske and Philipp Hennig. Dual control for approximate Bayesian reinforcement learning. *Journal of Machine Learning Research*, 17(127):1–30, 2016.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. RMA: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems (RSS)*, 2021.

Alexander Lambert, Adam Fishman, Dieter Fox, Byron Boots, and Fabio Ramos. Stein variational model predictive control. In *Proceedings of the Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 1278–1297, 2021.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations (ICLR)*, 2023.

Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Marius Memmel, Andrew Wagenmaker, Chuning Zhu, Patrick Yin, Dieter Fox, and Abhishek Gupta. ASID: Active exploration for system identification in robotic manipulation. In *International Conference on Learning Representations (ICLR)*, 2024.

Ali Mesbah. Stochastic model predictive control with active uncertainty learning: A survey on dual control. *Annual Reviews in Control*, 45:107–117, 2018.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2019.

Tao Pang, H. J. Terry Suh, Lujie Yang, and Russ Tedrake. Global planning for contact-rich manipulation via local smoothing of quasi-dynamic contact models. *IEEE Transactions on Robotics*, 39(6): 4691–4711, 2023.

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. In *International Conference on Machine Learning (ICML)*, 2025.

Anilkumar Parsi, Andrea Iannelli, and Roy S. Smith. Active exploration in adaptive model predictive control. In *IEEE Conference on Decision and Control (CDC)*, pp. 6186–6191, 2020. doi: 10.1109/CDC42340.2020.9304303.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

Robert Platt, Russ Tedrake, Leslie Kaelbling, and Tomás Lozano-Pérez. Belief space planning assuming maximum likelihood observations. In *Robotics: Science and Systems (RSS)*, 2010.

Federico Renda, Frédéric Boyer, Jorge Dias, and Lakmal Seneviratne. Discrete Cosserat approach for multisection soft manipulator dynamics. *IEEE Transactions on Robotics*, 34(6):1518–1533, 2018.

Hrishikesh Sathyanarayan, Victor Vantilborgh, Harish Ravichandar, Tom Lefebvre, and Ian Abraham. Distributionally robust control via Stein variational inference for contact-rich manipulation. In *Proceedings of Robotics: Science and Systems*, 2026. doi: 10.15607/RSS.2026.XXII.061.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

Andrew Wagenmaker, Mitsuhiko Nakamoto, Yunchu Zhang, Seohong Park, Waleed Yagoub, Anusha Nagabandi, Abhishek Gupta, and Sergey Levine. Steering your diffusion policy with latent space reinforcement learning. In *Proceedings of the Conference on Robot Learning*, 2025.

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 (ICRA)*, pp. 1714–1721, 2017. doi: 10.1109/ICRA.2017.7989202.

Wen Yang and Wanxin Jin. ContactSDF: Signed distance functions as multi-contact models for dexterous manipulation. *IEEE Robotics and Automation Letters*, 10(5):4212–4219, 2025. doi: 10.1109/LRA.2025.3548503.

APPENDIX

A PROOFS

Proof of Proposition 1. If the executed interaction is non-exciting on $\mathcal{V}$ over the identification window, the predicted observations entering equation 2 are invariant to $\boldsymbol{\theta}$ on $\mathcal{V}$. Hence $\mathcal{E}_t(\boldsymbol{\theta})$ is constant on $\mathcal{V}$ and

$$\nabla_{\boldsymbol{\theta}} \mathcal{E}_t(\boldsymbol{\theta}) = 0.$$

The executed data therefore contribute no parameter-discriminating gradient to the SVGD update.

