LOOPADAPT: PERSISTENT POLICY ADAPTATION VIA LOOP-TO-SCOPE CERTIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Policies adapted over long deployment horizons face a challenge beyond recovering from the current failure: every update also changes the adaptation options available in the future. We formalize this setting as persistent policy adaptation and introduce LOOPADAPT, a framework that links recurrent failure modes to the parameter scopes in which they can be jointly corrected. LOOPADAPT models low-progress behavior as recurrent loops, assigns each retained loop a differentiable objective, and tests whether their gradients admit a common descent direction within a prescribed update scope. This yields a loop-to-scope certificate that identifies first-order incompatibilities among recurrent failures and, when feasible, guarantees finite-step improvement while accounting for gradient estimation error, local curvature, and update radius. LOOPADAPT then ranks certified candidates by long-horizon value and validates them before commitment, explicitly accounting for how present updates reshape future checkpoints and adapter choices. Across 128 transient kernels, the certified direction improves every retained loop in all eight-dimensional settings, compared with 62.5% for mean descent and 96.875% for PCGrad. In a localized-shift PointMass ablation, incorporating the loop objective reduces mean loop potential by 41.6% while preserving 100% task success. Across ten 300-window streams, long-horizon and immediate update choices differ on 16.1% of windows, showing that locally attractive updates can alter future adaptation opportunities. These results show that persistent policy adaptation is fundamentally a joint problem of recurrent-failure geometry, scope-constrained update feasibility, and cross-window value.

1 INTRODUCTION

A policy updated once can be analyzed as a single input–output system. A policy updated repeatedly cannot: after each deployment window, the controller must decide whether to retain the current checkpoint, attach a local adapter, or modify a broader set of shared parameters. Each such decision affects not only current performance, but also the adaptation options available in future windows. We refer to this setting as persistent policy adaptation.

The central challenge is therefore not only how to adapt, but also where. A recurrent slip may be correctable with a lightweight output adapter; a shift in observations or dynamics may require a broader update; and benign novelty may warrant no intervention at all. Existing adaptation methods can optimize a prescribed parameter subset or select among predefined update candidates (Nagabandi et al., 2019; Xu et al., 2023; Dong et al., 2026; Bozkurt et al., 2026), but they generally do not determine whether the recurrent failures observed during deployment are jointly repairable within a given parameter scope. Recurrence provides the structure needed to make this determination. Revisited low-progress states induce behavioral loops, each of which can be treated as a distinct repair objective. This decomposition exposes conflicts that an aggregate loss may conceal: an update can reduce average failure burden while worsening a particular recurrent loop. The resulting requirement is to determine whether a declared parameter scope admits a direction that decreases every implicated loop and whether the achievable decrease is large enough to justify committing the update.

LOOPADAPT addresses this requirement through a loop-to-scope certificate. A transient dynamics model induces differentiable anchored-loop losses, whose gradients are projected into candidate local and shared parameter scopes under a common trust-region metric. For each scope, the distance from

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.

the origin to the convex hull of the projected loop gradients defines a common-descent margin. A sufficiently positive robust margin certifies the existence of a shared descent direction and yields a finite-step repair guarantee that accounts for estimation error, local curvature, and update radius; otherwise, the controller abstains or escalates to a broader scope. Classical MGDA solves the inner convex subproblem, while our contribution lies in constructing the behavior-to-scope test from recurrent deployment failures and integrating the resulting certificate into a persistent adaptation controller.

Figure 1 gives the full decision cycle. After certification, admissible candidates are ranked by task gain, burden reduction, retention, risk, cost, and future reuse. Independent evidence determines whether to commit, keep, or roll back. The resulting checkpoint and adapter memory become part of the next window's state.

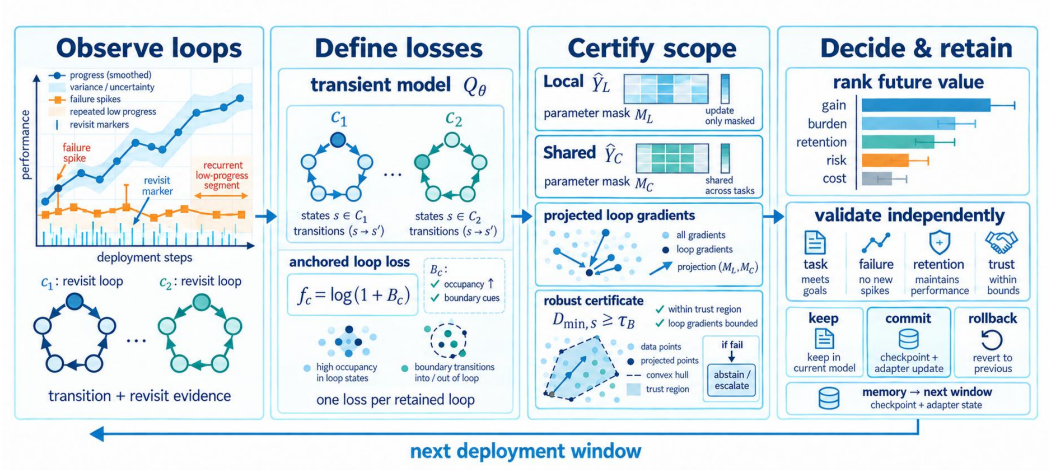


Figure 1: **Persistent policy adaptation with loop-to-scope certification.** A deployment window supplies transition, failure, and revisit evidence. A transient model turns recurrent low-progress behavior into one loss per retained loop. Scope-specific common-descent margins test whether local or shared parameters can jointly reduce those losses, and the least declared-cost scope meeting the robust target is proposed. A stateful controller ranks, validates, and retains candidates; committed parameters and adapter memory affect the next window.

Contributions. First, we formalize persistent adaptation as a sequence of coupled update decisions whose state includes the committed checkpoint and retained adapters. Second, we introduce a loop-to-scope certificate that maps recurrent behavior to trainable capacity, together with an obstruction result and a finite-step bound under gradient error and curvature. Third, we evaluate the mechanism in matched kernel comparisons, continuous control, multi-window routing, and repeated validation, keeping the role of each study explicit.

2 Related Work

Policy adaptation, recovery, and routing. Meta-RL and online policy adaptation use deployment experience to change behavior (Nagabandi et al., 2019; Xu et al., 2023; Dong et al., 2026). Residual control and low-rank adapters restrict the trainable subspace (Johannink et al., 2019; Hu et al., 2022), while budgeted selection and continual adapter routing decide which candidate to deploy (Bozkurt et al., 2026; Römer et al., 2026; Wang et al., 2025). Recent robot-learning systems emphasize retry and reset behavior, failure diagnosis, progress-aware control, and online world-model transfer (Zhao et al., 2026; Ke et al., 2026; Zeng et al., 2026; Liu et al., 2026; Jiang et al., 2026). Value-guided action selection and just-in-time intervention further show that task success alone can miss execution quality (Song et al., 2026; Mao et al., 2026). Diverse-reset training improves long-horizon dexterity (Yin et al., 2026). These methods adapt actions, skills, or supplied modules. LoopAdapt instead asks whether recurrent failures are jointly repairable within a proposed parameter scope, then carries the accepted update into a cross-window state.

Multiobjective descent and deployment safeguards. Common-descent geometry and minimum-norm convex-hull directions are classical (Fliege & Svaiter, 2000; Sener & Koltun, 2018); PCGrad modifies conflicting task gradients (Yu et al., 2020). These methods assume that the objective vector is already given. LOOPADAPT derives that vector from anchored behavioral loops, projects it into declared update scopes, and uses a robust finite-step bound to accept, reject, or escalate a scope. MGDA is therefore an inner solver rather than the proposed adaptation rule. Candidate validation remains separate from scope certification. When evidence is inspected repeatedly, time-uniform confidence sequences provide one principled monitoring tool under their stated sampling assumptions (Howard et al., 2021).

3 PROBLEM FORMULATION

At deployment window w, a policy π_θ can be kept unchanged, updated in a local scope, or updated in a larger shared scope. We first summarize recent low-progress behavior by a finite transient kernel Q_θ. The kernel is nonnegative and substochastic, with $\rho(Q_\theta) < 1$; missing probability mass denotes modeled exit, whether through useful progress or failure. For an initial distribution μ, let $N = (I - Q_\theta)^{-1}$ and $\eta^\mathsf{T} = \mu^\mathsf{T} N$. In the expressions below, Q_{ij} abbreviates $Q_\theta(i, j)$.

Let $c = (i_0, \dots, i_\ell = i_0)$ be a supported anchored loop. We define its expected traversal count $\mathcal{B}_c$, its stabilized loss f_c, and the scalar diagnostic Φ^{B} as

$$\begin{aligned} \mathcal{B}_c &= \eta_{i_0} \prod_{r=0}^{\ell-1} Q_{i_r i_{r+1}}, \\ f_c(\theta) &= \log(1 + \mathcal{B}_c), \\ \Phi^{\mathrm{B}} = \sum_{c \in \mathcal{C}} \nu_c f_c, &\qquad \nu_c \geq 0, \quad \sum_c \nu_c = 1. \end{aligned} \tag{1}$$

Here $\mathcal{B}_c$ counts completed visits to loop c before exit. The vector $(f_c)_{c \in \mathcal{C}}$ preserves which loops improve or regress; the aggregate Φ^{B} does not. Burden is also not a substitute for task success, because a rapid failure may reduce time spent in a loop. Every deployed candidate is therefore checked separately for task performance, failure risk, and retention.

For a supported loop with $\mathcal{B}_c > 0$ and a fixed abstraction, loop set, and initial distribution, the log-burden gradient is

$$\frac{\partial \log \mathcal{B}_c}{\partial \theta_k} = \frac{\mu^\mathsf{T} N (\partial Q / \partial \theta_k) N e_{i_0}}{\eta_{i_0}} + \sum_{r=0}^{\ell-1} \frac{\partial \log Q_{i_r i_{r+1}}}{\partial \theta_k}. \tag{2}$$

The first term measures how the update changes access to the loop anchor; the second measures how it changes transitions around the loop. Shared transitions couple these effects across loops, which is why scope adequacy must be tested jointly.

4 LOOPADAPT: FROM LOOPS TO PERSISTENT UPDATES

LOOPADAPT separates two decisions. The certificate asks which parameter scopes can support a joint repair. The persistent controller decides whether a certified candidate is worth committing after accounting for future windows. Figure 2 shows the computation and the state carried forward.

Let $M \succ 0$ define the common trust-region metric, $\|d\|_M^2 = d^\mathsf{T} M d$. Let $W_\mathcal{S}$ have orthonormal columns spanning $M^{1/2}\mathcal{S}$. A radius-r update in scope $\mathcal{S}$ is $d = -r M^{-1/2} W_\mathcal{S} v$ with $\|v\|_2 \leq 1$. For each loop, define the whitened gradient $h_c = M^{-1/2} \nabla f_c$ and the scope projection $a_c = W_\mathcal{S}^\mathsf{T} h_c$. All scope comparisons use the same loop set, loss scaling, metric, and radius.

4.1 LOOP-TO-SCOPE CERTIFICATE

The common-descent margin is

$$\gamma_\mathcal{S} = \max_{\|v\|_2 \leq 1} \min_{c \in \mathcal{C}} a_c^\mathsf{T} v = \min_{\alpha \in \Delta_{|\mathcal{C}|}} \left\| \sum_c \alpha_c a_c \right\|_2, \tag{3}$$

where Δ is the probability simplex. The right-hand side is the distance from the origin to the convex hull of the projected gradients. If its minimum-norm point is $z_\star \neq 0$, then $v_\star = z_\star/\|z_\star\|$ decreases every retained loop to first order. If the margin is zero, no such direction exists inside the scope. This is a first-order obstruction; it does not exclude nonlinear or multi-step repair.

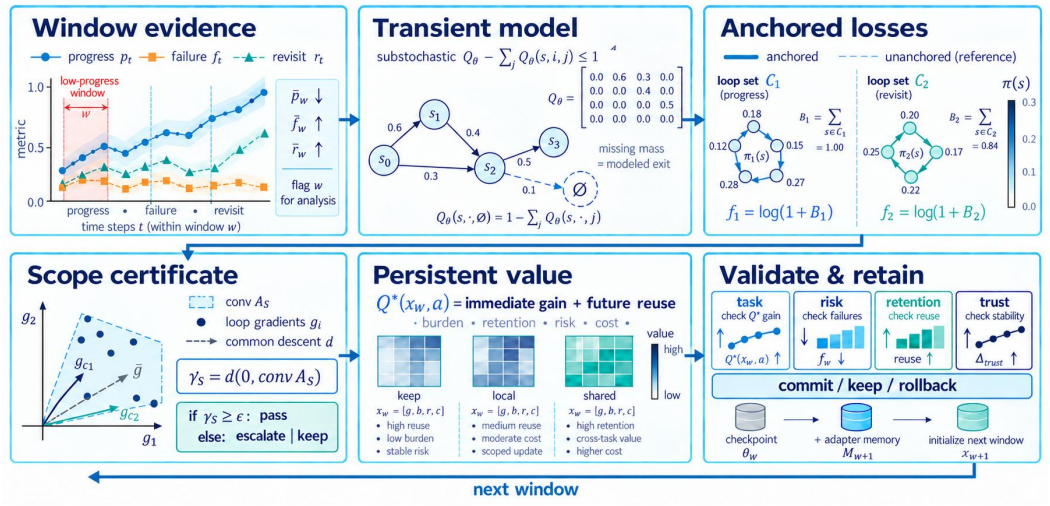


Figure 2: **Method overview.** Recent trajectories define a transient low-progress model and one loss per retained loop. Each scope receives projected loop gradients under the same metric, radius, and loop set. The certificate proposes the least declared-cost scope whose robust decrease bound meets the repair target. Keep, local, and shared candidates are then ranked by future value and independently checked before commitment. The committed checkpoint and adapter memory enter the next deployment state.

