ADAPTATION AND PROBING UNDER HIDDEN DYNAMICS: EVIDENCE FROM PLANAR PUSHING

Anonymous authors
Paper under double-blind review

ABSTRACT

Information about hidden dynamics can improve manipulation. Whether a dedicated probe adds to what ordinary task interaction already reveals is a separate question. We study this question in a controlled Push-T variant with an unobserved effective support-friction coefficient that spans more than a decade. Using diffusion policies with a shared backbone and training data, we compare true-parameter access, online estimation from task histories, history conditioning and per-tier specialists, and we measure the room left for a probe as the paired difference between a controller that knows the parameter from its first action and the same controller estimating it along the way. On held-out initial states, true-parameter access improves success by +5.5 percentage points on the evaluated training tiers and by +35.0 points at $\mu = 2.00$, above the training range; the gain appears as closer approaches to the goal at low friction and as completed episodes at high friction. An estimator trained on ordinary task histories locates the friction within 64 steps, and a policy fed its estimate recovers the success-rate gain within the training range (recovered fraction 1.02 [0.76, 1.35]) and 0.58 of it at $\mu = 2.00$. Parameter-input interventions change the amplitude of the first commanded action chunk and the timing of contact, and an incorrect value impairs success more than masking the input. The controller with the true parameter and the controller with the online estimate differ by 0.1 success points on the training tiers, with an interval that includes zero. These results establish ordinary task interaction as a strong adaptation baseline, and they set the reference against which dedicated probing should be evaluated: a controller that adapts during execution.

1 INTRODUCTION

A pushing controller has to account for how the object responds during contact and how it moves after release. When the surface is unknown, the controller can rely on behaviour learned across surfaces, infer the dynamics from its recent interaction, or spend actions on reducing its uncertainty before it proceeds. These options correspond to different designs: policies that condition on interaction history, policies that use an explicit online estimate, and policies that receive information from a dedicated exploration phase (Kumar et al., 2021; Yu et al., 2017; Zhou et al., 2019; Memmel et al., 2024). Whether a dedicated probe adds to what ordinary task interaction already reveals is a separate question, and the choice between these designs depends on the answer.

We study a controlled variant of Push-T (Florence et al., 2022; Chi et al., 2025). It keeps the T-shaped block, the fixed goal, the keypoint observation and 10 Hz control. Inside the modified simulator we vary a hidden effective support-friction coefficient μ with every other setting fixed. Evaluated unchanged, the released Push-T policy loses most of its success as the friction falls. The same commanded push moves the block three and a half times as far at the low end of our range as in the original environment (Figure 1). The policies we train for the friction variant share a diffusion-policy backbone, data and training budget and differ in their information pathway: no friction channel, the true value, an estimate from the agent's own history, or an encoding of that history. Per-tier specialists measure what a network of this size reaches when the friction is fixed and known.

The analysis runs from the task to the controller: what the hidden friction does to a trained skill (Section 4), how much the parameter improves a trained policy and where (Section 5), how much of that gain ordinary interaction exposes and a policy recovers (Section 6), what the input changes in

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.