For the second claim, let $\boldsymbol{U}^*$ be the unique minimiser of the belief-averaged task cost on the finite candidate set $\mathcal{B}_t$. From equation 5,

$$\frac{w_n^{\mathrm{T}}}{w_*^{\mathrm{T}}} = \frac{\rho_n}{\rho_*} \exp\left(-\frac{S_n - S_*}{\lambda}\right) \longrightarrow 0 \qquad (\lambda \to 0)$$

for every $\boldsymbol{U}^n \neq \boldsymbol{U}^*$ with $\rho_n, \rho_* > 0$. Thus the task-driven weights concentrate on $\boldsymbol{U}^*$. If its executed prefix is non-exciting and the same condition recurs after replanning, the controller repeatedly generates interaction windows with no parameter-discriminating gradient, and identification can stall. □

Extent of the information-seeking update. Let $\mathrm{TV}(p, q) = \frac{1}{2}\|p - q\|_1$ denote total variation distance between candidate-weight distributions. From equation 10,

$$w^{\alpha} - w^{\mathrm{T}} = \alpha_t (w^{\mathrm{I}} - w^{\mathrm{T}}),$$

and therefore

$$\mathrm{TV}(w^{\alpha}, w^{\mathrm{T}}) = \alpha_t \mathrm{TV}(w^{\mathrm{I}}, w^{\mathrm{T}}) \leq \alpha_t.$$

Since w^{I} has support only on $\mathcal{A}_t$,

$$w^{\alpha}(\mathcal{B}_t \setminus \mathcal{A}_t) = (1 - \alpha_t) w^{\mathrm{T}}(\mathcal{B}_t \setminus \mathcal{A}_t).$$

If the excitation scores are constant over $\mathcal{A}_t$, then $r_n = 0$ there and

$$w_n^{\mathrm{I}} \propto \mathbf{1}[n \in \mathcal{A}_t] \rho_n e^{-S_n/\lambda},$$

so w^{I} reduces to the task-driven weights restricted and renormalized on $\mathcal{A}_t$. For an action-space update, $\sum_n w_n^{\alpha} \boldsymbol{U}^n$ remains in $\mathrm{conv}\, \mathcal{B}_t$.

Bound behind the adaptive retreat rule. Assume that, under the fixed parameter normalization of equation 11, each predicted observation $\hat{y}_{h,d}(\boldsymbol{U}; \boldsymbol{\theta})$ is L-Lipschitz in the normalized particle coordinate $\boldsymbol{z}$. With $\bar{\boldsymbol{z}} = \sum_i a_i \boldsymbol{z}_i$,

$$\begin{aligned} V_{h,d} &= \mathrm{Var}_{q_t}[\hat{y}_{h,d}(\boldsymbol{U}; \boldsymbol{\theta})] \\ &\leq \sum_i a_i \left(\hat{y}_{h,d}(\boldsymbol{U}; \boldsymbol{\theta}_i) - \hat{y}_{h,d}(\boldsymbol{U}; \mathcal{N}_{\theta}^{-1}(\bar{\boldsymbol{z}}))\right)^2 \\ &\leq L^2 \sum_i a_i \|\boldsymbol{z}_i - \bar{\boldsymbol{z}}\|^2 = L^2 \sigma_t^2. \end{aligned}$$

Using $\log(1 + x) \leq x$ in equation 7 gives

$$\mathcal{I}(\boldsymbol{U}; q_t) \leq \frac{1}{2} \sum_{h,d} \frac{L^2 \sigma_t^2}{\sigma_d^2} \leq \frac{H D_y L^2}{2 \min_d \sigma_d^2} \sigma_t^2,$$

where D_y is the observation dimension. Hence $\mathcal{I}(\boldsymbol{U}; q_t) = O(\sigma_t^2)$ as the normalized belief spread contracts.

B EXPERIMENTAL AND ADAPTATION PARAMETERS

For lifting, the center of mass is the unknown parameter and the most recently executed interval is re-evaluated under the current parameter particles. Particle weights use forgetting factor $\delta = 0.995$. PushBox and PushLetter use the most recent 18 transitions and six SVGD steps per control update. The rope uses a six-step discounted interaction window. In every task, the same parameterized surrogate F_{p} is used for model-based policy improvement and online parameter adaptation.

Table 4: **Principal experimental settings.** All compared variants of a task use the same proposal budget, surrogate, parameter belief and execution protocol.