Proposition 1 (An aggregate gradient can hide a scope obstruction)**.** Let $M = I$ and $\mathcal{S} = \mathrm{span}(e_1)$. The gradient pairs

$$H_A = \{(1,1),(1,-1)\}, \qquad H_B = \{(3,1),(-1,-1)\} \tag{4}$$

have the same equally weighted mean $(1,0)$ and the same aggregate projected-energy fraction. Yet $\gamma_{\mathcal{S}}(H_A) = 1$ while $\gamma_{\mathcal{S}}(H_B) = 0$. Expanding H_B to the full plane gives $\gamma = 1/\sqrt{5} > 0$.

The example isolates why a scalar gradient or projected-energy score cannot certify joint repair. Figure 3(a) visualizes the conflict. With one nonzero loop, $\gamma_{\mathcal{S}}^2/\|h_1\|^2$ reduces to the usual scalar repairability fraction; the proposed criterion extends that diagnostic to several loops.

Theorem 2 (Conditional finite-step loop repair)**.** Suppose $\|\widehat{h}_c - h_c\|_2 \le \varepsilon$ holds simultaneously for all retained loops and every f_c is upper-smooth with constant L in the M-metric on the radius-$r_{\max}$ ball. Let $\widehat{v}_\star$ attain the estimated positive margin $\widehat{\gamma}_{\mathcal{S}}$. For $d = -rM^{-1/2}W_{\mathcal{S}}\widehat{v}_\star$ and $0 \le r \le r_{\max}$,

$$\min_c\{f_c(\theta) - f_c(\theta + d)\} \ge r(\widehat{\gamma}_{\mathcal{S}} - \varepsilon) - \tfrac{1}{2}Lr^2. \tag{5}$$

For $L > 0$, the largest nonnegative lower bound uses $r_\star = \min\{r_{\max}, (\widehat{\gamma}_{\mathcal{S}} - \varepsilon)_+/L\}$; when the robust margin is insufficient, the update is zero.

The guarantee requires simultaneous gradient-error control and a curvature bound over the trust region. It applies to the retained loops, not to cycles absent from the model. Appendix A gives the proofs, perturbation bounds, and numerical-solver conditions.

4.2 Cost-Ordered Scope Escalation

Let $\underline{D}_{\mathcal{S}}$ be the optimized lower bound in Equation (5), and let $C_{\mathcal{S}}$ be a declared construction cost. Given a positive repair target τ_B, LoopAdapt proposes

$$\mathcal{S}_\star \in \operatorname*{arg\,min}_{\mathcal{S}:\, \underline{D}_{\mathcal{S}} \ge \tau_B} C_{\mathcal{S}}. \tag{6}$$

If no scope passes, the controller keeps the checkpoint or gathers more evidence. For nested scopes with common $(M, \mathcal{C}, \varepsilon, L, r_{\max})$, the margin and its lower bound cannot decrease as the scope expands. The rule therefore returns the least declared-cost qualifying scope. It does not claim that a small scope must qualify, or that declared construction cost equals environment interaction. Because $\tau_B > 0$, an accepted scope has $\widehat{\gamma}_{\mathcal{S}} > \varepsilon$ and therefore $z_{\mathcal{S}} \neq 0$; the final normalization in Algorithm 1 is never evaluated for rejected scopes.

Algorithm 1 Cost-ordered loop-repair candidate proposal

Require: Checkpoint θ; loop set $\mathcal{C}$; metric $M \succ 0$; estimates $\{\widehat{h}_c\}$; scopes $\mathfrak{S}$ with bases $W_{\mathcal{S}}$ and costs $C_{\mathcal{S}}$; common ε, curvature constant $L > 0$, radius $r_{\max} > 0$, and target $\tau_B > 0$.
Ensure: Candidate $(\theta^+, \mathcal{S}_\star)$ or (θ, keep).

$\mathfrak{F} \leftarrow \varnothing$
for each $\mathcal{S} \in \mathfrak{S}$ **do**
  $\widehat{a}_c \leftarrow W_{\mathcal{S}}^{\mathsf{T}} \widehat{h}_c$ for all $c \in \mathcal{C}$
  $z_{\mathcal{S}} \leftarrow \arg\min_{z \in \mathrm{conv}\{\widehat{a}_c\}} \|z\|_2$; $\widehat{\gamma}_{\mathcal{S}} \leftarrow \|z_{\mathcal{S}}\|_2$
  $r_{\mathcal{S}} \leftarrow \min\{r_{\max}, (\widehat{\gamma}_{\mathcal{S}} - \varepsilon)_+ / L\}$
  $\underline{D}_{\mathcal{S}} \leftarrow r_{\mathcal{S}}(\widehat{\gamma}_{\mathcal{S}} - \varepsilon) - \frac{1}{2} L r_{\mathcal{S}}^2$
  if $\underline{D}_{\mathcal{S}} \geq \tau_B$ **then**
    $\mathfrak{F} \leftarrow \mathfrak{F} \cup \{\mathcal{S}\}$
  end if
end for
if $\mathfrak{F} = \varnothing$ **then**
  return (θ, keep)
end if
$\mathcal{S}_\star \leftarrow \arg\min_{\mathcal{S} \in \mathfrak{F}} C_{\mathcal{S}}$ (ties use a declared order)
$\theta^+ \leftarrow \theta - r_{\mathcal{S}_\star} M^{-1/2} W_{\mathcal{S}_\star} z_{\mathcal{S}_\star} / \|z_{\mathcal{S}_\star}\|_2$
return $(\theta^+, \mathcal{S}_\star)$

4.3 Persistent Ranking and Deployment Validation

Certification answers whether a candidate can reduce the retained loop losses. It does not decide whether that candidate should be deployed. For candidate a at window w, let $\Delta J_w^{\mathrm{task}}$ and ΔB_w denote improvements, while $\Delta L_w^{\mathrm{retain}}$ and $\Delta R_w^{\mathrm{fail}}$ denote increases in retention loss and failure risk. We use

$$\mathcal{H}_w(a) = \Delta J_w^{\mathrm{task}}(a) + \lambda_B \Delta B_w(a) - \lambda_R \Delta L_w^{\mathrm{retain}}(a) - \lambda_F \Delta R_w^{\mathrm{fail}}(a) - \lambda_C C_w(a), \tag{7}$$

$$\mathcal{Q}^\star(x, a) = \sup_{\varrho} \mathbb{E}_\varrho \left[\sum_{j \geq 0} \gamma_R^j \mathcal{H}_{w+j}(a_{w+j}) \mid x_w = x, a_w = a \right]. \tag{8}$$

Here x_w includes the current checkpoint and retained adapters, and ϱ ranges over later admissible routing policies. Bounded rewards and $0 \leq \gamma_R < 1$ make the value finite. Evidence masks inadmissible candidates and independently checks the selected update; keep is always available. The synthetic study evaluates multi-window value learning, whereas planar contact uses repeated fixed-condition scoring. Appendix C gives both implementations.

5 Experiments

The evaluation isolates four mechanisms: scope-conflict detection, recurrent-burden reduction after task success saturates, long-horizon action selection, and finite-data validation. We use a separate replication unit for each mechanism and report the unit with every aggregate. Table 1 maps each claim to its protocol and primary result.

5.1 Experimental Design

We evaluate scope geometry on 128 exact transient kernels with analytic derivatives. PointMass uses five policy seeds and separates localized, diffuse, and benign shifts. Planar contact evaluates local and

global output adapters on four held-out shifts for each of five policies. The routing generator trains ten routers and evaluates 300 held-out windows per router; null streams measure repeated-inspection errors. The studies share the same decision interface but are component evaluations rather than one physical deployment. Full protocols and per-seed results appear in Appendices E.1 and F.3. Table 1 reports the independent unit and primary signal for each claim.

Table 1: Evidence map. Contact rows list fixed local, fixed global, family rule, and learned scorer in that order; each row gives its independent unit and supported claim.

Study	Units	Result and supported claim
Kernel	128	8-D all-improve: 100/96.9/62.5% (cert./PCGrad/mean); scope conflicts.
Burden	96	ΔB: 6.57/7.68/-0.07% (local/global/unaware); recurrence cost.
PointMass	5	Φ^{B}: 0.0936$\rightarrow$0.0547; success 100%; beyond success.
Routing	10×300	regret 0.0014 vs. 0.0436/0.0875; delayed value.
Contact success	5	Shift/Nom.: 99.99/100; 100/100; 99.84/99.86; 99.84/99.86%; contact success.
Contact cost	5	sizes 33/12; Ψ: 2.437/1.975/2.296/2.107; U: 1.5470/1.5674/1.5542/1.5590; learned $\Delta\Psi = -0.189$.
Null checks	4,000	zero false accepts; repeated-check control.

Baselines and endpoints. Kernel baselines are mean descent, the classical multiobjective steepest-descent direction (Fliege & Svaiter, 2000; Sener & Koltun, 2018), and PCGrad (Yu et al., 2020); the certificate uses the exact MGDA solution only for its inner convex problem. The control studies compare task-only updates, fixed local and global scopes, a family-label rule, and a learned scorer. We report the metric that matches each claim: per-loop regression and all-loop improvement for kernels, task success and recurrent burden for PointMass, and success, proxy burden, retention, and utility for contact. Recent batch-online and adapter-routing methods provide context (Dong et al., 2026; Wang et al., 2025; Römer et al., 2026); their interfaces are different and they are not treated as reproduced baselines.

5.2 Scope-Conflict Diagnostics

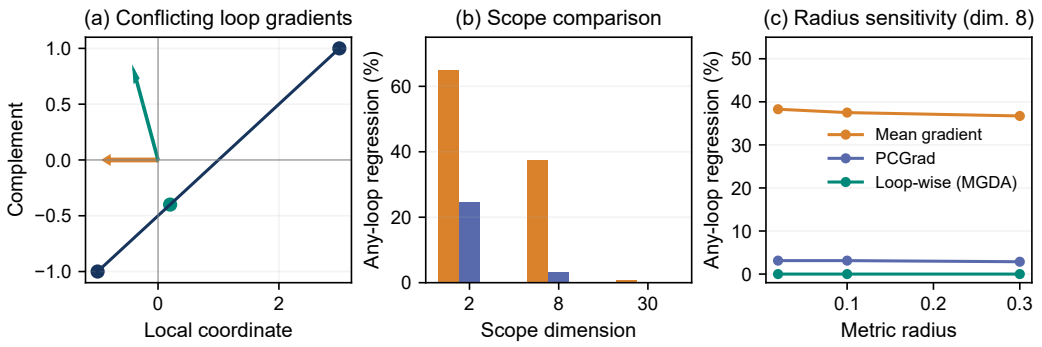


Figure 3: **Matched-scope diagnostic.** (a) The gradients from Proposition 1. (b) Any-loop regression across 128 kernels at metric radius 0.10. (c) Radius sensitivity in eight-dimensional scopes. The panels link incompatible projected directions to regression frequency and radius robustness. PCGrad averages all six task orders within each kernel; the proposed diagnostic uses MGDA as its inner solver. Zero regression includes abstentions, which Table 2 reports separately. Lines in (c) guide the eye.

Panels (a)–(c) link projected conflict to nonlinear regression and radius robustness. We sample 128 five-state kernels with three anchored loops and scopes 2, 8, and 30 under shared losses, analytic gradients, metric, and radius. PCGrad averages six orders; the certificate abstains at zero robust margin. Table 2 gives outcome frequencies, and the Evidence Map gives eight-dimensional gains.

Table 2: Loop-level outcomes after the nonlinear radius-0.10 update. Entries are count (percentage); K denotes kernels and O order–kernel evaluations. Mean/certificate rows use 128 K; PCGrad uses six orders per kernel (768 O). The kernel is the independent unit.

Scope	Direction	Denominator	Any regression ↓	All improve ↑	Abstain
2	Mean gradient	128 K	83 (64.8)	45 (35.2)	0 (0.0)
2	PCGrad	768 O	189 (24.6)	579 (75.4)	0 (0.0)
2	Loop certificate	128 K	**0 (0.0)**	98 (76.6)	30 (23.4)
8	Mean gradient	128 K	48 (37.5)	80 (62.5)	0 (0.0)
8	PCGrad	768 O	24 (3.1)	744 (96.9)	0 (0.0)
8	Loop certificate	128 K	**0 (0.0)**	**128 (100.0)**	0 (0.0)
30	Mean gradient	128 K	1 (0.8)	127 (99.2)	0 (0.0)
30	PCGrad	768 O	**0 (0.0)**	**768 (100.0)**	0 (0.0)
30	Loop certificate	128 K	**0 (0.0)**	**128 (100.0)**	0 (0.0)

At radius 0.10 in the eight-dimensional scope, mean descent regresses on 48/128 kernels and PCGrad on 24/768 order–kernel evaluations, whereas the certificate has zero regressions and improves every loop in all 128 kernels. Its largest worst-loop gain confirms that it protects the weakest loop rather than only the average. In two dimensions it abstains on 30/128 kernels instead of forcing an incompatible update. Under the predeclared 80%-of-full-margin target, every selector chooses dimension 30, so this result supports rejection of under-capacity scopes but does not establish cost savings; the fixed-radius study also does not calibrate the theorem's error and curvature constants.