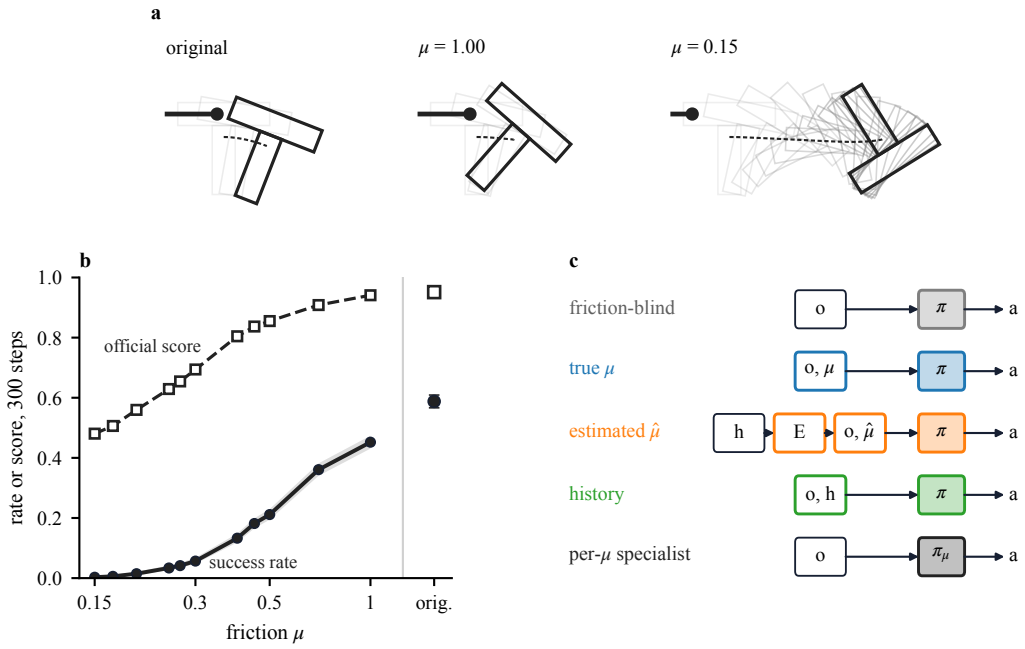

Figure 1: **Hidden friction in Push-T.** (a) The same commanded pusher motion, 80 px/s for 1 s and then a retreat, in the original environment, at $\mu = 1.00$ and at $\mu = 0.15$; the block moves 57.5, 48.5 and 200 px, of which 0.6, 5.5 and 178 px after the pusher has let go. (b) Success within 300 steps and official score of the released DP-CNN checkpoint on the test set as a function of μ; the points at the right are the original environment. (c) The five information pathways compared: the friction-blind policy acts on the observation o alone; the friction-input policy also receives the true μ; the estimator-conditioned policy receives $\hat{\mu}$, read from its own history h by the estimator E; the history-conditioned policies receive h itself; one specialist per tier is trained at a fixed μ.

the actions (Section 7), and how much room is left for a dedicated probe, in Push-T and in a delivery task with no later correction (Section 8). On held-out states, access to the true friction improves success by +5.5 points on the training tiers and by +35.0 points above them. A policy that estimates the friction along the way comes within 0.1 success points of one that knows it from the first step.

2 Related work

2.1 Task-relevant information and adaptation from interaction history

The value of information lies in the decisions it improves (Howard, 1966). Information gathering can stop once the remaining hypotheses support a common decision (Javdani et al., 2014). In control, inputs both move the state and inform the estimator (Feldbaum, 1960; Bar-Shalom & Tse, 1974), and identification is judged by the control objective it serves (Gevers, 2005). We measure the benefit of parameter access for trained policies and the cost of errors in the parameter input.

Interaction history provides a route to adaptation when the parameter is hidden from direct observation. Rapid Motor Adaptation learns a history-based encoding of environmental conditions (Kumar et al., 2021), while universal policies with online system identification use explicit parameter estimates to condition control (Yu et al., 2017). Meta-reinforcement learning learns adaptation through recurrent state or inferred task beliefs (Duan et al., 2016; Zintgraf et al., 2020). These approaches motivate our comparison of history-conditioned and estimator-conditioned policies, with per-tier specialists serving as parameter-specific reference controllers. A friction-blind policy trained on a mixture of tiers serves as a dynamics-randomization baseline (Peng et al., 2018).

Imitation under information asymmetry shows that the actions of a privileged expert can be inappropriate for a learner with restricted observations (Ortega et al., 2021; Walsman et al., 2023). Our

parameter-informed experts demonstrate the task with the friction known, and we measure adaptation from the resulting histories separately.

2.2 Information from task execution and dedicated probing

Interactive perception uses action to generate sensory evidence that passive observation may not provide (Bohg et al., 2017). Dedicated exploration methods differ in what makes this evidence useful. Environment-Probing Interaction policies learn a preliminary interaction phase whose trajectory is encoded for a task policy (Zhou et al., 2019). ASID learns informative exploration for identifying physical parameters, refining a simulator, and subsequently synthesizing a task controller (Memmel et al., 2024). Push to Know selects exploratory pushes using visual and tactile feedback to improve object-parameter inference (Dutta et al., 2023). Task-oriented approaches connect exploration more directly to subsequent control: Liang et al. (2020) optimize exploration for downstream model-based task performance, while Poke and Strike uses controller sensitivity to define exploration rewards and estimation uncertainty to determine when to execute the task (Aoyama et al., 2025).

The timing of information acquisition is particularly important when a task includes an irreversible action. In Poke and Strike, for example, striking a puck into a goal outside the robot's reachable workspace leaves no opportunity to retry (Aoyama et al., 2025). Continuous closed-loop pushing offers a different opportunity: observations generated during task execution can inform later actions. The value of a dedicated probe is the gain it provides beyond the information that the task's own actions already provide. We measure that room as the difference between a controller that knows the parameter from its first action and the same controller estimating the parameter along the way, and we set it beside a one-push task in which no later action can correct the first.

Human studies separate the production of evidence from its interpretation (Gibson, 1988; Lederman & Klatzky, 1987; Bramley et al., 2018; Karimpur et al., 2024). We keep the two apart by measuring the estimator's error and the estimator-conditioned policy's performance separately.

2.3 Physical-context adaptation and controlled Push-T evaluation

Push-T is a contact-rich planar benchmark of implicit behavioral cloning and diffusion-policy research (Florence et al., 2022; Chi et al., 2025), and the value of its state observations has been measured by withholding them (Krishna et al., 2025). Pushing under uncertain dynamics has been studied with vision-language priors fused with online estimates (Wang et al., 2026) and with context-conditioned policies and estimator supervision (Iannotta et al., 2026). We vary one effective support friction inside a fixed simulator and compare parameter-conditioned, history-conditioned and friction-blind controllers on one task.

3 Testbed, behavioural measures and evaluation

3.1 Push-T with a hidden effective friction

Push-T (Florence et al., 2022; Chi et al., 2025) asks a circular pusher to move a T-shaped block on a 512×512 canvas into a fixed goal pose. The policy observes nine block keypoints and the pusher centre and commands the pusher's absolute target at 10 Hz; the released diffusion policy executes eight of its sixteen predicted actions per call. In the original simulator the pusher is kinematic and reaches wherever it is sent, and the block loses its whole velocity within one substep of losing contact, so the block stays where each push leaves it (Appendix A).