	Rod lift	Plate lift	PushBox / PushLetter	Rope
Particles N	16	16	48 (12 for planning)	8
Candidates / horizon	512 / 0.5 s	256 / 0.5 s	1536 / 24 steps	128 / 16 steps
Temperature λ	0.1	0.1	1.0	0.05
Task-competitive fraction τ	0.1	0.1	0.1	0.1
Observation scale σ_d	3 mm, 0.03 rad	3 mm, 0.03 rad	1 cm, 0.1 rad (object), 2 cm (pusher)	25 mm
Adaptation window	executed interval	executed interval	18 transitions	6 steps, $\gamma = 0.75$
Surrogate F_{p}	ContactSDF	ContactSDF	planar ContactSDF	GVS

Particle weighting and Stein updates. The particle belief is updated from the same recent interaction data used in equation 2, but the weighting rule differs slightly across tasks.

Lifting. The most recently executed control interval forms the identification window. Particle weights accumulate evidence across attempts with exponential forgetting,

$$\log a_{i,t+1} = \delta \log a_{i,t} - \mathcal{E}_t(\boldsymbol{\theta}_i), \qquad \delta = 0.995,$$

followed by normalization. The updated weights are then used in one weighted SVGD step equation 3. Thus, both the particle locations and their relative weights are carried across lifting attempts. When the executed interval does not discriminate the hypotheses, the data-dependent part of the update vanishes.

Pushing. The uncertain positive parameters are represented in log coordinates and the most recent 18 transitions are re-evaluated under each particle. We perform six SVGD steps toward the resulting window-conditioned target using equally weighted transport particles. For policy improvement, the transported cloud is compressed to a smaller weighted particle set, whose weights reflect the relative mass of the corresponding particle clusters. Hence the larger cloud is used for parameter adaptation, while the compressed weighted belief is used for candidate evaluation.

Rope shaping. The rope update uses a six-step discounted interaction window, with positive physical parameters represented in log coordinates and rest-strain parameters represented directly. One equally weighted SVGD step moves the parameter particles using the discounted prediction error. The moved particles are then reweighted from the current window only,

$$a_{i,t+1} = (1-\rho)\frac{\exp[-\beta\mathcal{E}_t(\boldsymbol{\theta}_i)]}{\sum_j \exp[-\beta\mathcal{E}_t(\boldsymbol{\theta}_j)]} + \frac{\rho}{N}, \qquad \rho = 0.05.$$

Unlike lifting, these weights are not accumulated recursively across windows.

C Surrogate models

C.1 ContactSDF surrogate for rigid-object tasks

All rigid-object tasks use a ContactSDF-based parameterized surrogate F_{p} (Yang & Jin, 2025). The rod and plate lifting tasks use its three-dimensional form, while PushBox and PushLetter use a planar form. In all cases, the same parameterized surrogate is used for model-based policy improvement and online parameter adaptation. The task specialization lies in the manipulated geometry and in the uncertain physical parameters: the lifting tasks vary the object center of mass, whereas the pushing tasks use their corresponding planar physical parameterization.

Quasi-dynamic step. Let $\boldsymbol{q} = (\boldsymbol{q}_o, \boldsymbol{q}_m)$ collect the object configuration and the manipulation coordinates. For a manipulation command $\boldsymbol{u}$ over one step of duration h, the configuration increment is obtained from

$$\Delta\boldsymbol{q}^\star = \arg\min_{\Delta\boldsymbol{q}} \ \frac{1}{2}\Delta\boldsymbol{q}^\top Q(\boldsymbol{\theta})\Delta\boldsymbol{q} - \boldsymbol{b}(\boldsymbol{u},\boldsymbol{\theta})^\top\Delta\boldsymbol{q} \quad \text{s.t.} \quad \Delta\boldsymbol{q} \in \mathcal{K}(\boldsymbol{q}), \tag{13}$$

where $Q(\boldsymbol{\theta})$ represents the task-dependent quasi-dynamic metric, $\boldsymbol{b}(\boldsymbol{u},\boldsymbol{\theta})$ collects the manipulation input and external wrenches, and $\mathcal{K}(\boldsymbol{q})$ is the feasible set induced by the active frictional contacts. For

rod and plate lifting, $\boldsymbol{q}_m$ denotes the manipulation-bar coordinates and the uncertain center of mass enters both the object inertia and gravity wrench. For PushBox and PushLetter, the same formulation is used in the plane with the corresponding object–pusher contact geometry and task-specific physical parameters. For the lifting tasks specifically,

$$Q(\boldsymbol{\theta}) = \operatorname{diag}\left(M_o(\boldsymbol{\theta})/h^2,\, K_r\right),$$

where $M_o(\boldsymbol{\theta})$ is the object inertia under the hypothesized center of mass and K_r is the manipulation-bar stiffness.