5.3 Persistence and Adaptation Decisions

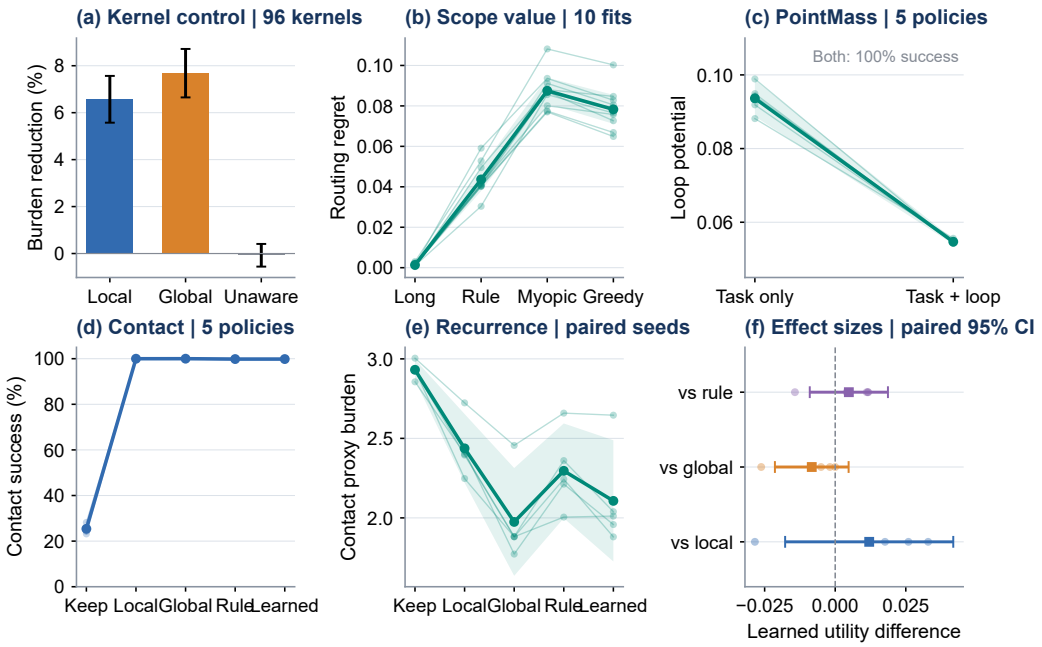


Figure 4: **Quantitative evidence.** (a) Burden reduction with approximate normal 95% intervals. (b–e) Paired method comparisons: thin lines join matched seeds, thick lines show means, and bands show pointwise t intervals; success intervals are clipped to $[0, 100]$. The connecting lines compare methods and are not training trajectories. (f) Learned-contact utility minus each baseline, with paired differences and t intervals. Contact panels share five policies and four shifts.

Burden-aware updates. Figure 4(a) shows that, on 96 kernels, local and global burden descent reduce scalar burden by $6.57 \pm 1.00\%$ and $7.68 \pm 1.03\%$, respectively; a matched-norm burden-unaware

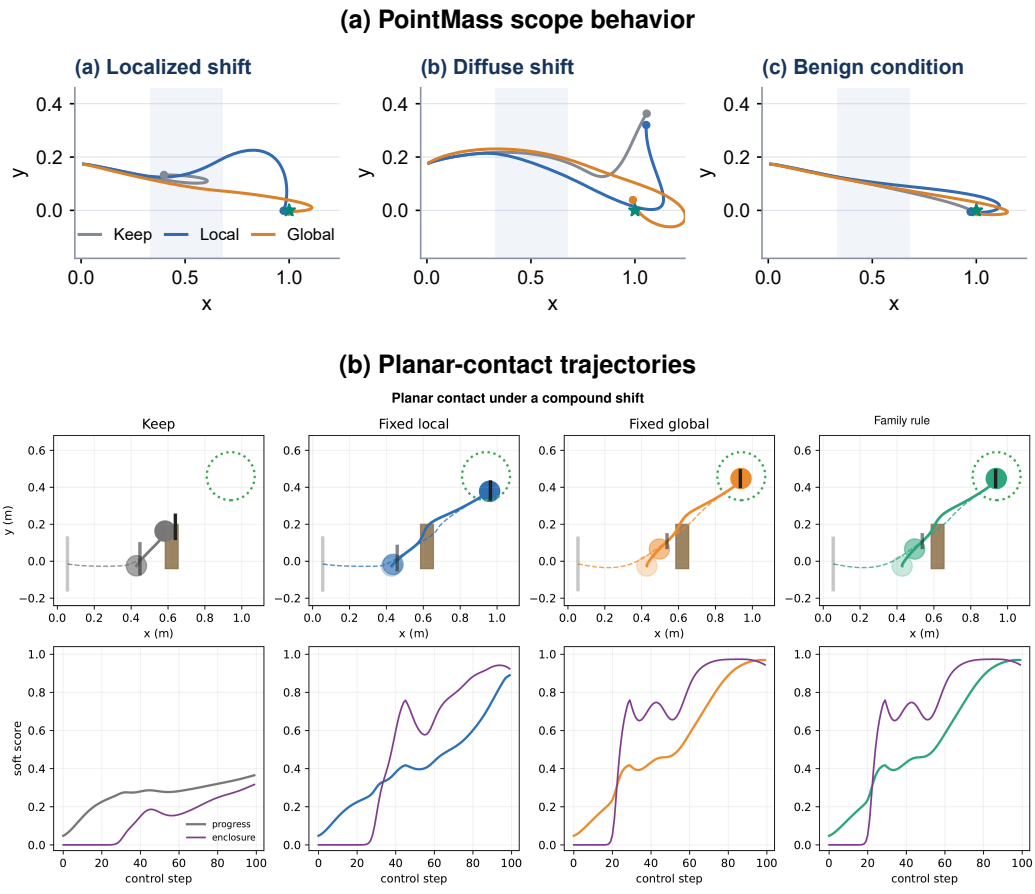
direction changes it by $-0.07 \pm 0.48\%$. In the localized-shift PointMass ablation shown in panel (c), task-only and task-plus-burden updates both reach 100% success over five policy seeds. The recurrent objective still lowers mean potential from 0.0936 to 0.0547, a 41.6% reduction, with paired difference 0.0389 and bootstrap 95% interval $[0.0353, 0.0422]$. Thus terminal success alone does not distinguish the two controllers.

Cross-window value. Panel (b) reports the scope-value comparison. Across ten independently trained routers and 300 held-out windows per router, immediate and long-horizon choices differ on $16.1 \pm 1.3\%$ of windows. The disagreement is concentrated in localized and recurring shifts, where adapter reuse and retention debt affect later value. Mean long-horizon regret is 0.0014 ± 0.0005, compared with 0.0436 ± 0.0050 for the hand rule and 0.0875 ± 0.0056 for the separately trained myopic scorer. Since gain and risk features are supplied by the generator, this result isolates cross-window value estimation rather than end-to-end failure diagnosis.

Planar contact. Panels (d)–(f) summarize the planar-contact study. Learned scoring lowers the mean proxy by 0.189 relative to the family rule at the same observed success rate, while the paired utility differences remain small. The paired t interval is $[-0.027, 0.405]$, the sign-randomization value is $p = 0.125$, and the percentile-bootstrap interval excludes zero (Appendix F.3). We therefore treat the comparison as descriptive: the learned scorer improves the recorded proxy, while fixed global remains best in both proxy and utility. Both branches freeze the backbone, so this study evaluates output-adapter selection rather than the full shared-parameter update shown in Figure 2.

5.4 Rollout Analysis

Figure 5 groups the three rollout views so that their roles can be compared directly.



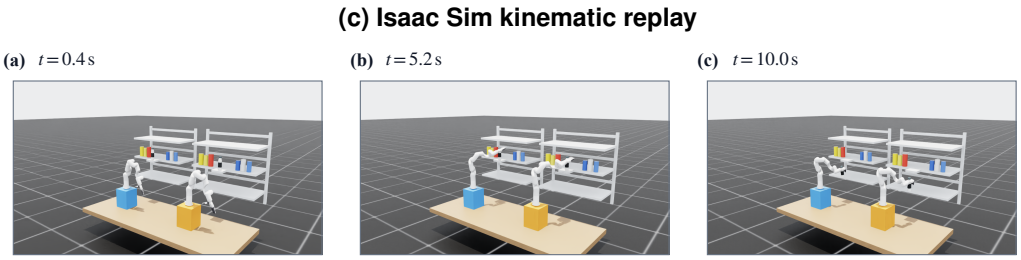


Figure 5: **Rollout analysis.** Panels (a)–(c) provide complementary views of deployment behavior. Panel (a) shows recorded PointMass trajectories for localized, diffuse, and benign shifts (seed 71); panel (b) shows compound-shift planar-contact rollouts with progress and enclosure traces (seed 99173); and panel (c) shows the Isaac Sim kinematic replay. Dots mark endpoints, stars mark goals, and the Family rule is the fixed heuristic.

Panel (a) shows that several updates can reach the goal while differing in recurrent motion and scope, which is why terminal success is paired with the loop metrics above. In panel (b), the unadapted controller stalls near the obstacle, whereas the adapted outputs progress toward the goal; the lower traces localize slow or enclosed motion. Panel (c) preserves the three-dimensional Isaac Sim context and is qualitative rather than an additional quantitative comparison. Appendix E.3 records the replay frames, trace indices, and procedure.

Across these views, endpoint success, recurrent burden, and geometric behavior provide complementary evidence. The rollout panels therefore explain why the aggregate metrics are reported together: a method can preserve the endpoint while changing the motion needed to reach it, and a successful contact trace can still carry different progress or enclosure costs.

6 Conclusion

LoopAdapt treats repeated policy adaptation as a stateful decision problem. Its loop-to-scope certificate links observed recurrence to parameter capacity, and its controller accounts for the future value of committed updates and retained adapters. Matched kernel comparisons show that the certificate prevents loop regressions missed by aggregate directions; control studies show that recurrent burden distinguishes policies after success saturates; and synthetic streams show that delayed consequences alter adaptation choices. The evidence supports the proposed mechanisms individually and provides a concrete basis for a future integrated physical deployment.

AI Use Statement

Generative AI tools were used solely for language polishing. All scientific content, claims, and interpretations remain the responsibility of the authors.

Reproducibility Statement

The release contains figure sources, per-seed observations, analysis scripts, executable checks, proofs, and study protocols. Appendices A and E.1 identify the assumptions and independent units used by each result.

REFERENCES

Jianibieke Adalibieke, Qianwei Han, Xueyi Liu, Yuzhe Qin, and Li Yi. AdaDexTrack: Dynamic modulation for adaptive and generalizable dexterous manipulation tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 28021–28031, 2026.

Alper Kamil Bozkurt, Xiaoan Xu, Shangtong Zhang, Miroslav Pajic, and Yuichi Motai. Adaptive policy selection and fine-tuning under interaction budgets for offline-to-online reinforcement learning. In Proceedings of The 8th Annual Learning for Dynamics and Control Conference, volume 331 of Proceedings of Machine Learning Research, pp. 1804–1817. PMLR, 2026.

Hal Caswell. Matrix Population Models: Construction, Analysis, and Interpretation. Sinauer Associates, second edition, 2001.

Perry Dong, Suvir Mirchandani, Dorsa Sadigh, and Chelsea Finn. What matters for batch online reinforcement learning in robotics? In International Conference on Learning Representations, 2026.

Senyu Fei, Siyin Wang, Li Ji, Ao Li, Shiduo Zhang, Liming Liu, Jinlong Hou, Jingjing Gong, Xianzhong Zhao, and Xipeng Qiu. SRPO: Self-referential policy optimization for vision-language-action models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6718–6728, 2026.

Jörg Fliege and Benar Fux Svaiter. Steepest descent methods for multicriteria optimization. Mathematical Methods of Operations Research, 51(3):479–494, 2000.

Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform, nonparametric, nonasymptotic confidence sequences. The Annals of Statistics, 49(2):1055–1080, 2021.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In International Conference on Learning Representations, 2022.

Guangfeng Jiang, Yueru Luo, Jun Liu, Yi Huang, Yiyao Zhu, Zhan Qu, Dave Zhenyu Chen, Bingbing Liu, and Xu Yan. WPT: World-to-policy transfer via online world model distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 17842–17852, 2026.

Tobias Johannink, Shikhar Bahl, Ashvin Nair, Jianlan Luo, Avinash Kumar, Matthias Loskyll, Juan Aparicio Ojea, Eugen Solowjow, and Sergey Levine. Residual reinforcement learning for robot control. In 2019 International Conference on Robotics and Automation (ICRA), pp. 6023–6029, 2019.

Sophia L. Kalpazidou. Cycle Representations of Markov Processes. Springer, second edition, 2006.

Zundong Ke, Junlin Chen, Jiayi Zhu, Kuanhao Xia, Boyi Zhao, and Jiayuan Gu. REACH: Explicit recovery behavior for diffusion policies. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 38498–38508, 2026.

Stephen J. Kirkland, Michael Neumann, N. Ormes, and Jianhong Xu. On the elasticity of the Perron root of a nonnegative matrix. SIAM Journal on Matrix Analysis and Applications, 24(2):454–464, 2002.

Yuanzhe Liu, Jingyuan Zhu, Yuchen Mo, Gen Li, Xu Cao, Jin Jin, Yifan Shen, Zhengyuan Li, Tianjiao Yu, Wenzhen Yuan, Fangqiang Ding, and Ismini Lourentzou. PALM: Progress-aware policy learning via affordance reasoning for long-horizon robotic manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 28096–28110, 2026.

Yanbo Mao, Jianlong Fu, Ruoxuan Zhang, Hongxia Xie, and Meibao Yao. Beyond success: Refining elegant robot manipulation from mixed-quality data via just-in-time intervention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13508–13518, 2026.

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, 2019.

Soomin Park, Eunseong Lee, Kwang Bin Lee, and Sung-Hee Lee. MaskAdapt: Learning flexible motion adaptation via mask-invariant prior for physics-based characters. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2285–2294, 2026.

Ralf Römer, Yi Zhang, Yuming Li, and Angela P. Schoellig. CLARE: Continual learning for vision-language-action models via autonomous adapter routing and expansion. IEEE Robotics and Automation Letters, 11(7): 8331–8338, 2026.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In Advances in Neural Information Processing Systems, volume 31, 2018.

Eugene Seneta. Non-negative Matrices and Markov Chains. Springer, second edition, 2006.

Abhishek Sharma, Leo Benac, Sonali Parbhoo, and Finale Doshi-Velez. Decision-point guided safe policy improvement. In Proceedings of the 28th International Conference on Artificial Intelligence and Statistics, volume 258 of Proceedings of Machine Learning Research, pp. 2935–2943, 2025.