We keep the task, the observation, the action and the goal and change two things in the physics. The block receives support friction from the table: every surface element of the block feels a friction force of magnitude $\mu\, m\, g_{\text{eff}}\, w_j$ opposite to its sliding velocity, smoothed near zero speed, and the resulting force and torque stop the block without ever reversing its motion; it is the only dissipation. For a block in free translation the deceleration is μg_{eff} with $g_{\text{eff}} = 100\,\text{px/s}^2$, independent of the mass, and a block released at speed v coasts

$$d(v, \mu) = \frac{v^2}{2\,\mu\, g_{\text{eff}}}. \tag{1}$$

g_{eff} is about one hundredth of physical gravity at a millimetre-per-pixel scale, so μ is an *effective* coefficient: the range 0.15–2.00 spans coasting distances from tens of pixels to under one pixel at the pushing speeds of our policies, and the original environment sits at the high-friction end. The coefficient is fixed within an episode and absent from the observation. The pusher becomes a 10 kg dynamic body under the same PD law with a force limit and a speed cap of 180 px/s, so every policy moves the pusher at most 18 px per control step. The friction sweep varies μ with this executor fixed; the difference between the original environment and ours at $\mu = 1.00$ is a separate measurement (Section 4). Simulator checks, including bit-for-bit replay and the free-sliding deceleration against Eq. (1), are in Appendix A.

3.2 Friction tiers, initial states and collection

Friction takes values on a logarithmic grid of 18 tiers from 0.15 to 2.00. Policies and the estimator are trained on eleven tiers between 0.15 and 1.00; four tiers inside that range (0.17, 0.27, 0.45, 0.85) and one above it (2.00) are held out from both. The test set is evaluated on thirteen tiers: eight of the training tiers (0.15, 0.20, 0.25, 0.30, 0.40, 0.50, 0.70, 1.00), the four held-out interpolation tiers and 2.00; group averages weight tiers equally. Initial states follow the official reset distribution in three disjoint sets: a pool P of 7,250 states for data collection and estimator training, a selection set V of 128 states for every choice made during development, and a test set T of 512 states for every number reported. On P, ten to eleven source policies (the released checkpoint with a per-tier displacement multiplier and earlier-round policies of the designs below) were rolled out three times per state and tier: 1,989,753 episodes stored step by step, of which 803,305 succeed within 300 steps. The collection is the training data of the policies and the estimator, and the population for the behavioural description of Section 4 (Appendix A).

3.3 Policies with different access to the friction

All policies share the released diffusion-policy backbone, a one-dimensional U-Net that predicts 16 actions from two observation steps and executes eight, and all are fine-tuned from the released checkpoint on the same collected data with the same normaliser and optimiser for 80,000 steps; the specialists' checkpoints are chosen on V between 20,000 and 120,000 steps. They differ in what they receive about the friction. The *friction-blind* policy receives the observation alone and is trained on the union of the training tiers, the dynamics-randomisation baseline. The *friction-input* policy receives the true $\log \mu$ through a zero-initialised adapter that adds one token to the network's global condition, so its training starts from the friction-blind solution. The *estimator-conditioned* policy is the friction-input network deployed with the estimate of Section 3.4 in place of the true value, recomputed at every replanning call, with the prior before the first estimate. The *history-conditioned* policies receive no friction value and read their own recent transitions into the same conditioning slot, through a recurrent encoder over the last 128 steps or a window of the last eight. One *specialist* per training tier is trained on that tier alone and marks what a network of this size reaches when the friction is fixed and known throughout training. Training seeds: three for the friction-blind, friction-input and estimator-conditioned policies, two for the window policy, one for the others.

3.4 Friction estimator

A single-layer GRU reads, at every step, a 19-dimensional feature vector built from what the policy itself sees, the observation, its commanded target and the pusher position before and after the action, and after N steps outputs $\hat{y}_N$ for $y = \log_{10} \mu$; the loss sums a squared error and a tier-classification term over $N \in \{1, 2, 4, \ldots, 128\}$ (Appendix C). It is trained on ordinary task histories of the collection from the eleven training tiers. The deployed readout clips $\hat{y}_N$ to $[-1.0,\ 0.5]$, a window wider than the training range. Its error is reported as the mean absolute log error, on held-out collection episodes and on the test episodes of the estimator-conditioned policy, in which the estimate itself drives the controller and no condition receives the true μ.

3.5 BEHAVIOURAL MEASURES

The goal is fixed, so the initial state sets the task. Every stored episode is described at two levels. The *path level* looks at the block: the corners of its centre-of-mass path, the share of its rotation completed by mid-translation, and the direction of each translation step in the block's own frame. The *execution level* looks at the pusher: a *push* is a maximal run of contact whose contact point stays within 30 px on the block, an episode is a sequence of pushes separated by repositioning, and the *fine-alignment phase* begins at the first step at which coverage reaches 0.8. The same measures apply to the 90 human demonstrations of the official training set and to every policy (Appendix B).