Differentiable contact representation. Following ContactSDF, the constrained quasi-dynamic step is written as a distance projection. With

$$\boldsymbol{z} = Q^{1/2}\Delta\boldsymbol{q}, \qquad \boldsymbol{z}_0 = Q^{-1/2}\boldsymbol{b},$$

the contact update corresponds to projecting $\boldsymbol{z}_0$ onto the transformed feasible set

$$\mathcal{Z} = Q^{1/2}\mathcal{K}.$$

The contact geometry is represented by signed-distance functions, while the distance to the transformed contact-feasible set is approximated smoothly from its normalized facet violations. This yields a differentiable update

$$\boldsymbol{q}^+ = \boldsymbol{q} \oplus Q^{-1/2}\boldsymbol{z},$$

and therefore a surrogate

$$F_{\mathrm{p}}(\boldsymbol{x}, \boldsymbol{u}; \boldsymbol{\theta})$$

that is differentiable with respect to state, action and the center-of-mass parameter. This surrogate is evaluated in batch over action candidates and parameter hypotheses in both the policy-improvement and parameter-adaptation loops.

C.2 GVS SURROGATE FOR ROPE SHAPING

For Rope-U, F_{p} is a planar geometric-variable-strain (GVS) rod model following Renda et al. (2018); Boyer et al. (2021). The model is used as a compact parameterized surrogate for predicting the evolution of the rope under direct manipulation commands.

Kinematics. The rope of length ℓ is parameterized by arc length $s \in [0, \ell]$ and a strain field $\boldsymbol{\xi}(s) \in \mathbb{R}^6$. We retain the planar strain components and use a piecewise-constant representation over E elements,

$$\boldsymbol{\xi}(s) = \boldsymbol{\xi}^{\circ}(s; \boldsymbol{\theta}) + B(s)\boldsymbol{q},$$

where $\boldsymbol{\xi}^{\circ}(s; \boldsymbol{\theta})$ denotes the parameterized rest strain. The frame along the rope is obtained by integrating the strain field, which in the piecewise-constant representation gives

$$g(s) = g_0 \prod_{e<e(s)} \exp\left(\widehat{\boldsymbol{\xi}}_e\, \Delta s\right) \exp\left(\widehat{\boldsymbol{\xi}}_{e(s)}(s - s_{e(s)})\right).$$

The base pose $g_0 \in SE(2)$ is free, so neither end of the rope is anchored. Material-point positions $\boldsymbol{p}_k$ and their Jacobians

$$J_k = \frac{\partial \boldsymbol{p}_k}{\partial \boldsymbol{z}}$$

follow from the integrated strain representation.

Implicit dynamics step. Let $\boldsymbol{z}$ collect the strain coordinates and the free planar base pose. One step of duration h is obtained from the incremental potential

$$\begin{aligned}\Delta\boldsymbol{z}^{\star} = \arg\min_{\Delta\boldsymbol{z}} \; & \frac{1}{2}\Delta\boldsymbol{z}^{\top}\left(\frac{\mathcal{M}(\boldsymbol{\theta})}{h^2} + K(\boldsymbol{\theta}) + \frac{D}{h}\right)\Delta\boldsymbol{z} \\ & - \Delta\boldsymbol{z}^{\top}\left(\frac{\mathcal{M}(\boldsymbol{\theta})}{h}\dot{\boldsymbol{z}} - K(\boldsymbol{\theta})(\boldsymbol{q} - \boldsymbol{q}^{\circ})\right) \\ & + \sum_k \Psi_{\mu}\left(J_k\Delta\boldsymbol{z}\right) + \Phi_{\boldsymbol{u}}(\Delta\boldsymbol{z}),\end{aligned} \tag{14}$$

where $\mathcal{M}(\boldsymbol{\theta})$ is the parameterized mass metric, $K(\boldsymbol{\theta})$ the strain stiffness, D the strain damping, and Ψ_μ the distributed interaction with the supporting surface. The term $\Phi_{\boldsymbol{u}}$ represents the direct manipulation command applied to the selected rope material points; it is an input term in the surrogate, not a separate actuator-dynamics model.