Haoming Song, Delin Qu, Yuanqi Yao, Qizhi Chen, Jiarui Li, Qi Lv, Yiwen Tang, Li Kang, Heng Zhou, Xianqiang Gao, Yuhang Tang, Xiaofan Li, Modi Shi, Guanghui Ren, Maoqing Yao, Bin Zhao, Dong Wang, and Xuelong Li. FM-Steer: Enhance generalist policies with value-guided cascaded denoising. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13407–13418, 2026.

Huajie Tan, Sixiang Chen, Yijie Xu, Zixiao Wang, Cheng Chi, Yuheng Ji, Yaoxu Lyu, Zhongxia Zhao, Xiansheng Chen, Peterson Co, Shaoxuan Xie, Guocai Yao, Pengwei Wang, Zhongyuan Wang, and Shanghang Zhang. General process reward modeling for robotic reinforcement learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22412–22422, 2026.

Huiyi Wang, Haodong Lu, Lina Yao, and Dong Gong. Self-expansion of pre-trained models with mixture of adapters for continual learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10087–10098, 2025.

Mengdi Xu, Yuchen Lu, Yikang Shen, Shun Zhang, Ding Zhao, and Chuang Gan. Hyper-Decision Transformer for efficient online policy adaptation. In International Conference on Learning Representations, 2023.

Patrick Yin, Tyler Westenbroek, Zhengyu Zhang, Ignacio Dagnino, Eeshani Shilamkar, Numfor Mbiziwo-Tiapo, Simran Bagaria, Xinlei Liu, Galen Mullins, Andrey Kolobov, and Abhishek Gupta. Emergent dexterity via diverse resets and large-scale reinforcement learning. In International Conference on Learning Representations, 2026.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. In Advances in Neural Information Processing Systems, volume 33, pp. 5824–5836, 2020.

Xianchao Zeng, Xinyu Zhou, Youcheng Li, Jiayou Shi, Tianle Li, Liangming Chen, Lei Ren, and Yong-Lu Li. Diagnose, correct, and learn from manipulation failures via visual symbols. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 42386–42395, 2026.

Ganlong Zhao, Zijia Tang, Xingping Chen, Zhanghui Kuang, Ye Tian, and Guanbin Li. FLARE: A failure-aware framework for autonomous correction and recovery in visual-language robotic manipulation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22391–22401, 2026.

A Theoretical Analysis

A.1 Common-descent identity and obstruction

Fix a nonempty supported loop set, a policy checkpoint, and a positive definite metric. Write $a_c = W_S^\mathsf{T} M^{-1/2} \nabla f_c$ and let $K = \mathrm{conv}\{a_c\}$. Since the unit ball and simplex are compact convex sets and $\sum_c \alpha_c a_c^\mathsf{T} v$ is bilinear, minimax interchange gives

$$\max_{\|v\|\le 1} \min_{\alpha\in\Delta} \sum_c \alpha_c a_c^\mathsf{T} v = \min_{\alpha\in\Delta} \max_{\|v\|\le 1} \left(\sum_c \alpha_c a_c\right)^\mathsf{T} v = \min_{z\in K} \|z\|.$$

For the projection $z_\star$ of zero onto K, the variational inequality gives $a_c^\mathsf{T} z_\star \ge \|z_\star\|^2$ for every c. If $z_\star \neq 0$, normalizing it attains simultaneous decrease at least $\|z_\star\|$, and minimax equality makes this optimal. If zero belongs to K, some convex combination of the a_c is zero; hence no vector can have strictly positive inner product with all a_c. This is a first-order statement about the frozen losses only. These arguments are standard common-descent geometry (Fliege & Svaiter, 2000; Sener & Koltun, 2018).

A.2 Proof of Proposition 1

Both means equal $(1, 0)$, whose projection onto $\mathrm{span}(e_1)$ retains all its energy. Projecting H_A gives $\{1, 1\}$, with hull distance one. Projecting H_B gives $\{3, -1\}$, whose convex hull contains zero. In the full plane, the segment for H_B is $(-1, -1) + t(4, 2)$, $0 \le t \le 1$. Its closest point occurs at $t = 3/10$, giving $(1/5, -2/5)$ and norm $1/\sqrt{5}$. Linear losses realize these gradients exactly, so the example gives a local optimization counterexample. The transient-kernel experiment separately evaluates gradients of traversal losses.

A.3 Proof of Theorem 2

For $r = 0$, the claimed lower bound is zero. Consider $r > 0$. Metric upper-smoothness means $f_c(\theta + d) \le f_c(\theta) + \nabla f_c^\mathsf{T} d + (L/2)\|d\|_M^2$ throughout the declared ball. Orthogonality of W_S and Cauchy–Schwarz give

$$\nabla f_c^\mathsf{T}(-d)/r = h_c^\mathsf{T} W_S \widehat{v}_\star \ge \widehat{h}_c^\mathsf{T} W_S \widehat{v}_\star - \varepsilon \ge \widehat{\gamma}_S - \varepsilon.$$

Since $\|d\|_M = r$ for the nonzero normalized direction, substituting proves the bound. Maximizing the concave quadratic on $[0, r_{\max}]$ yields $r_\star$. If $L = 0$ and the robust margin is positive, use $r_{\max}$; otherwise retain zero. A direction with numerical primal margin $\min_c \widehat{a}_c^\mathsf{T} v$ can replace the optimum conservatively, while an unconverged dual objective alone cannot certify a lower bound.

A.4 Error stability, scope monotonicity, and conditional probability

If $\|\widehat{h}_c - h_c\| \le \varepsilon$ for all c, then every convex combination changes by at most ε after any orthogonal scope projection. Thus $|\widehat{\gamma}_S - \gamma_S| \le \varepsilon$. If $S_1 \subseteq S_2$, the feasible primal directions are nested, so $\gamma_{S_1} \le \gamma_{S_2}$. The optimized lower bound is monotone under common error, curvature, and radius assumptions. The selected scope is the least declared cost among those satisfying the sufficient bound; a rejected scope need not be truly incapable of repair.

If the gradient-error event holds jointly over all inspected loops/scopes with probability at least $1 - \delta$, and the smoothness bound is deterministic on the common ball, the selected update satisfies the bound on that same event, hence with probability at least $1 - \delta$. No additional conditioning on acceptance is implied. This does not create a valid error radius: adaptive loop discovery, checkpoint changes, estimated metrics, and black-box gradients require their own calibration. Smoothness cannot be established merely by observing a handful of successful finite steps. The current experiments do not estimate these quantities.

A.5 Loss scaling and feasibility

All compared scopes must use the same fixed losses, anchors, normalization, metric and trust radius. Multiplying one loop loss by a positive constant changes the common margin and therefore the

preference; there is no invariance to arbitrary loss scaling. A zero-gradient retained loop blocks strict common descent. Removing such a loop changes the protected set and must be a declared rule, not an outcome-dependent deletion. The method concerns retained loop losses; task, safety, retention, new-cycle, and coverage constraints remain separate. Costs in Equation (6) require measured or independently modeled units if an efficiency claim is made.

A.6 Proof of expected anchored loop burden

For each time $t < T$, condition on $X_t = i_0$. By the Markov property, the probability that the next ℓ low-progress states follow the anchored cycle is $p_c(Q)$. Therefore

$$\mathbb{E}[N_c] = \sum_{t \geq 0} \mathbb{P}(t < T, X_t = i_0)\, p_c(Q).$$

The sum is the expected number of visits to i_0 before exit, equal to $[\mu_w^{\mathsf{T}}(I - Q)^{-1}]_{i_0}$. Multiplying by $p_c(Q)$ gives Equation (15). Overlapping occurrences do not invalidate the identity because expectation is linear and each anchored start time is counted separately.

Anchor dependence and failure exits. Canonical anchoring specifies the counted event; it does not give rotation-invariant burden. For $Q_{01} = 0.8$, $Q_{10} = 0.5$, other entries zero, and $\mu = (1, 0)$, the two possible anchored traversal counts have expectations $2/3$ and $8/15$. Numerical state relabeling can therefore change a lexicographically chosen anchor. A semantic anchor should be fixed and audited; averaging over anchors defines a different objective. Similarly, a one-state loop with $Q_{00} = 0.9$ has burden 9, whereas immediate failure with $Q_{00} = 0$ has burden 0. The joint task/failure terms are essential, and neither loop weighting nor a success interpretation follows from the counting identity alone.

A.7 Gradient of expected loop burden

Let $N = (I - Q)^{-1}$ and $\eta^{\mathsf{T}} = \mu^{\mathsf{T}} N$. Matrix differentiation gives $dN = N(dQ)N$ and hence $d\eta_{i_0} = \mu^{\mathsf{T}} N(dQ) N e_{i_0}$. Dividing by η_{i_0} and adding the differential of $\log p_c(Q)$ yields Equation (17). The derivation treats μ and the macro-state definition as fixed during one candidate update; their outer-loop refresh is not differentiated.

A.8 Proof of perturbation stability

The resolvent identity gives

$$\widehat{N} - N = N \Delta \widehat{N}.$$

Because $\widehat{N} = (I - N\Delta)^{-1} N$ and $\|N\Delta\| < 1$, the Neumann-series bound yields

$$\|\widehat{N}\| \leq \frac{\|N\|}{1 - \|N\|\|\Delta\|},$$

and therefore Equation (27). For the occupancy term, apply the mean-value bound for $\log x$ between η_{i_0} and $\widehat{\eta}_{i_0}$. For each loop edge, $\widehat{Q}_{ij} \geq q_{\min}/2$, and the derivative of $\log x$ is at most $2/q_{\min}$ on this interval. Summing over ℓ edges gives Equation (28).

A.9 Proofs of repairability capacity and metric-scoped burden descent

Before proving the descent bound, we establish the capacity interpretation in Proposition 5. Let $U = U_{\mathcal{S}}$, $A = U^{\mathsf{T}} M U$, and $v = M^{-1} g$. The M-orthogonal projector onto $\mathcal{S}$ is $P_{\mathcal{S}}^M = U A^{-1} U^{\mathsf{T}} M$, so

$$\|P_{\mathcal{S}}^M v\|_M^2 = g^{\mathsf{T}} U A^{-1} U^{\mathsf{T}} g,$$
$$\|v\|_M^2 = g^{\mathsf{T}} M^{-1} g.$$

For $q(d) = g^{\mathsf{T}} d + \frac{1}{2} d^{\mathsf{T}} M d$, the unrestricted minimizer is $-M^{-1} g$ and the minimizer restricted to $\mathcal{S}$ is $-U A^{-1} U^{\mathsf{T}} g$. Their predicted decreases are respectively $\frac{1}{2} g^{\mathsf{T}} M^{-1} g$ and $\frac{1}{2} g^{\mathsf{T}} U A^{-1} U^{\mathsf{T}} g$. Taking the ratio proves Equation (19); M-orthogonal projection is nonexpansive, giving the interval $[0, 1]$.

The expression depends only on $\mathcal{S}$, not on the selected full-rank basis U. Finally, substituting the stabilized denominator from Equation (18) yields the stated shrinkage factor $E/(E+\epsilon)$.

Let $U = U_{\mathcal{S}}$, $M = M_w$, $g = g_w$, and $A = U^{\mathsf{T}}MU$. Because $M \succ 0$ and U has full column rank, $A \succ 0$. For $d = -\eta UA^{-1}U^{\mathsf{T}}g$,

$$g^{\mathsf{T}}d = -\eta\, g^{\mathsf{T}}UA^{-1}U^{\mathsf{T}}g$$

and

$$d^{\mathsf{T}}Md = \eta^2 g^{\mathsf{T}}UA^{-1}U^{\mathsf{T}}MUA^{-1}U^{\mathsf{T}}g = \eta^2 g^{\mathsf{T}}UA^{-1}U^{\mathsf{T}}g.$$

Substituting these identities into the metric-smoothness inequality in Equation (20) gives Equation (21). The decrease coefficient is positive for $0 < \eta < 2/L_{B,M}$. At an equal trust-region radius $\|d\|_M \le r$, Cauchy–Schwarz gives a maximum linear decrease $r\sqrt{g^{\mathsf{T}}UA^{-1}U^{\mathsf{T}}g}$ in the subspace and $r\sqrt{g^{\mathsf{T}}M^{-1}g}$ globally. Their ratio is the square root of the unstabilized capacity fraction. At $g = 0$ the ratio is undefined; the stabilized score is zero and no first-order descent is implied. The ratio in Equation (18) compares this restricted descent energy with the full-space energy $g^{\mathsf{T}}M^{-1}g$; the restricted quantity cannot exceed the full one because it is the optimum of the same quadratic model over a subset of directions.

A.10 Proof of shielded learned-routing bound

Let $\widehat{a} = \arg\max_{a\in\mathcal{A}^{\mathrm{cert}}} \widehat{\mathcal{Q}}_\psi(x,a)$ and let $a^{\mathrm{cert}} = \arg\max_{a\in\mathcal{A}^{\mathrm{cert}}} \mathcal{Q}^\star(x,a)$. Uniform approximation implies

$$\mathcal{Q}^\star(x,\widehat{a}) \ge \widehat{\mathcal{Q}}_\psi(x,\widehat{a}) - \epsilon_Q \ge \widehat{\mathcal{Q}}_\psi(x,a^{\mathrm{cert}}) - \epsilon_Q \ge \mathcal{Q}^\star(x,a^{\mathrm{cert}}) - 2\epsilon_Q,$$

which proves Equation (26). If the best safe action $a^\star$ belongs to $\mathcal{A}^{\mathrm{cert}}$, then a^{cert} is at least as valuable as $a^\star$ over the certified subset and the same inequality applies relative to $a^\star$.

A.11 Proof of repeatedly monitored accepted updates

By assumption, each confidence sequence is time-uniform under the filtration generated by the actual adaptive collection process. Allocate their error probabilities across all windows, candidates, constraints, and monitoring phases so the total is at most δ. A union bound then implies simultaneous coverage for all times with probability at least $1-\delta$. On this event, every acceptance rule based on the declared lower or upper bounds implies the corresponding true inequality. The argument does not establish validity of a biased off-policy estimator or a misspecified estimand; those are premises that must be audited separately. The result concerns acceptance time, and later environment changes can still invalidate performance.