3.6 INTERVENTIONS, THE ROOM FOR A PROBE, AND THE DELIVERY VARIANT

Content interventions. On the friction-input checkpoint trained with masking probability 0.1, the μ channel receives, on the same test states, the true value, no value (availability flag zero, in distribution for this checkpoint), or a wrong value: rule A tells the policy 1.00 when the true μ is at most 0.40 and 0.20 otherwise; rule B tells it 1.00 and 0.15 at the same split. The same checkpoint given the true value is the reference.

Room for a probe. A probe is worth at most what the controller gains from knowing the friction before its first action instead of learning it along the way. We measure that ceiling as the paired difference, true input minus online estimate, between the friction-input network given the true value from the first step and the same network fed the online estimate, on the same states, tiers and noise.

Delivery variant. A separate one-push task on the same physics: the pusher drives the block along a line up to a release line, the block coasts, and the episode succeeds when the block's centre stops within $\pm w$ px of a target at distance D beyond the release point, for $D \in \{60, 80, 100\}$ and $w \in \{5, 10, 20, 40\}$ px. Three analytic controllers share the geometry: one aims with the true μ, one uses the single fixed speed with the best mean success over the training tiers, and one first pushes the block a few pixels along the direction of travel, reads $\hat{\mu}$ from the coast through Eq. (1), and then aims with $\hat{\mu}$; its probing steps are not charged. Under Eq. (1) the band condition is $|\hat{\mu}/\mu - 1| \le w/D$ (Appendix D).

3.7 EVALUATION

Let $c_e(t)$ be the coverage after action t of episode e. With budget B,

$$S_e^{(B)} = \mathbb{1}\big[\exists\, t \le B :\ c_e(t) > 0.95\big], \qquad R_e^{(B)} = \max_{t \le B} \min\Big(\frac{c_e(t)}{0.95},\, 1\Big), \tag{2}$$

the binary success (the official termination rule: coverage above 0.95 within the budget) and the official continuous score. We report both for every comparison, with $B = 300$, the budget of the original evaluation, as the main budget; every evaluation episode runs to 400 steps and is stored step by step, so both criteria are available at any budget. Every condition is run once per initial state and tier; the noise seed depends on the state, the tier and the sample index and not on the policy, so conditions are paired episode by episode. Outcomes are averaged within an initial state over training seeds and then across states. A comparison between two conditions is the paired difference over states with a 95% percentile interval from 2,000 bootstrap resamples; for conditions with several training seeds the bootstrap resamples states and seeds together.

4 WHAT HIDDEN FRICTION DOES TO A TRAINED PUSHING POLICY

Figure 1b evaluates the released Push-T checkpoint, unchanged, on the test states at every friction tier for 300 steps. The executor and the dissipation law together cost the checkpoint about a quarter of its success between the original environment and ours at $\mu = 1.00$, from 58.8% to 45.2%, while the official score is nearly unchanged. Below $\mu = 1.00$ success falls with the friction, to 0.3% at $\mu = 0.15$, and five further public checkpoints follow the same course, every one below 3.5% at $\mu \le 0.20$ (Appendix C). A longer budget does not recover the low-friction failures: in the collection, where every episode runs to 400 steps, the extra hundred steps rescue 4% of this checkpoint's failed

Table 1: **Main comparisons on the test set at 300 steps.** Paired differences from the friction-blind policy over initial states on the eight evaluated training tiers, the four held-out interpolation tiers and the extrapolation tier $\mu = 2.00$; success in percentage points, score in score units; brackets are 95% intervals. The last row compares the specialists with the friction-input policy on the training tiers.

	Success, train	Score, train	Success, held-out	Success, $\mu = 2.00$
Friction-input	+5.5 [+4.0, +7.2]	+0.033 [+0.027, +0.040]	+4.7 [+2.4, +7.0]	+35.0 [+30.5, +39.5]
Estimator-cond. (E_1)	+5.6 [+4.0, +7.4]	+0.030 [+0.023, +0.036]	+5.1 [+3.0, +7.1]	+20.4 [+15.9, +24.9]
History, window 8	+2.5 [+0.6, +4.5]	+0.010 [+0.004, +0.016]	+2.8 [+0.4, +5.0]	+4.6 [+0.2, +8.9]
History, recurrent	+0.1 [−1.6, +1.9]	+0.007 [+0.001, +0.014]	+0.1 [−2.1, +2.4]	+5.6 [+0.5, +10.5]
Specialists − input	+2.2 [+0.5, +3.7]	+0.007 [+0.002, +0.014]	–	–

episodes at $\mu = 0.15$ against 28% at $\mu = 1.00$. Figure 1a shows the physical effect behind this drop. The same commanded push moves the block 57.5 px in the original environment and 200 px at $\mu = 0.15$, most of them after the pusher has let go. The released policy was trained where the block stops close to where the pusher releases it; at low friction every release leaves the block in a position its training never produced.

Paths and pushes in the collection. The collection and the 90 human demonstrations are measured with the same tools. At the path level the sources agree: the median block path has 2 corners for every source, and translation runs along the block's bar and stem axes. At the execution level they differ: a demonstration has 3 pushes of 17 steps at the median, and the released checkpoint at $\mu = 1.00$ makes 6 pushes of 11 steps. Figure A1 in Appendix B shows the paths, the translation directions and the fine-alignment share per tier.