The uncertain parameter vector $\boldsymbol{\theta}$ changes the task-relevant physical properties of the rope, including stiffness, mass, surface interaction and rest strain. Consequently, the same candidate manipulation command can produce different predicted shapes under different parameter hypotheses. These hypothesis-dependent rollouts are used both to evaluate candidate actions and to update the online parameter belief from observed rope motion.

D Additional experimental details

Lifting tasks. The rod and plate rest on two supports and are lifted from below by a position-controlled bar. The unknown parameter is the center-of-mass offset: one-dimensional along the rod and two-dimensional in the plate plane. The rod uses true offsets of ± 4 cm, while the plate uses eight directions on a 4.5 cm-radius circle. The observations used for online adaptation are the object pose and bar position. A trial is successful when the object reaches the target height within 1 cm, its tilt is below 3°, and contact with the supports has been broken. Plate trials are terminated early if the tilt exceeds 15°.

Rope shaping. Rope-U asks the controller to deform a free planar rope toward a prescribed U-shaped target using the GVS surrogate of Appendix C.2. A manipulation command is

$$\boldsymbol{u} = \big((s_1, \varphi_1, d_1), (s_2, \varphi_2, d_2)\big),$$

where s_i selects a material location, φ_i a planar manipulation direction, and d_i the commanded displacement. The inputs act directly on the selected rope points; no separate actuator dynamics are modeled. The uncertain parameters describe task-relevant stiffness, mass, surface interaction, and rest-strain properties. Performance is measured by the terminal RMSE between 49 ordered rope points and the target shape.

Learned action proposals. The learned proposals used for the lifting experiments are trained from successful trajectories generated by an oracle model-based planner with the true center of mass available during data generation, and are frozen during all reported test-time experiments. The flow policies are trained by rectified flow matching on action-sequence targets obtained from the highest-weight candidates of the planner's final policy-improvement iteration. The randomized flow is conditioned only on the current state, whereas the parameter-conditioned flow additionally receives the center-of-mass hypothesis. The Gaussian actor is initialized by behaviour cloning on successful demonstrations and subsequently refined with PPO; at test time it is used only to provide the mean of the Gaussian action proposal.

E Additional experimental results

E.1 Lifting diagnostics

Table 5 breaks down the plate results of Table 3 by center-of-mass direction. The improvement from DAPI is distributed across the evaluated directions rather than being concentrated in a single load configuration.

Episode diagnostics. Figures 4 and 5 illustrate the closed-loop adaptation mechanism for individual lifting groups. In the plate example, belief-only adaptation does not generate parameter-discriminating object motion and its center-of-mass belief remains essentially unchanged. With DAPI, informative lifts contract the belief before a subsequent level lift. In the rod example, the belief-only particles evolve primarily through the data-independent prior and repulsion components of the SVGD update when the executed interaction is non-exciting. Once DAPI produces an informative lift, the weighted center-of-mass estimate moves toward the true offset.

Table 5: **Plate lifting by center-of-mass direction.** Successful seeds out of three for each direction; attempts last 1 s.

Proposal	0°	45°	90°	135°	180°	225°	270°	315°	total
Sampling, belief-only	0	0	0	0	0	0	0	0	0/24
Sampling, DAPI	2	2	2	2	1	0	3	2	14/24
Gaussian actor, belief-only	0	0	0	0	0	0	0	0	0/24
Gaussian actor, DAPI	0	2	1	2	0	0	0	2	7/24
Randomised flow, belief-only	1	2	2	2	2	1	2	2	14/24
Randomised flow, DAPI	2	2	2	2	2	1	3	3	17/24
Conditioned flow, belief-only	1	2	2	2	2	0	2	2	13/24
Conditioned flow, DAPI	2	3	2	2	3	1	2	3	18/24