An explicit one-sided allocation. For bounded observations $X_t \in [l,h]$ with a fixed conditional mean under the sampling filtration, a one-sided time-uniform Hoeffding radius is

$$r_n = (h-l)\sqrt{\frac{\log(n(n+1)/\alpha)}{2n}}.$$

It uses time-wise error spending $\alpha/[n(n+1)]$. To cover both lower and upper tails jointly with error α, replace the logarithm by $\log(2n(n+1)/\alpha)$. The historical primitive returns both endpoints using the one-sided radius, so its pair is not a joint $(1-\alpha)$ confidence sequence; the reported null experiments use only the lower endpoint and retain their one-sided interpretation. The controller uses one required direction per declared metric, which must be allocated accordingly.

For a predeclared K candidates, J required bound directions, and P monitoring phases, assigning $\alpha_{w,k,j,p} = \delta/[w(w+1)KJP]$ gives total error at most δ. Candidate identity, policy version, and estimand must be fixed during each stream; newly trained candidates or changed conditions require fresh allocations. The reference controller allocates within a window only, so its caller must supply the window/phase budget. Signed risk differences require support such as $[-1,1]$, not $[0,1]$, and a KL bound requires an independently justified finite range. Neither clipping nor a confidence formula supplies unbiasedness. Positive aggregate utility also does not entail task non-degradation: $\Delta J = -0.1$ and $\lambda_B \Delta B = 0.2$ can pass the utility gate.

B Additional Related Work

Relation to prior adaptation methods. Residual control and LoRA specify a restricted update family (Johannink et al., 2019; Hu et al., 2022); our criterion asks whether a chosen family admits a common repair. HDT learns a demonstration-conditioned adaptation interface (Xu et al., 2023), while batch-online learning studies policy improvement from autonomous data (Dong et al., 2026). The present ablations isolate burden and routing within a fixed implementation. They support these components, but a matched robot benchmark against complete recent systems remains necessary to assess end-to-end competitiveness.

Online and parameter-efficient robot adaptation. Residual control and low-rank adaptation restrict updates around pretrained policies (Johannink et al., 2019; Hu et al., 2022). Meta-RL and batch-online RL adapt dynamics or shared policies from deployment data (Nagabandi et al., 2019; Dong et al., 2026). Masked and modulated policies expose restricted adaptation modules (Park et al., 2026; Adalibieke et al., 2026), while WPT distills an online world model into a driving policy (Jiang et al., 2026). Closely related, Bozkurt et al. (2026) allocate interaction jointly to policy selection and fine-tuning. Our formulation associates candidate update scopes with recurrent structure and retention/reuse value. The experiments compare scope geometry and value learning rather than budget-allocation algorithms.

Routing and continual adaptation. Hyper-Decision Transformer meta-adapts policies from demonstrations (Xu et al., 2023); SEMA routes among expanding adapters under continual shift (Wang et al., 2025); and CLARE expands and routes VLA adapters without task labels (Römer et al., 2026). Budgeted policy selection additionally couples candidate evaluation with fine-tuning (Bozkurt et al., 2026). These methods motivate distinguishing policy routing from selection of parameters to update. LoopAdapt chooses the scope of a persistent update from recurrent failure structure and future retention/reuse, including the option to make no parameter change.

Progress, recovery, and safe improvement. Process supervision can expose stage progress for long-horizon manipulation (Liu et al., 2026; Tan et al., 2026; Fei et al., 2026), while runtime recovery systems retry, reset, or invoke explicit recovery behavior (Zhao et al., 2026; Ke et al., 2026; Zeng et al., 2026). LoopAdapt asks when a recurrent failure supports a persistent parameter update and at what scope. Safe policy improvement restricts weakly supported changes (Sharma et al., 2025), and confidence sequences give time-uniform coverage under repeated inspection (Howard et al., 2021). Fundamental matrices, cycle representations, and sensitivity tools predate this work (Seneta, 2006; Kalpazidou, 2006; Caswell, 2001; Kirkland et al., 2002); our contribution is their coupling to burden control, trainable scope, and persistent deployment decisions.

C Additional Method Details

Consider deployment windows $w = 1, 2, \ldots$ with current safe policy π_{θ_w}. A progress model maps history to stage-aware progress. We discretize low-progress histories into macro-states $z \in \{1, \ldots, n_w\}$ and estimate the substochastic kernel

$$Q_{ij}^{w}(\theta) = \mathbb{P}_\theta(z_{t+1} = j,\ Y_t = \text{low} \mid z_t = i, \omega_w), \tag{9}$$

where missing row mass is exit through productive progress, success, failure, ambiguity, or the abstraction boundary. For candidate update a, the immediate deployment reward is

$$\begin{aligned}\mathcal{H}_w(a) = \Delta J_w^{\text{task}}(a) - \lambda_F \Delta R_w^{\text{fail}}(a) - \lambda_R \Delta L_w^{\text{retain}}(a) \\ + \lambda_B \Delta B_w(a) - \lambda_C C_w(a).\end{aligned} \tag{10}$$

For candidate a, write $d_w(a)$ for its parameter increment and $\pi_w^a = \pi_{\theta_w + d_w(a)}$, with $d_w(\textsf{keep}) = 0$. Every non-keep action must also satisfy

$$\begin{aligned}\Delta R_w^{\text{fail}}(a) \le \tau_F, \\ \Delta L_w^{\text{retain}}(a) \le \tau_R.\end{aligned} \tag{11}$$

$$\begin{aligned}\Delta A_w^{\text{limit}}(a) \le \tau_A, \\ \mathbb{E}_{o \sim \zeta_w} D_{\text{KL}}(\pi_{\theta_w} \| \pi_w^a) \le \tau_{\text{KL}}.\end{aligned} \tag{12}$$

The KL divergence is constrained separately rather than included again in the reward. The distribution ζ_w in the KL expectation is the declared validation distribution. Δ denotes candidate minus checkpoint for task and risk terms; ΔB_w is a reduction, defined below, and all weights are nonnegative. C_w is incremental construction/validation interaction cost. Past probing spend is charged once when incurred; remaining action-dependent cost enters the decision. The contact diagnostic uses fixed action penalties rather than measured C_w.

Scope selection acts at a slower timescale than the policy. The router state x_w^{R} summarizes progress, shift severity, burden structure, repairability, uncertainty, protected-condition risk, candidate cost, and adapter-retrieval features. Its variable action set is

$$\mathcal{A}_w^{\mathrm{R}} = \{\mathsf{keep}, \mathsf{global}\} \cup \{\mathsf{local}(c) : c \in \mathcal{C}_w\}. \tag{13}$$

The optimal adaptation value is defined by Equation (8), with $x_w = x_w^{\mathrm{R}}$, bounded rewards, $0 \le \gamma_R < 1$, and optimization over subsequent admissible routing policies. This makes delayed forgetting and recurring-condition reuse part of the routing target.

C.1 Candidate Generation

The controller accepts a declared proposal policy. Algorithm 1 supplies a common-descent proposal for a fixed repair target; multiple predeclared targets can supply a candidate bank. The aggregate proposals below define the alternative used in the scalar-diagnostic component experiments. Their average-descent property does not imply the loop-wise bound. Any further optimization of a common-descent proposal requires a new bound before that property can be claimed.

Figure 2 shows the proposed procedure. The structural layer defines burden and update capacity; the adaptation layer assigns persistent value; certification restricts admissibility and validates commitment. Burden reduction, task improvement, predictive value, and statistical evidence are different quantities. Each requires its own evaluation rather than being inferred from another component's success; Table 1 identifies the implemented components.

C.2 Loop Burden

Assume the estimated low-progress chain is transient, $\rho(Q) < 1$, and let $N = (I - Q)^{-1}$. For initial distribution μ, the expected pre-exit occupancy is $\eta^{\mathsf{T}} = \mu^{\mathsf{T}} N$. For a canonically anchored simple directed loop $c = (i_0, i_1, \ldots, i_{\ell-1}, i_0)$ define

$$\begin{aligned} p_c(Q) &= \prod_{r=0}^{\ell-1} Q_{i_r i_{r+1}}, \\ \mathcal{B}_c(Q;\mu) &= \eta_{i_0} p_c(Q). \end{aligned} \tag{14}$$

Theorem 3 (Expected anchored loop burden)**.** If N_c counts completed anchored traversals of c before the first exit from the low-progress process, then

$$\mathbb{E}[N_c] = \mathcal{B}_c(Q;\mu) = \left[\mu^{\mathsf{T}}(I-Q)^{-1}\right]_{i_0} \prod_{r=0}^{\ell-1} Q_{i_r i_{r+1}}. \tag{15}$$

The identity separates access to the loop anchor from persistence around the loop. For a frozen candidate set $\mathcal{C}_w$, initial distribution μ_w, and nonnegative weights ν_c summing to one, we optimize

$$\begin{aligned} \Phi_w^{\mathrm{B}}(\pi) &= \sum_{c \in \mathcal{C}_w} \nu_c \log(1 + \mathcal{B}_c(Q_w^{\pi};\mu_w)), \\ \Delta B_w(a) &= \Phi_w^{\mathrm{B}}(\pi_{\theta_w}) - \Phi_w^{\mathrm{B}}(\pi_w^a), \end{aligned} \tag{16}$$

with $\Phi^{\mathrm{B}} = 0$ if no recurrent structure is retained. For selected loops with positive burden,

$$\frac{\partial \log \mathcal{B}_c}{\partial \theta_k} = \frac{\mu^{\mathsf{T}} N (\partial Q/\partial \theta_k) N e_{i_0}}{\eta_{i_0}} + \sum_{r=0}^{\ell-1} \frac{\partial \log Q_{i_r i_{r+1}}}{\partial \theta_k}. \tag{17}$$

The derivative holds with fixed abstraction, anchors, weights, and initial distribution. Canonical anchoring removes duplicate cycle representations but does not make the burden invariant to the

choice of anchor. In differentiable references, gradients pass through the estimated kernel; extending them to black-box dynamics requires a separately validated policy-gradient estimator. Appendix C.10 specifies an actor–critic interface, whose black-box end-to-end use is not evaluated here.

Remark 4 (Burden is not a success surrogate). Because failure and productive progress both remove probability mass from the transient low-progress kernel, minimizing Φ^{B} alone can prefer rapid failure to repeated stagnation. We therefore use burden only jointly with task return and explicit failure constraints in Equations (10) and (11); empirical claims always report task and burden outcomes together.

C.3 Aggregate Repairability as a Diagnostic Special Case

Let $g = \nabla_\theta \Phi^{\mathrm{B}}_w$ and $M \succ 0$ be the shared local trust-region metric (a declared positive metric in the controlled studies; a damped Fisher is a possible policy implementation). For a trainable tangent subspace $\mathcal{S}$ with full-column-rank basis $U_{\mathcal{S}}$, define

$$\Lambda_{\mathcal{S},M} = \frac{g^{\mathsf{T}} U_{\mathcal{S}} (U_{\mathcal{S}}^{\mathsf{T}} M U_{\mathcal{S}})^{-1} U_{\mathcal{S}}^{\mathsf{T}} g}{g^{\mathsf{T}} M^{-1} g + \epsilon}. \tag{18}$$

Here $\epsilon > 0$ is a denominator stabilizer, distinct from the gradient-error bound ε. This is a regularized fraction of first-order burden-descent energy available inside the restricted scope under the same step geometry as the full update. It is invariant to a change of basis within $\mathcal{S}$ and reduces to Euclidean projected-gradient energy for $M = I$ with an orthonormal basis.

Proposition 5 (Repairability is relative quadratic capacity)**.** Let $g \neq 0$, $v = M^{-1} g$, $P_{\mathcal{S}}^{M} = U_{\mathcal{S}} (U_{\mathcal{S}}^{\mathsf{T}} M U_{\mathcal{S}})^{-1} U_{\mathcal{S}}^{\mathsf{T}} M$, $\|x\|_M^2 = x^{\mathsf{T}} M x$, and $q(d) = g^{\mathsf{T}} d + \frac{1}{2} d^{\mathsf{T}} M d$. The unregularized score satisfies

$$\widetilde{\Lambda}_{\mathcal{S},M} = \frac{\|P_{\mathcal{S}}^{M} v\|_M^2}{\|v\|_M^2} = \frac{-\min_{d \in \mathcal{S}} q(d)}{-\min_d q(d)} \in [0, 1]. \tag{19}$$

The implemented score is $\Lambda_{\mathcal{S},M} = \widetilde{\Lambda}_{\mathcal{S},M}\, E/(E+\epsilon)$ with $E = g^{\mathsf{T}} M^{-1} g$, and hence is a conservative shrinkage of this capacity fraction.

Proposition 6 (Metric-scoped burden descent)**.** If $L_{B,M} > 0$ and, for $\|d\|_M \leq R$,

$$\Phi^{\mathrm{B}}(\theta + d) \leq \Phi^{\mathrm{B}}(\theta) + g^{\mathsf{T}} d + \frac{L_{B,M}}{2} d^{\mathsf{T}} M d, \tag{20}$$

then for $d_{\mathcal{S}} = -\eta U_{\mathcal{S}} (U_{\mathcal{S}}^{\mathsf{T}} M U_{\mathcal{S}})^{-1} U_{\mathcal{S}}^{\mathsf{T}} g$ with $0 < \eta < 2/L_{B,M}$ and $\|d_{\mathcal{S}}\|_M \leq R$,

$$\Phi^{\mathrm{B}}(\theta + d_{\mathcal{S}}) \leq \Phi^{\mathrm{B}}(\theta) - \eta \left(1 - \frac{L_{B,M}\eta}{2}\right) g^{\mathsf{T}} U_{\mathcal{S}} (U_{\mathcal{S}}^{\mathsf{T}} M U_{\mathcal{S}})^{-1} U_{\mathcal{S}}^{\mathsf{T}} g. \tag{21}$$

This score concerns local quadratic capacity. Under an equal M-norm radius, the ratio of maximum linear decreases is $\sqrt{\widetilde{\Lambda}_{\mathcal{S},M}}$, not $\widetilde{\Lambda}_{\mathcal{S},M}$. It is neither a task-improvement guarantee nor a finite-optimization oracle. Distinct adapters require a common tangent representation and metric before this comparison is meaningful.