5 What knowing the friction is worth

We first compare the friction-input policy, given the true μ, with the friction-blind policy on the same test states, tiers and noise samples. Figure 2a shows both policies' success per tier and Table 1 the paired differences. Over the eight training tiers, access to the true value improves success by +5.5 points. On the four held-out interpolation tiers the gain is +4.7. At $\mu = 2.00$, above every training tier, the gain is +35.0. Under the official score the training-tier difference is +0.033. The friction-input policy scores 0.918 on the training tiers and 0.974 at $\mu = 1.00$, above the released policy's 0.951 in its own environment. Its success stays at 32.5%.

The gain from the true parameter is concentrated at the two ends of the friction range and takes a different form at each end (Figure 2b). At high friction it appears as completed episodes: the gap in success is +19.8 points at $\mu = 1.00$, where the friction-blind policy has stopped improving and the friction-input policy keeps rising. At low friction it appears as closer approaches to the goal: the gap in success at $\mu = 0.15$ is +5.9 points, while the gap in score is +0.105, the largest of any tier. Between 0.27 and 0.50 the two policies perform comparably under both criteria. Per tier, the success interval excludes zero on 5 of the 13 tiers and the score interval on 9.

The specialists, one network per tier with the friction fixed throughout training, reach 34.7% on the training tiers against 32.5% for the friction-input policy, a difference of +2.2 points. The specialists' advantage is concentrated at the two lowest tiers, +5.2 points at 0.15 and +7.1 at 0.20, and elsewhere the interval includes zero, so a single network with a friction input comes within two success points of eight networks trained for their tiers. At 400 steps the training-tier difference is +7.9 points; both criteria and both budgets appear in Appendix C.

6 Recovering the friction from ordinary interaction

Estimator E_1 was trained on ordinary task histories. Figure 3a shows its error during the estimator-conditioned policy's own test episodes, where no condition receives the true μ. On the training tiers the mean absolute log error falls from 0.205 at step 16 to 0.096 at step 32 and 0.026 at step 64. On the held-out interpolation tiers it is 0.033 at step 64 against a prior of 0.230. At $\mu = 2.00$ it stays at 0.287 from step 64 onward, which places the estimate near the top of the training range. Inside

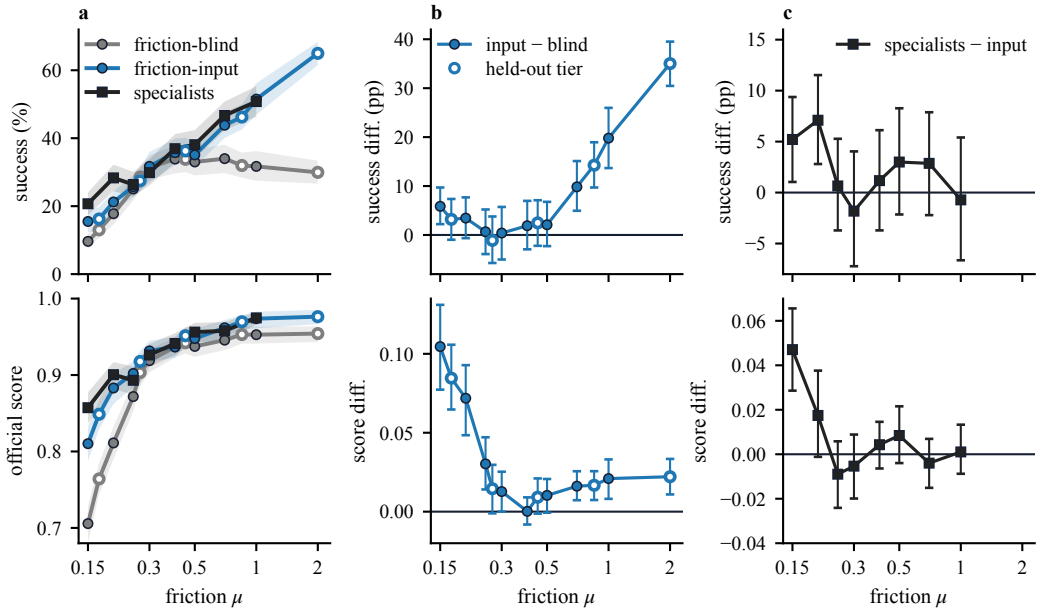

Figure 2: **What knowing the friction is worth, and where.** (a) Success within 300 steps (top) and official score (bottom) per tier on the test set for the friction-blind and friction-input policies and the specialists. (b) Friction-input minus friction-blind per tier, in success points (top) and score (bottom). (c) Specialists minus friction-input on the training tiers. Open markers are held-out tiers; bands and bars are 95% intervals.

the training range, the estimate after sixty-four ordinary pushing steps is therefore within a tenth of a decade of the true friction.

When the friction-input controller receives that estimate at every replanning call, it recovers the success-rate gain of the true input (Table 1). On the training tiers the estimator-conditioned policy gains +5.6 points over the friction-blind policy, a recovered fraction of 1.02 [0.76, 1.35] under success and 0.88 [0.77, 1.01] under the score. On the held-out interpolation tiers the fractions are 1.08 and 0.72. The difference from the true-input policy on the training tiers is +0.1 points. At $\mu = 2.00$ the estimate recovers 0.58 [0.47, 0.70] of the gain, and the estimator-conditioned policy is -14.6 points below the true-input policy there. The shortfall in this one condition reflects the upper limit of the estimator's training range.