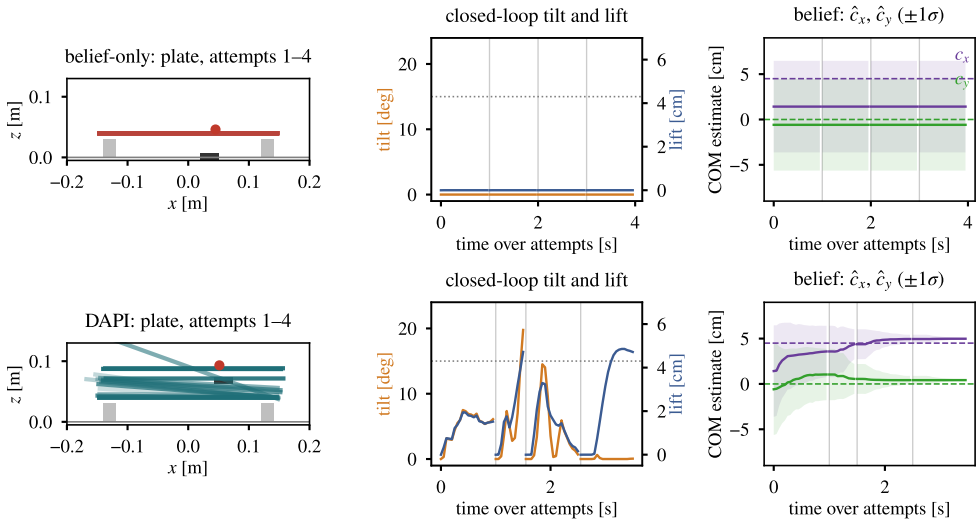


Figure 4: **Episode diagnostic, plate lift** (0°, seed 0, Gaussian sampling proposal). Left: plate configuration at five instants of each attempt (darker indicates later time), with the manipulation bar and true center of mass shown at the end. Middle: tilt and lift over concatenated attempts; vertical lines mark attempt boundaries and the dotted line the 15° termination threshold. Right: weighted center-of-mass estimate with $\pm 1\sigma$ belief spread; dashed lines denote the true values.

Figure 6 complements Figure 3 with the learned proposal classes. The same qualitative behaviour is observed: informative interaction contracts the physical belief and reduces the contribution of the information-seeking channel, although the rate of retreat depends on the proposal and task.

Figure 7 provides representative visual examples of the corresponding lifting behaviour.

E.2 PushLetter

PushLetter extends the planar pushing experiment from a box to the V-, D-, and R-shaped compound objects of the Letters-Planar-Pushing geometry. The same manipulation protocol and mismatch construction are used as for PushBox, while the planar ContactSDF surrogate represents each letter by its compound convex geometry. The more irregular shapes introduce stronger rotational and geometric effects than the box, which are only approximately represented by the reduced planar surrogate. This provides a more demanding model-mismatch setting and is consistent with the smaller gain from online adaptation observed for PushLetter in Table 2.

E.3 Rope shaping: limitation of spread-based retreat

Rope-U tests the adaptation and retreat mechanisms in a higher-dimensional deformable setting. Over 24 paired seeds, receding-window belief-only adaptation reduces the median terminal shape error