C.4 Matched Candidate Updates

A local action inserts a zero-initialized low-rank latent adapter gated by stage/condition/loop proximity while freezing the shared backbone; a global action exposes shared control parameters. For the aggregate proposal policy, both start from the same checkpoint. Let $\widehat{\widetilde{G}}_w$ estimate task gain minus retention, failure, and incremental cost penalties, excluding burden. The common aggregate objective is:

$$\Delta\vartheta_{\mathcal{S}} = \operatorname*{arg\,max}_{\Delta\vartheta \in \mathcal{S}} \widehat{\widetilde{G}}_w(\Delta\vartheta) + \lambda_B \left[\widehat{\Phi^{\mathrm{B}}}_w(\pi_{\theta_w}) - \widehat{\Phi^{\mathrm{B}}}_w(\pi_{\theta_w + \Delta\vartheta})\right], \tag{22}$$

subject to the same declared constraints. This defines a comparison in which the trainable subspace changes while the learning objective and interaction cap are held fixed. PointMass instantiates shared-policy fine-tuning. The contact diagnostic instead compares a 33-parameter gated gripper residual with a 12-parameter affine adapter over all outputs; both freeze the backbone and use different retention weights. It therefore tests output-support choices rather than isolating shared-parameter update scope.

C.5 Candidate Selection and Validation

For candidate a, feature vector e_a includes action type, burden, access/persistence gradients, $\Lambda_{\mathcal{S}_a,M}$, predicted task/failure/retention changes, interaction cost, parameter count, retrieval similarity, and confidence widths. Global features u_w summarize the window. A shared encoder with set pooling gives candidate-wise values

$$\widehat{\mathcal{Q}}_\psi(x_w^{\mathrm{R}}, a) = f_\psi\big(e_a, \mathrm{Pool}_{b\in\mathcal{A}_w^{\mathrm{R}}}\, g_\psi(e_b), u_w\big)\,, \tag{23}$$

which is permutation-equivariant to candidate reordering. The synthetic reference uses masked mean pooling and tests padding invariance. Meta-training uses observed counterfactual transitions from training-only streams; exact adaptation values are used for evaluation, never as deployment labels. Candidate features in this generator include simulated immediate gains and retention risks, so the experiment conditions on informative value-related features. Fitted Q-learning uses

$$\begin{aligned} Y &= \mathcal{H}_w(a) + \gamma_R \max_{b\in\mathcal{A}_{w+1}^{\mathrm{R}}} \widehat{\mathcal{Q}}_{\bar\psi}(x_{w+1}^{\mathrm{R}}, b), \\ \mathcal{L}_R &= \mathbb{E}_{(x,a,u,x')\sim\mathcal{D}_R}\left[\mathrm{Huber}\Big(\widehat{\mathcal{Q}}_\psi(x,a) - Y\Big)\right]. \end{aligned} \tag{24}$$

Here $\mathcal{D}_R$ is the training transition distribution, $u = \mathcal{H}_w(a)$, $x = x_w^{\mathrm{R}}$, $x' = x_{w+1}^{\mathrm{R}}$, and $\bar\psi$ denotes frozen target parameters. Terminal transitions use $Y = u$; the displayed target applies to continuing transitions. Synthetic OOD studies hold out generator families. The contact scorer is a separate supervised regressor on same-condition repeated-evaluation values, with three fixed action types and no set pooling or Bellman targets; it does not test this multi-window learning rule.

Choice and certification are separated. For candidate-specific time-uniform lower/upper bounds, the shield retains

$$\mathcal{A}^{\mathrm{cert}} = \{\mathsf{keep}\} \cup \{a \in \mathcal{A}_w^{\mathrm{R}} \setminus \{\mathsf{keep}\} : L_a^{\mathcal{H}} > 0,\ U_a^F \le \tau_F,\ U_a^R \le \tau_R,\ U_a^A \le \tau_A,\ U_a^{\mathrm{KL}} \le \tau_{\mathrm{KL}}\}, \tag{25}$$

and deploys $\widehat{a} = \arg\max_{a\in\mathcal{A}^{\mathrm{cert}}} \widehat{\mathcal{Q}}_\psi(x,a)$ only after independent shadow validation. keep is always available; post-deployment violations trigger rollback.

Proposition 7 (Routing error within the certified set)**.** If $|\widehat{\mathcal{Q}}_\psi(x,a) - \mathcal{Q}^\star(x,a)| \le \epsilon_Q$ for all $a \in \mathcal{A}^{\mathrm{cert}}$, then

$$\mathcal{Q}^\star(x,\widehat{a}) \ge \max_{a\in\mathcal{A}^{\mathrm{cert}}} \mathcal{Q}^\star(x,a) - 2\epsilon_Q. \tag{26}$$

If the best safe action is certified, the same bound holds relative to that action.

The bound separates value approximation from shield exclusion. It is a single-decision bound conditional on uniform value accuracy, not a return guarantee for the policy induced by repeatedly applying the router.

Theorem 8 (Repeatedly monitored accepted updates)**.** Suppose every acceptance statistic has a time-uniform confidence sequence valid under the actual adaptive data-collection filtration, and the error allocation over all windows, candidates, constraints, and monitoring phases sums to at most δ. Then, with probability at least $1-\delta$, every accepted update satisfies its declared aggregate-utility, failure, retention, action-limit, and trust-region inequalities at acceptance time despite adaptive stopping and repeated inspection.

Positive aggregate utility need not imply nonnegative task gain; such a claim requires a separate task-gain bound. Confidence validity, bounded/unbiased estimands, and a summable allocation across windows are premises. The theorem does not establish these premises for biased off-policy estimates or the contact shadow heuristic. Proofs and explicit allocation details are in the appendix.

C.6 Estimation Stability

The fundamental matrix can be sensitive when expected exit time is large. The following result makes the operational condition explicit.

Proposition 9 (Perturbation stability)**.** Let Q and $\widehat{Q} = Q + \Delta$ be nonnegative substochastic kernels, $\rho(Q) < 1$, and $N = (I-Q)^{-1}$. For any submultiplicative matrix norm, if $\|N\|\,\|\Delta\| < 1$, then

$\widehat{N} = (I - \widehat{Q})^{-1}$ exists and

$$\|\widehat{N} - N\| \le \frac{\|N\|^2 \, \|\Delta\|}{1 - \|N\| \, \|\Delta\|}. \tag{27}$$

For an anchor with $\eta_{i_0} > 0$ and $\widehat{\eta}_{i_0} > 0$, if all edges of c satisfy $Q_{ij} \ge q_{\min} > 0$ and $\|\Delta\|_{\max} \le q_{\min}/2$, then

$$|\log \widehat{\mathcal{B}}_c - \log \mathcal{B}_c| \le \frac{|(\mu^{\mathsf{T}}(\widehat{N} - N))_{i_0}|}{\min(\eta_{i_0}, \widehat{\eta}_{i_0})} + \frac{2\ell \|\Delta\|_{\max}}{q_{\min}}. \tag{28}$$

The bound motivates monitoring $\|N\|$, supported edge mass, and uncertainty jointly. In the controlled conditioning diagnostic, $\|N\|$ alone has weak predictive value. Structural instability calls for abstention or additional evidence; it does not certify a global candidate.

C.7 Loop Discovery

The positive-support graph $G_Q = (V, E)$ contains edge (i, j) when the lower confidence bound on Q_{ij} exceeds the support threshold. We enumerate simple directed cycles up to length $L_{\max}$ using a bounded Johnson-style search. Each rotational equivalence class receives a canonical anchor based on the lexicographically smallest tuple of stage, progress bin, and state identifier. Reverse traversals remain distinct because they represent different dynamics.

A candidate score balances burden, relevance, and confidence:

$$S(c) = \mathrm{LCB}(\log(1 + \mathcal{B}_c)) \cdot \widehat{\Gamma}_c \cdot \widehat{A}_c, \tag{29}$$

where $\widehat{\Gamma}_c$ is the fraction of low-progress occupancy touching the candidate and $\widehat{A}_c$ is a preliminary alignment score between the loop gradient and common utility. Candidates with low burden, unstable support, or non-positive alignment are discarded. The search retains at most B cycles and reports the covered low-progress occupancy

$$\Gamma_{L,B} = \sum_{i \in U_{L,B}} \frac{\eta_i}{\sum_j \eta_j}, \tag{30}$$

where $U_{L,B}$ is the union of states touched by retained candidates. Low coverage forbids claims that the discovered set exhausts the recurrent structure.

C.8 Local Adaptation

Let a pretrained policy contain a representation $z = f_{\theta_b}(o, g)$ and an action generator $\pi_{\theta_h}(a \mid z)$. For loop c, the local branch inserts a low-rank adapter

$$\begin{gathered} \widetilde{z} = z + s_c(h_t) \left(U_{c,w} V_{c,w}^{\mathsf{T}} z + b_{c,w} \right), \\ \mathrm{rank}(U_{c,w} V_{c,w}^{\mathsf{T}}) \le r, \end{gathered} \tag{31}$$

where $s_c(h_t) \in [0, 1]$ gates the adapter by stage, condition, and proximity to the selected recurrent structure. For $z \in \mathbb{R}^m$, $U_{c,w}, V_{c,w} \in \mathbb{R}^{m \times r}$ and $b_{c,w} \in \mathbb{R}^m$. Shared parameters θ_b, θ_h remain frozen. Zero output recovers the base policy; initialize one low-rank factor to zero and the other nonzero, with zero bias, to avoid blocking both factor gradients.

The local proposal optimizes task utility and the decrease in aggregate loop burden,

$$\begin{gathered} \Delta\alpha_c = \underset{\Delta\alpha \in \mathcal{S}_c}{\arg\max} \, \widehat{\widetilde{G}}_w(\Delta\alpha) + \lambda_B \left[\widehat{\Phi^{\mathrm{B}}}_w(\pi_{\theta_w}) - \widehat{\Phi^{\mathrm{B}}}_w(\pi_{\theta_w + \Delta\alpha}) \right] - \lambda_s \|s_c\|_1 \\ \text{s.t. constraints in Equations (11) and (12),} \end{gathered} \tag{32}$$

where $\mathcal{S}_c$ is the adapter parameter subspace. Candidate initialization and ranking use $\Lambda_{\mathcal{S}_c, M}$ from Equation (18), burden magnitude, and alignment between burden descent and task-utility ascent. The loop-specific gate still encourages spatial locality, but the optimized structural objective is the aggregate potential over the affected recurrent component; increasing another retained cycle contributes to the average penalty, but can be outweighed by improvements elsewhere. The loop-wise extension in the main paper addresses this limitation. A newly created cycle outside the frozen set can be missed and requires a separate post-update structure audit.

The method does not depend on latent adapters specifically. Action residuals, parameter masks, low-rank weight updates, or skill-specific modules can be substituted, but the intervention interface must expose a compact restricted parameterization and an exact zero-update baseline so projected repairability and off-region retention remain interpretable.

C.9 Global Adaptation

The global branch updates shared control parameters using all recent productive, low-progress, and failed experience together with protected replay. It optimizes the same burden-aware objective as the local branch,

$$\Delta\theta_{\text{glob}} = \underset{\Delta\theta\in\mathcal{S}_{\text{glob}}}{\arg\max}\ \widehat{\widetilde{G}}_w(\Delta\theta) + \lambda_B\left[\widehat{\Phi^{\text{B}}}_w(\pi_{\theta_w}) - \widehat{\Phi^{\text{B}}}_w(\pi_{\theta_w+\Delta\theta})\right] \quad \text{s.t. constraints in Equations (11) and (12).} \tag{33}$$

Unlike the local proposal, the global branch uses the full active recurrent set and may change both loop access and escape behavior throughout the shared policy. To make scope comparisons meaningful, local and global branches share the same safe checkpoint, replay access, task utility, burden estimator, safety constraints, and counted interaction budget. This is the intended common-objective comparison; study-specific departures and the available timing evidence are disclosed below.

C.10 Actor–Critic Objectives

This subsection specifies a generic actor–critic candidate interface and the accompanying objective-level reference primitives. It is not the optimizer used by the reported differentiable contact diagnostic. To isolate scope in a future black-box comparison, each candidate should start from the same checkpoint and receive the same replay, objective coefficients, update schedule, and interaction cap.

For transition $x_t = (h_t, a_t, r_t, h_{t+1}, d_t)$, define a scalar constrained-learning reward

$$\widetilde{r}_t = r_t^{\text{task}} - \lambda_f r_t^{\text{fail}} - \lambda_n r_t^{\text{np}} - \lambda_A r_t^{\text{limit}}, \tag{34}$$

Here $d_t \in \{0,1\}$ indicates termination, $0 \le \gamma < 1$, and $\alpha_{\text{ent}} \ge 0$ weights entropy. The barred critics are frozen target networks. The non-progress term is zero on productive transitions and positive on low-progress transitions. Hard failure, retention, and trust-region checks remain separate from this shaped reward. Two candidate-specific critics Q_{ψ_1}, Q_{ψ_2} are trained with clipped double-Q targets

$$y_t^Q = \widetilde{r}_t + \gamma(1-d_t)\mathbb{E}_{a'\sim\pi_a(\cdot|h_{t+1})}\left[\min_{j\in\{1,2\}} \bar{Q}_{\bar{\psi}_j}(h_{t+1}, a') - \alpha_{\text{ent}}\log\pi_a(a' \mid h_{t+1})\right], \tag{35}$$

$$\mathcal{L}_Q = \sum_{j=1}^{2}\mathbb{E}_{x_t\sim\mathcal{D}_w\cup\mathcal{D}_{\text{retain}}}\left(Q_{\psi_j}(h_t,a_t) - y_t^Q\right)^2. \tag{36}$$

The actor objective makes burden control explicit for both scopes,

$$\begin{aligned}\mathcal{L}_\pi(a) = \mathbb{E}_{h\sim\mathcal{D}_w,\, a'\sim\pi_a(\cdot|h)}&\left[\alpha_{\text{ent}}\log\pi_a(a' \mid h) - \min_j Q_{\psi_j}(h,a')\right] \\ &+ \beta_{\text{safe}}\mathbb{E}_{h\sim\mathcal{D}_w} D_{\text{KL}}(\pi_{\text{safe}}(\cdot \mid h)\|\pi_a(\cdot \mid h)) \\ &+ \beta_{\text{retain}}\mathbb{E}_{h\sim\mathcal{D}_{\text{retain}}} D_{\text{KL}}(\pi_{\text{safe}}(\cdot \mid h)\|\pi_a(\cdot \mid h)) \\ &+ \lambda_B\widehat{\Phi^{\text{B}}}_w(\pi_a) + \mathbb{I}\{a = \textsf{local}(c)\}\lambda_s\|s_c\|_1.\end{aligned} \tag{37}$$

In this subsection, a indexes a candidate update and a_t, a' are environment actions. All loss coefficients are nonnegative; $\|s_c\|_1$ denotes mean gate activation on the candidate replay distribution. The local candidate updates only $(U_{c,w}, V_{c,w}, b_{c,w})$ and its gate s_c; the global candidate updates the shared control backbone and action head. In a high-dimensional vision implementation, freezing the observation encoder in both branches would isolate control-scope adaptation from large-scale visual fine-tuning. The present controlled studies are state based and do not report a vision or full-backbone comparison.