The history-conditioned policies, which read their own transitions without an explicit estimate, recover less: the window-8 policy recovers about half of the gain and the recurrent policy almost none (Table 1, Figure 3b).

7 What the input changes

Action scale and content. On the test set the first action chunk is issued from the same observation at every tier, so its commanded amplitude isolates what the input changes before the block has moved (Figure 4a). The friction-input policy's first-chunk amplitude grows from 110 px at $\mu = 0.15$ to 153 px at 2.00, a slope of +32.0 px per decade of μ. The friction-blind policy issues the same first chunk at every tier, and so does the estimator-conditioned policy, whose first chunk is issued on the prior. During contact the friction-blind policy moves the pusher farther per step at low friction than at high, and the friction-input policy does the opposite, a slope difference of +2.6 px per step per decade. The interventions on the masking-trained checkpoint show that the policy's actions follow the value supplied in the channel (Figure 4c; the follow index per tier is in Appendix C, Figure A8). When a low-friction state is labelled 1.00, the first-chunk amplitude moves to the value the policy issues at $\mu = 1.00$: the follow index, 0 for the true value and 1 for the told value, is 1.05 [0.95, 1.17], and 0.99 when a high-friction state is labelled 0.20 or 0.15. On the training tiers a wrong value costs

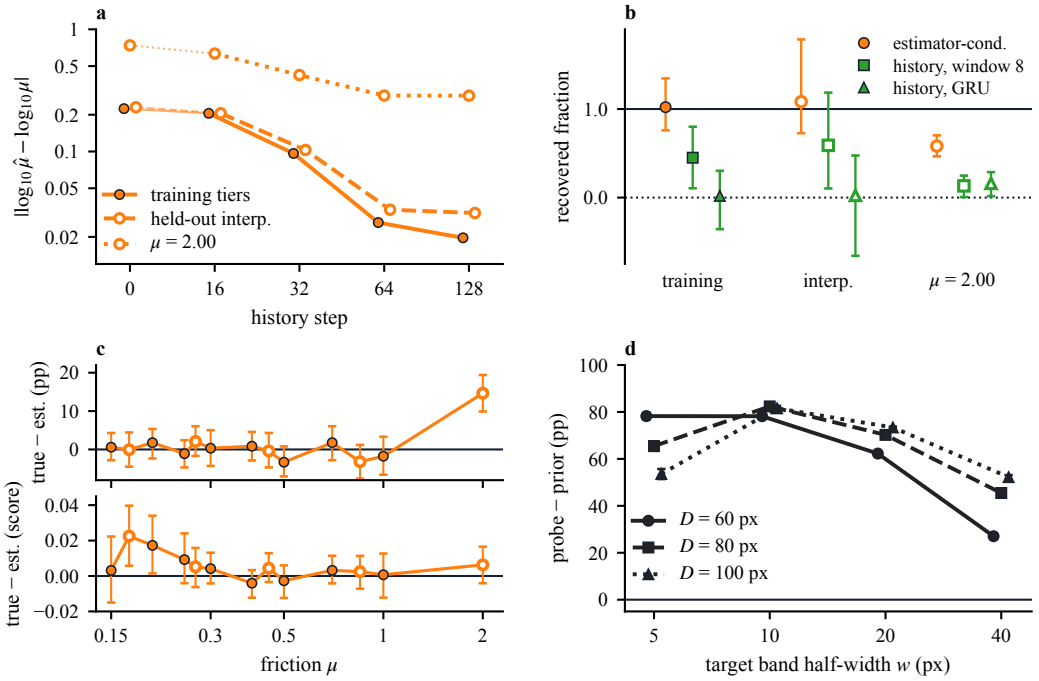

Figure 3: **Reading the friction from history, and the room left for a probe.** (a) Error of the online estimate, $|\log_{10}\hat{\mu} - \log_{10}\mu|$, by the step at which it was made, during the estimator-conditioned policy's own test episodes, per tier group; step 0 is the prior. (b) Fraction of the true-input gain in success recovered by the estimator-conditioned, window-8 and recurrent policies, per tier group. (c) True input minus online estimate on the same checkpoints, per tier, in success points (top) and score (bottom). (d) Delivery task: probe controller minus prior controller by band half-width w, for three distances D.

-16.4 to -18.8 success points against the true value, and the masked input costs -2.7. A wrong value therefore impairs success more than masking the input, by -13.7 to -16.0 points. The costs grow with the friction.

Contact timing. The two policies change their contact timing in opposite directions (Figure 4b). On the test set, in successful episodes, the friction-input policy's mean push length shortens from 26.9 steps at the lowest tier to 14.5 at the highest, and the friction-blind policy's lengthens from 17.5 to 28.6; the two cross near $\mu = 0.40$–0.45. Under a wrong value the push length follows the told value: at $\mu = 0.15$ told 1.00, the same checkpoint pushes for 10.6 steps, against 27.7 with the true value. The choice of face and push direction changes little with the friction or the input. The length and number of pushes change with both (Appendix B).