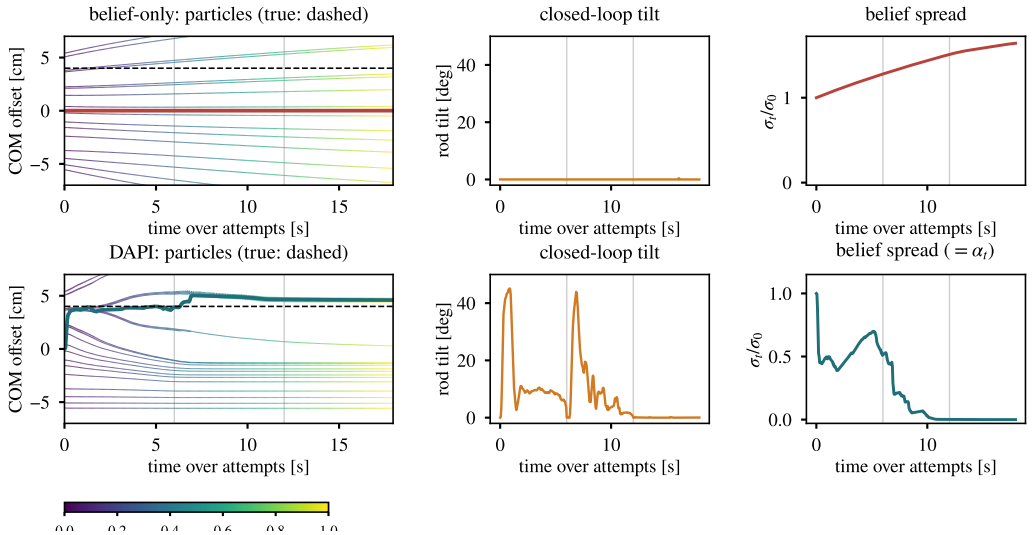


Figure 5: **Particle evolution, rod lift** (COM +4 cm, seed 0, Gaussian sampling proposal). Left: center-of-mass particles, their weights, weighted mean, and true value. Middle: rod tilt. Right: the normalized belief spread that determines the information-seeking weight α_t.

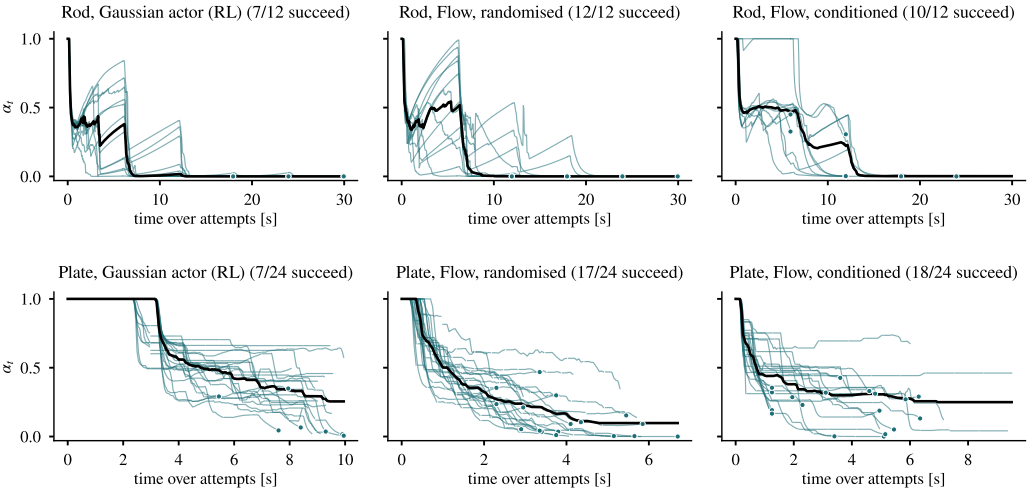


Figure 6: **Information-seeking weight α_t for the learned proposal classes.** Rod: 12 groups; plate: 24 direction–seed pairs, with the same layout as Figure 3. On successful rod groups, α_t approaches zero before or around the final level lift. Plate groups can succeed while α_t remains nonzero because their shorter attempts provide less time for belief contraction.

from 80.4 mm with the nominal model to 40.1 mm and improves over the nominal model in 23/24 seeds. Online parameter adaptation therefore remains effective despite the nine-dimensional physical uncertainty.

The spread-based retreat behaves differently. With DAPI, the normalized belief spread contracts only weakly, with a final-to-initial ratio of 0.86. Consequently, α_t never falls below 0.1 and has an episode mean of 0.83, while the median terminal error increases to 63.0 mm. The issue is therefore not that the online parameter update ceases to be useful; rather, the current retreat rule uses belief spread alone and cannot distinguish uncertainty that remains reducible by the available actions from uncertainty that stays broad despite useful task interaction. This is the limitation discussed in Section 6. The persistent information-seeking weight is shown in Figure 8, and the complete quantitative comparison is reported in Table 6.