Algorithm 2 Persistent controller with a declared proposal policy

Require: Pretrained policy π_{θ_0}, progress model r_φ, router $\widehat{\mathcal{Q}}_\psi$, protected replay $\mathcal{D}_{\mathrm{retain}}$, error budget δ

Ensure: Validated checkpoint sequence and adapter library, subject to the stated evidence assumptions.

Initialize safe checkpoint $\theta_{\mathrm{safe}} \leftarrow \theta_0$ and empty validated adapter library
for deployment window $w = 1, 2, \ldots$ **do**
  Collect trajectories; audit progress calibration and local stationarity
  if audit fails **then**
    retain current policy; collect data or refresh representation; continue
  end if
  Estimate Q_w, N_w, support, candidate loops, burdens $\mathcal{B}_c$, and aggregate potential Φ_w^{B}
  Compute loop gradients and the scope scores required by the declared proposal policy
  Build router state x_w^{R} and score keep, global, and local(c) actions with $\widehat{\mathcal{Q}}_\psi$
  Prioritize candidate construction/probes by learned value, unresolved confidence width, and interaction cost
  Construct candidates using Algorithm 1 or the declared aggregate objective
  Maintain anytime confidence sequences for $\mathcal{H}_w$ and every hard constraint
  Form certified set with Equation (25) and choose $\widehat{a}_w = \arg\max_{a \in \mathcal{A}_w^{\mathrm{cert}}} \widehat{\mathcal{Q}}_\psi(x_w^{\mathrm{R}}, a)$
  if $\widehat{a}_w =$ keep **then**
    retain θ_{safe}; continue deployment/evidence collection
  else
    run independent shadow validation on current and protected conditions
    if all validation bounds pass **then**
      deploy $\widehat{a}_w$; archive checkpoint, burden change, router features, cost, and confidence bounds
      update θ_{safe} and store validated local adapter when applicable
    else
      reject candidate and retain θ_{safe}
    end if
  end if
  Continue anytime monitoring; rollback on any declared violation
end for

Independent candidate critics and matched optimizer/replay settings are requirements of this proposed scope-isolation experiment. They do not describe the contact experiment, which uses differentiable rollouts and scope-dependent retention/consistency terms. No reported black-box run establishes the full actor–critic path in this subsection.

C.11 Online Algorithm

Algorithm 2 summarizes the proposed controller, including validation and rollback. Executed component studies are distinguished in Table 1.

C.12 Computational Complexity

For n active macro-states, dense inversion costs $O(n^3)$; sparse solves can obtain occupancies without forming the inverse. Bounded simple-cycle enumeration remains exponential in the worst case. The reference caps cycle length/count, and candidate-set scoring is linear in the number of candidates. A complete deployment ledger must count probes, candidate training, rejected candidates, shadow checks, and rollback diagnostics; compute and parameter storage should be reported separately. Current artifacts include router-only CPU timing, not end-to-end accelerator or candidate-construction cost.

D Assumptions and Limitations

Locally stationary windows and change detection. The theory requires a sufficiently stationary data-generating process during certification. A deployment implementation can monitor progress residuals and kernel rows, then close a window on a detected shift. The executed controlled studies use known window boundaries and do not measure detector delay, false alarms, or late-detection cost.

Progress-model error. Progress estimators can confuse necessary backtracking with regression. Ranking, stage calibration, and false-stagnation audits on successful and recovering trajectories would help diagnose this failure. The contact artifacts provide scalar progress regression error, low-progress AUC, and controlled noisy/aliased comparators; they do not provide a physical backtracking benchmark or stage-boundary F1.

State aliasing and physical meaning. Expected loop burden is exact for the specified transient Markov model and fixed anchor. Physical interpretation additionally requires an adequate history abstraction. Markov-residual, history-length, and state-resolution audits are proposed checks, not completed evidence here. The planar contact statistic is a separate soft stagnation/reversal proxy and does not inherit the exact traversal-count interpretation.

Support instability and near reducibility. The perturbation bound in Proposition 9 deteriorates when $\|N\|$ is large, the spectral gap closes, or loop edges approach zero support. A deployment implementation should monitor these quantities and retain the current policy when the estimates are unstable.

Off-policy and adaptive-data bias. A black-box extension must account for adaptive off-policy data, importance weighting, and effective sample size. Merely splitting or cross-fitting data does not remove clipping or model bias. The executed differentiable references do not validate these off-policy estimators.

Retention under finite memory. Finite protected replay cannot establish retention on every past condition. The contact study evaluates nominal success; the PointMass reuse sweep tests a fixed stored adapter at a finite set of severities. Neither result establishes indefinite memory retention or a condition-stratified reservoir deployment system.

E Experimental Details

Evidence map. The experiments isolate four parts of the framework: loop geometry, burden-aware control, multi-window routing, and planar contact. PointMass differentiates the fundamental-matrix potential through a soft estimated kernel. Planar contact uses a separate soft proxy and supervised action-value regression. Table 3 states the independent unit and the claim supported by each study; the studies are component evaluations rather than one integrated physical deployment.

Isolation and leakage control. Kernel and synthetic-router partitions use separate generation/training/test seeds. Contact uses three development and five final policy seeds with no overlap, but only one fitted progress model and one primary fitted router. The non-nominal final parameter tuples differ from development tuples; their families are shared. Development nominal and final benign parameters are identical. The common initial-state evaluation seeds are reused across policies and methods for pairing, so the contact intervals condition on this fixed evaluation design.

Interaction accounting. Contact methods share the same base policy and a bank containing one trained local and one trained global candidate per seed–shift group. Saved per-method ledgers assign 1,280 shifted and 640 nominal training rollouts, 192 common-evidence rollouts, 384 shadow rollouts, and 768 final rollouts. The implementation builds a shared candidate bank rather than five independent method runs and also evaluates each of three candidate actions on 384 future rollouts omitted from those ledgers. Static call reconstruction gives 8,640 rollouts per group for the shared primary candidate bank, including evidence, both candidate fits, and shadow/final/future evaluations;

Table 3: Experimental implementations, independent units, and measured outcomes.

Study	Independent unit	Reported role
Common-descent kernels	128 independently generated kernels	loop-wise feasibility and nonlinear update outcomes
Aggregate transient kernels	96 independently generated kernels	burden descent, finite-data error, and repairability calibration
Known-value adaptation MDP	10 independently trained routers; 300 held-out windows each	routing regret, horizon interventions, OOD and noise diagnostics
PointMass	5 independently pretrained policies	end-to-end expected-burden optimization and scope feasibility
E2E planar soft contact	5 independently pretrained policies across 4 held-out shifts	learned progress, soft proxy, output adapters, supervised scoring, nominal success, and threshold veto
Family-label contact diagnostic	5 independently pretrained policies across 4 held-out shift families	equal-interaction candidate construction and predeclared family-rule commitment
Repeated monitoring	independently generated bounded streams	false-acceptance and certification-delay diagnostics

it excludes development, calibration, and ablations. These counts describe executed calls rather than a deployment trace and therefore do not support an interaction-savings claim.

Statistics and reporting. Intervals use independent kernels, trained synthetic routers, or pretrained policy seeds. Contact first averages its four shifts within each of five policy seeds. Its historical percentile bootstrap intervals are retained, alongside paired t intervals and exhaustive sign-randomization sensitivity analyses. The latter require their own normality or sign-exchangeability assumptions; none gives assumption-free inference from five policies. Shared candidate outcomes are paired measurements, not independent method replications.

Additional high-fidelity diagnostic. A 25-group Shadow Hand diagnostic is excluded from the performance comparison because all four methods reach 100% cumulative goal success and the router always selects the local branch. Those outcomes reproduce the fixed-local aggregate and do not distinguish scope selection. We retain the Isaac Sim material only as the kinematic replay in Figure 5; a discriminating study would need per-budget completions, rotation error, non-progress, and non-degenerate scope choices.

E.1 Kernel Experiment

The generator, seeds, and evaluation radii were fixed before execution; the protocol was not externally preregistered. Seeds are 27000–27127. For each seed, five-by-six logits are standard normal with $+1$ added to diagonal low-progress logits and $+0.5$ to the exit column. A row softmax yields a transient kernel. The initial distribution is uniform and loops are $(0, 1, 0)$, $(1, 2, 1)$ and $(2, 3, 2)$. The 30-dimensional diagonal metric has log-diagonal entries drawn from $\mathcal{N}(0, 0.5^2)$; a random orthogonal basis defines nested dimensions 2, 8 and 30. Gradients differentiate the actual $\log(1 + \mathcal{B}_c)$ losses through the kernel.

Normalized mean and common directions use the same radii 0.02, 0.10 and 0.30. Each method's outcomes are recomputed on its nonlinear updated kernel. A decrease below -10^{-10} counts as a regression; a minimum decrease above 10^{-10} counts as all-loop improvement; a zero update is an abstention. The independent unit is a kernel; 2,304 method/scope/radius rows are not 2,304 independent environments. All rows are released, with no selection or rerun based on their outcomes.

The small reference solver enumerates faces of the three-gradient convex hull and solves the constrained quadratic problem on each face. Its complexity is exponential in the number of loops, so this implementation is intended for the three-loop experiment. A projected-gradient or conic solver

would be needed for larger retained sets. Maximum finite-difference directional-gradient discrepancy is 3.24×10^{-10}; maximum numerical primal–dual gap is below 8.70×10^{-10}. Six independent tests check the analytic separation, one-loop reduction, rotation invariance/nesting, bounded gradient errors with exact quadratic curvature, abstention/selection, and a dense angular-grid comparison.

The 80%-of-full-margin selector and the 80%-of-aggregate-energy selector both choose dimension 30 on every kernel. This setting does not reduce scope dimension. Figure 3 evaluates the application of common descent to loop losses.

E.2 PCGrad comparison protocol

The added comparison reuses the original 128 kernels, metric-whitened scopes, analytic derivatives, and radii without changing the generator. PCGrad (Yu et al., 2020) sequentially projects each gradient against the original gradients when their inner product is negative, then averages the projected vectors. All nonzero method directions are normalized to the same metric radius. With three losses, we exhaust the six common projection orders and average outcomes within each kernel; an order is not an independent kernel. The sequential projection rule follows the authors' reference implementation, with the ordering made explicit. Zero original gradients are skipped to avoid division by zero; a zero final direction abstains.

This extension is a development comparison, not a new preregistered study. All 6,912 order/scope/radius outcomes are released. Regeneration of every original common-descent outcome agrees exactly with the saved study. Independent checks cover compatible gradients, opposed gradients, and a two-gradient analytic projection. At radius 0.10, PCGrad's order-averaged regression rates are 24.609375%, 3.125%, and 0% in dimensions 2, 8, and 30. Respectively 32, 8, and 0 kernels have at least one regressing order. No PCGrad direction abstains. The common-descent row is the classical MGDA solution in the same coordinates, not a separate algorithm claimed to outperform MGDA. The study compares update directions on exact models and does not replace robot-level comparisons to recent adaptation systems.

E.3 Figure provenance and statistical display

Native 3D scene replay. Figure 5 reuses saved Isaac Sim viewport frames 0, 48, and 96, captured every six renderer steps after step 24 at 60 Hz (0.4, 5.2, and 10.0 s; episode 0). The original renderer applies forward kinematics to recorded target joints (blue base) and recorded policy-action joints (yellow base). Object motion uses a scripted attachment after a designated grasp index. Thus the sequence illustrates scene geometry and commanded motion only; it is not contact-physics validation, a LoopAdapt rollout, or evidence of successful retrieval. Full frames are retained, with no synthesized scene content. The frame manifest records source filenames and trace indices. No new robot training or simulation run was performed to make this strip.

Figure 6 provides a qualitative hardware view that is separate from the controlled quantitative studies.

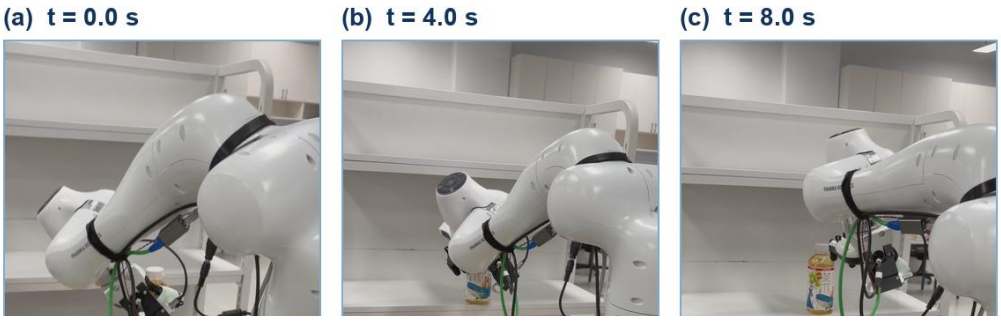


Figure 6: **Qualitative hardware replay.** Frames from a 16-s video demonstration at $t = 0.0$, 4.0, and 8.0 s, rotated clockwise for display and cropped below the robot workspace to remove the cardboard box. The strip documents the real robot and tabletop scene; it is included as deployment context and is not used to compute any quantitative result, success rate, or adaptation claim.