8 The room left for a probe

The most a probe can add to the friction-input controller is the difference between knowing the friction from the first step and estimating it along the way. Figure 3c shows that difference per tier. On the training tiers the two differ by 0.1 success points, and on the held-out interpolation tiers by 0.4, with intervals that include zero (Figure 3c). Under the score the true input is ahead by at most 0.01 (+0.004 [0.000, +0.008] on the training tiers, +0.009 [+0.003, +0.015] on the held-out tiers). Inside the training range the room is below two success points at every tier. At $\mu = 2.00$ the true input is ahead by +14.6 points [+9.9, +19.4]; there the estimate stops at the top of the estimator's training range, and the remedy is a wider training range for the estimator. Inside the training range, ordinary task actions reveal the friction early enough to leave a dedicated probe no measurable gain.

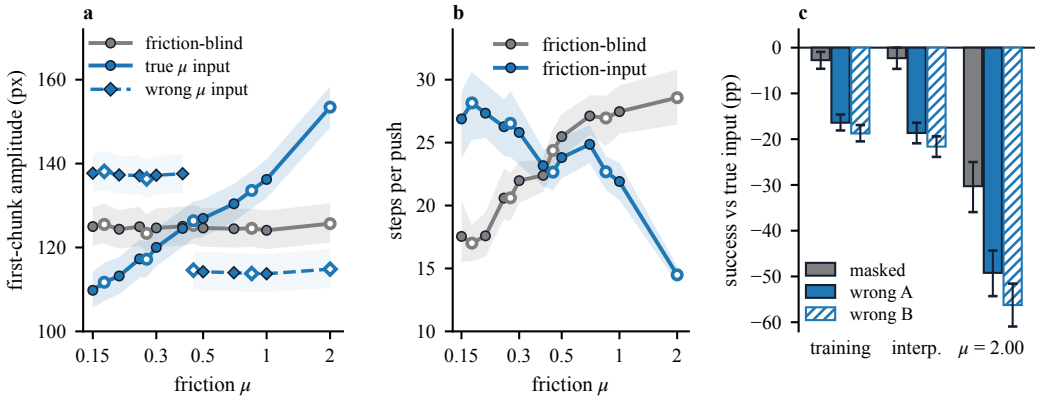

Figure 4: **The input sets the scale of the first push and the timing of contact, and a wrong value costs more than none.** (a) Commanded amplitude of the first action chunk, issued from the same observation at every tier, for the friction-blind and friction-input policies and for the masking-trained checkpoint given a wrong value under rule A (told 1.00 at $\mu \leq 0.40$, 0.20 above; dashed). (b) Mean push length per tier on the test set, successful episodes. (c) Cost of the masked input and of the two wrong inputs against the true input on the same checkpoint, in success points, per tier group. Open markers are held-out tiers; bands and error bars are 95% intervals.

The one-push delivery task gives the opposite result. The true-μ controller exceeds the prior controller on 8 to 11 of 11 tiers at every narrow band. Knowing the parameter before the single push therefore improves success. A light push along the direction of travel, followed by an analytic estimate from the block's coasting motion, supplies the parameter before the delivery: success rises over the prior by 54 to 82 percentage points at $w \leq 20$ px and by 27 to 52 at $w = 40$ px, averaged over the eleven tiers (Figure 3d). The probe's estimate is biased low by 0.4 to 12% depending on the push length and the tier, and the tolerance relation $|\hat{\mu}/\mu - 1| \leq w/D$ predicts which push lengths succeed at each band (Appendix D). The two tasks therefore differ in when the parameter has to be available. In Push-T, where later actions can correct earlier ones, ordinary execution supplies the parameter in time. In the delivery task, where one action decides the outcome, the parameter has to be acquired before that action.

9 Discussion and Conclusion

Criterion and budget. Of the 249 tier-level comparisons in this paper, 60 change their verdict between the success rule, the score and the 300- and 400-step budgets. The group-level comparisons on the training and held-out tiers keep their verdict under all four settings, and every comparison is reported under both criteria.

Scope. The study covers imitation-trained diffusion policies on one planar task with one hidden scalar parameter, one noise sample per state, one to three training seeds per policy and intervals over initial states; every policy descends from the human demonstrations through the released checkpoint and the collection. The room for a probe is measured for one controller design and one estimator; the delivery task uses an analytic estimator and an uncharged probe.

Conclusion. A pushing policy that faces an unobserved friction can obtain the parameter from the interaction the task already requires or from actions spent on finding it out. In Push-T, access to the true parameter improves success by +5.5 points on the training tiers and by +35.0 points above them, and a controller that estimates the parameter from ordinary pushing comes within 0.1 success points of one that knows it from the first step, so the task leaves no room for a probe. In a delivery task that allows no correction, a light push before the delivery improves success by 54 to 82 points. The value of a probe therefore depends on two measurable quantities: how much of the parameter ordinary execution already supplies, and whether the task allows a later action to correct the first. These results establish ordinary task interaction as a strong adaptation baseline and the reference for evaluating dedicated probing.

REPRODUCIBILITY STATEMENT

We commit to releasing the code and data upon acceptance: the environment, the initial-state sets, the collection manifests, training configurations, checkpoints, estimators and analysis scripts, together with the collection of 1,989,753 episodes. Every number in the paper is produced by a script that reads the stored per-episode records; The appendix, provided as supplementary material, gives the protocol, the simulator checks and the provenance of the reported values (Appendix A).