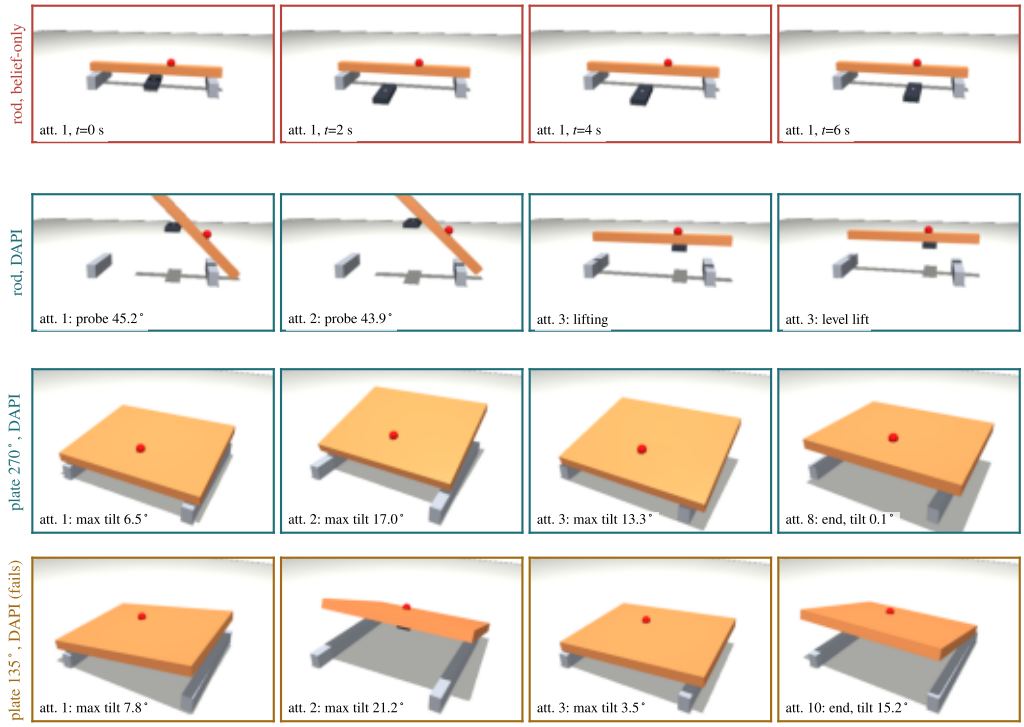


Figure 7: **Representative lifting sequences** (MuJoCo renders of recorded attempts; red: true center of mass). For the rod example (COM $+4$ cm, seed 0), belief-only adaptation keeps the manipulation bar under the object without producing an informative lift, whereas DAPI first generates parameter-discriminating motion and later lifts the rod level. The successful plate example similarly shows informative interactions followed by a level lift, while the final row shows a representative failure.

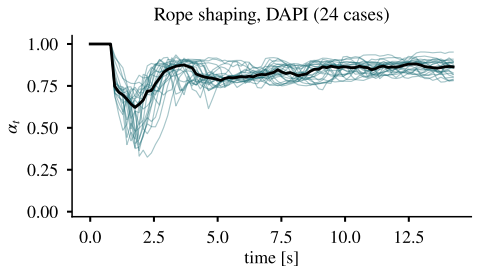


Figure 8: **Information-seeking weight on Rope-U.** α_t over 24 seeds; black denotes the median. The high-dimensional belief remains broad, so the spread-based retreat does not close the information-seeking channel.

F Computational cost

All controllers operate online in the sense that model-based policy improvement and parameter adaptation are recomputed after every executed control step, but the present implementation is not real time.

For the pushing comparison in Table 2, the recursive DuSt-style parameter update requires approximately 2.0 s per control step in our implementation, whereas the receding-window update used by DAPI requires approximately 0.39 s. Thus the improvement of the receding-window update is not obtained by allocating it a larger computational budget.

Table 6: **Rope-U terminal shape error** over 24 paired seeds [mm].

Method	median	mean
Nominal model	80.4	80.7
True parameters (oracle)	48.0	46.8
Recursive Stein dynamics inference	42.6	44.2
Recursive, with DuSt Stein MPC	74.0	81.3
Receding-window adaptation, belief-only	**40.1**	**43.5**
DAPI	63.0	59.6