AI USE STATEMENT

Generative AI tools were used for drafting and revising the manuscript, for literature search and citation checking, for feedback on the experimental design, and for writing and orchestrating analysis and evaluation code. Every citation was checked against its publisher or arXiv record, and each attributed claim against the paper's abstract or text; analysis scripts were checked by recomputing reference values from the stored per-episode records; and the authors reviewed all AI-assisted text and code. The authors take responsibility for the final content of this work.

REFERENCES

Marina Y. Aoyama, João Moura, Juan Del Aguila Ferrandis, and Sethu Vijayakumar. Poke and strike: Learning task-informed exploration policies. In *Proceedings of the 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pp. 3008–3034, 2025.

Yaakov Bar-Shalom and Edison Tse. Dual effect, certainty equivalence, and separation in stochastic control. *IEEE Transactions on Automatic Control*, 19(5):494–500, 1974. doi: 10.1109/TAC.1974.1100635.

Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017. doi: 10.1109/TRO.2017.2721939.

Neil R. Bramley, Tobias Gerstenberg, Joshua B. Tenenbaum, and Todd M. Gureckis. Intuitive experimentation in the physical world. *Cognitive Psychology*, 105:9–38, 2018. doi: 10.1016/j.cogpsych.2018.05.001.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *International Journal of Robotics Research*, 44(10–11):1684–1704, 2025. doi: 10.1177/02783649241273668.

Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Anirvan Dutta, Etienne Burdet, and Mohsen Kaboli. Push to know! – visuo-tactile based active object parameter inference with dual differentiable filtering. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3137–3144, 2023. doi: 10.1109/IROS55552.2023.10341832.

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

Pete Florence, Corey Lynch, Andy Zeng, Oscar A. Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 158–168. PMLR, 2022. URL https://proceedings.mlr.press/v164/florence22a.html.

Michel Gevers. Identification for control: From the early achievements to the revival of experiment design. *European Journal of Control*, 11(4–5):335–352, 2005. doi: 10.3166/ejc.11.335-352.

Eleanor J. Gibson. Exploratory behavior in the development of perceiving, acting, and the acquiring of knowledge. *Annual Review of Psychology*, 39:1–41, 1988. doi: 10.1146/annurev.ps.39.020188.000245.

Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966. doi: 10.1109/TSSC.1966.300074.

Marco Iannotta, Yuxuan Yang, Johannes A. Stork, Erik Schaffernicht, and Todor Stoyanov. Can context bridge the reality gap? sim-to-real transfer of context-aware policies. *Robotics and Autonomous Systems*, 205:105594, 2026. doi: 10.1016/j.robot.2026.105594.

Shervin Javdani, Yuxin Chen, Amin Karbasi, Andreas Krause, J. Andrew Bagnell, and Siddhartha Srinivasa. Near optimal bayesian active learning for decision making. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 33 of *PMLR*, 2014.

Harun Karimpur, Christian Wolf, and Katja Fiehler. The (un)ideal physicist: How humans rely on object interaction for friction estimates. *Psychological Science*, 35(2):191–201, 2024. doi: 10.1177/09567976231221789.

Arjun Krishna, Edward Hu, and Dinesh Jayaraman. The value of sensory information to a robot. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=ikr5XomWHS`.

Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems (RSS)*, 2021. doi: 10.15607/RSS.2021.XVII.011.

Susan J. Lederman and Roberta L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19(3):342–368, 1987. doi: 10.1016/0010-0285(87)90008-9.

Jacky Liang, Saumya Saxena, and Oliver Kroemer. Learning active task-oriented exploration policies for bridging the sim-to-real gap. In *Robotics: Science and Systems (RSS)*, 2020. doi: 10.15607/RSS.2020.XVI.085.

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

Pedro A. Ortega, Markus Kunesch, Grégoire Delétang, Tim Genewein, Jordi Grau-Moya, Joel Veness, Jonas Buchli, Jonas Degrave, Bilal Piot, Julien Perolat, Tom Everitt, Corentin Tallec, Emilio Parisotto, Tom Erez, Yutian Chen, Scott Reed, Marcus Hutter, Nando de Freitas, and Shane Legg. Shaking the foundations: Delusions in sequence models for interaction and control. *arXiv preprint arXiv:2110.10819*, 2021.

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. doi: 10.1109/ICRA.2018.8460528.

Aaron Walsman, Muru Zhang, Sanjiban Choudhury, Dieter Fox, and Ali Farhadi. Impossibly good experts and how to follow them. In *International Conference on Learning Representations (ICLR)*, 2023.

Maggie Wang, Stephen Tian, Aiden Swann, Ola Shorinwa, Jiajun Wu, and Mac Schwager. Phys2real: Fusing vlm priors with interactive online adaptation for uncertainty-aware sim-to-real manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Wenhao Yu, Jie Tan, C. Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *Robotics: Science and Systems (RSS)*, 2017. doi: 10.15607/RSS.2017.XIII.048.

Wenxuan Zhou, Lerrel Pinto, and Abhinav Gupta. Environment probing interaction policies. In *International Conference on Learning Representations (ICLR)*, 2019.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. In *International Conference on Learning Representations (ICLR)*, 2020.