Figures 1 and 2 are water-blue technical schematics that separate observed evidence, scope certification, candidate ranking, and deployment validation. They describe the method rather than experimental observations. Figure 5 uses the family-label rule rollout (seed 99173); Figure 5 uses all recorded coordinates for seed 71. Figures 4 and 7 are generated from the saved per-seed and sweep observations. The diagram edits do not modify simulation frames or experimental data.

E.4 Cross-fitting and data partitions

Discovery, candidate fitting, and final validation should have explicit data partitions. The generic method permits episode-level cross-fitting when justified by the estimator. The contact implementation instead has separate named rollout-seed blocks and a common candidate bank, as specified in its protocol; it does not execute the full cross-fitting procedure.

E.5 Progress-model training and audit

In contact, a 15–64–32–1 network with SiLU hidden activations and sigmoid output regresses simulator-derived scalar progress on development state observations. It is then frozen. Proposed pairwise/stage-supervised variants are not reported experiments. Access to full simulated object state and task geometry is part of this diagnostic's information assumptions.

E.6 Change-point detection

The deployment interface can monitor progress residuals, failure rate, condition-embedding means, and selected kernel-row discrepancies. A candidate change is declared only when a predeclared sequential test remains above threshold for a persistence period. The reported controlled studies use known window boundaries, so they do not establish change-detector accuracy.

E.7 Adapter indexing and recurring conditions

The generic adapter-library interface stores validated adapters with condition and stage metadata. A PointMass sweep evaluates a stored adapter under repeated and changed severities. Contact resets to a base policy for each group and does not evaluate an evolving adapter library, retrieval decisions, or sequential rollback history.

E.8 Physical loop decoding

A domain-specific event decoder can map macro-state sequences to interpretable events such as slip onset, partial regrasp, unstable hold, or orientation loss. Such decoding should be frozen before final evaluation and reported separately from surrogate burden and task outcome. The current experiments do not validate a learned physical-event decoder.

E.9 Router meta-training and leakage control

The synthetic router learns from training-only counterfactual transitions and is tested against known adaptation values. Its candidate features include simulator-derived gains and retention risks, so value-feature estimation is not being solved by this test. Contact uses a separately fitted supervised scorer without shift-name inputs; the earlier family-label diagnostic remains a distinct experiment.

E.10 Executable algorithm-consistency checks

The NumPy smoke core checks burden counts, derivatives, scoped descent, and a toy router/shield. PyTorch primitives implement differentiable kernels, repairability, fitted Q-learning, and certified-set selection. These checks support component consistency, not the full deployment procedure. The reproducibility checks also recompute contact aggregates and verify source consistency. Test results are recorded with the corresponding implementation version.

E.11 Reference configuration and search ranges

The generic interface permits history, progress margin, rank, cycle budget, support, and replay hyperparameters. The following ranges describe design options, not an executed comprehensive tuning sweep:

$$
\begin{gathered}
H \in \{4, 8, 16\}, \\
\epsilon_p \in \{0.02, 0.05, 0.10\}, \\
r \in \{2, 4, 8, 16\}, \\
L_{\max} \in \{3, 4, 6, 8\}, \\
B \in \{8, 16, 32\}, \\
\tau_{\sup} \in \{10^{-4}, 10^{-3}, 10^{-2}\}, \\
\varepsilon_{\text{route}} \in \{0, 0.01, 0.05\}, \\
\text{replay fraction} \in \{0.10, 0.25, 0.50\}, \\
\text{UTD} \in \{1, 2, 4\}.
\end{gathered}
$$

Per-study settings, normalization choices, random seeds, and exact commands for reported controlled runs are stored with their result artifacts. The ranges above describe the generic method interface; only values attached to executed artifacts support empirical claims.

F Additional Results

F.1 Finite-Sample and Validation Sensitivity

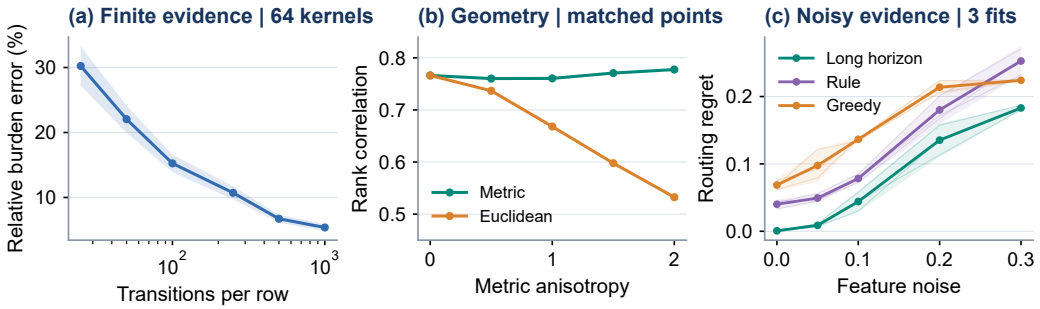


Figure 7: **Sensitivity to finite samples, metric choice, and noisy features.** (a) Relative burden-estimation error versus transitions per row. (b) Correlation between repairability and realized descent under metric anisotropy. (c) Router regret under feature noise. Thin traces show three fits; shading shows approximate normal 95% intervals, clipped at zero for nonnegative regret.

Estimation Error. Figure 7 varies sample size, metric anisotropy, and feature noise. Burden error falls from $30.24 \pm 3.04\%$ at 25 transitions per row to $5.38 \pm 0.53\%$ at 1,000. Metric repairability better tracks descent under anisotropy, but these diagnostics do not establish Theorem 2's uniform gradient-error bound. At feature noise 0.30, router regret is 0.1831 versus 0.2528 for the rule.

Repeated Validation. Over 4,000 null streams and 1,500 looks, the time-uniform bound has zero observed false acceptances versus 10.4% for reused pointwise 95% bounds. Gains of 0.10 and 0.20 are accepted in 3.65% and 97.15% of streams. This supports the simulated statistical primitive, not the contact veto; deployment still requires valid estimands and summable error budgets across candidate versions and windows.

Table 4: Self-contained PointMass continuous-control reference study (five pretrained-policy seeds; mean ±95% CI across seeds). Continuous-control simulation.

Shift	Action	Success (%)	Soft-loop Φ_B	Retention (%)
Localized	Keep	0.0±0.0	0.928±0.007	100.0±0.0
Localized	Local	100.0±0.0	0.055±0.000	100.0±0.0
Localized	Global	100.0±0.0	0.038±0.003	100.0±0.0
Diffuse	Keep	0.0±0.0	0.093±0.002	100.0±0.0
Diffuse	Local	0.0±0.0	0.029±0.000	100.0±0.0
Diffuse	Global	100.0±0.0	0.021±0.001	100.0±0.0
Benign	Keep	100.0±0.0	0.054±0.002	100.0±0.0
Benign	Local	100.0±0.0	0.015±0.000	100.0±0.0
Benign	Global	100.0±0.0	0.009±0.001	100.0±0.0

F.2 Continuous PointMass control

To complement exact-kernel and adaptation-MDP mechanism tests, we execute a self-contained differentiable PointMass environment with continuous state and action. A small policy is behaviorally cloned from a stabilizing nominal controller, then frozen into a safe checkpoint. Local candidates add a gated low-rank residual active only in the bottleneck region; global candidates fine-tune shared policy parameters. The localized shift changes actuator authority only in the gated region, the diffuse shift combines global observation/dynamics corruption with a terminal disturbance outside that gate, and benign novelty leaves the dynamics unchanged. Five independently pretrained policy seeds are used. This continuous-control environment permits differentiation through both the policy and the transition model.

For this differentiable reference, Φ^{B} is instantiated from a soft four-state low-progress abstraction over bottleneck position. Soft state memberships and low-progress weights produce a substochastic transition kernel; a fixed 6% leakage keeps the estimated process transient. The candidate set contains self-recurrence and adjacent two-state cycles, and the same fundamental-matrix expression $N = (I - Q)^{-1}$ defines their aggregate log-burden potential. Gradients therefore pass from Φ^{B} through the estimated kernel and continuous rollout into policy or adapter parameters. This is a differentiable relaxation of the discrete abstraction used in the main method, not an empirical stall-count surrogate.

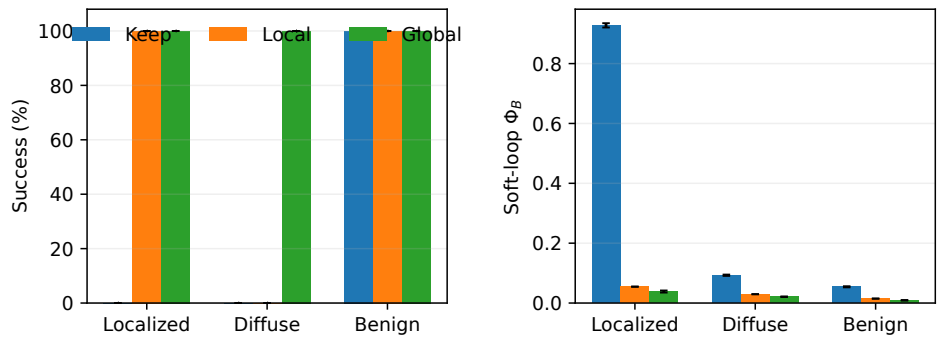


Figure 8: PointMass success and loop potential under localized, diffuse, and benign conditions.

Figure 8 shows that the local candidate recovers from the localized shift, whereas the diffuse shift requires shared-policy adaptation in this setup. Table 4 reports the means and intervals, including nominal retention.

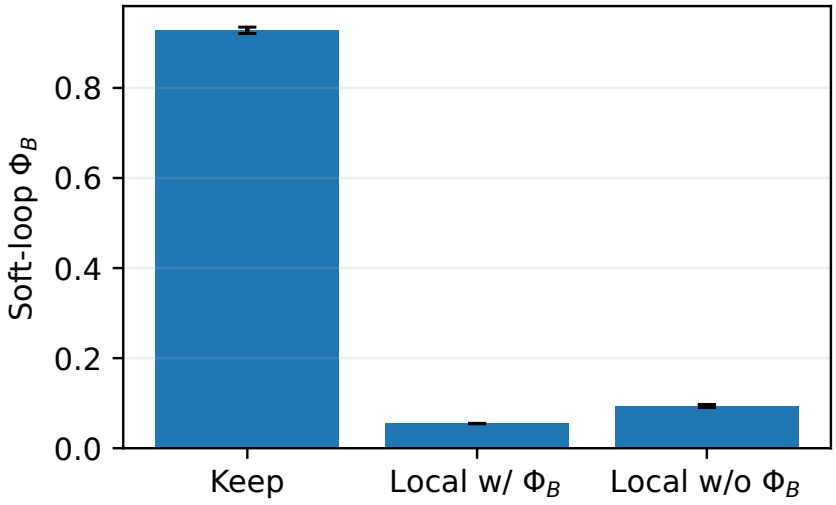


Figure 9: Effect of the loop objective on PointMass potential after task success saturates.

Figure 9 separates success from recurrent inefficiency: adding Φ^{B} lowers loop potential even when both adapted policies reach the goal.

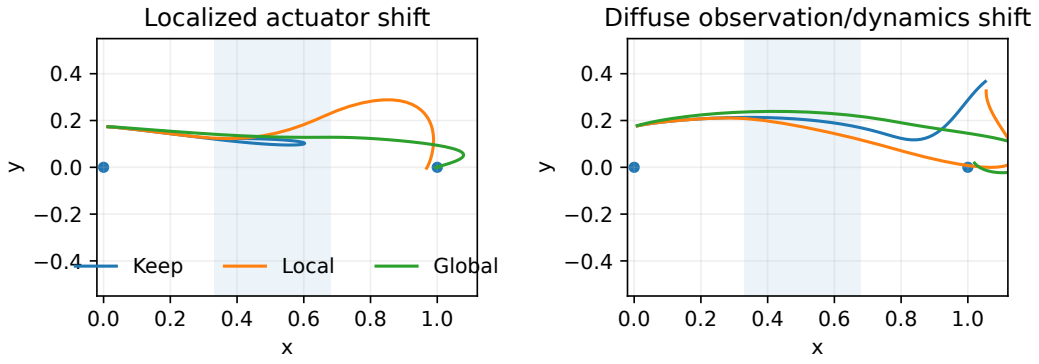


Figure 10: PointMass trajectories. Shading denotes the local adapter region; the diffuse terminal disturbance lies outside this region.

Figure 10 explains the scope difference spatially. The local residual acts inside the shaded bottleneck, while the diffuse disturbance also changes behavior near the terminal region.

F.3 Planar contact with learned scoring

This diagnostic uses differentiable planar soft-contact dynamics, scalar progress regression, and a frozen supervised action scorer. Progress/router training uses three development policies and six development conditions; evaluation uses five new policies and four conditions. Only the three non-nominal final parameter tuples are new. All four families occur in development, and final benign equals development nominal. Shift-name exclusion from router inputs is therefore distinct from generalization to an unseen family.

The local candidate is a gated gripper-output residual with 33 trainable parameters; the global candidate is an affine residual over three action channels with 12 trainable parameters. Both freeze the base policy. The local network has a two-unit nonlinear bottleneck; this is not a matrix-rank-2

guarantee for a one-dimensional output. Adam learning rates are $0.012/0.007$, nominal-loss weights $0.18/0.55$, and consistency losses compare the gripper channel/all channels respectively. Both perform 32 updates with 40 shifted and 20 nominal rollouts per update. Thus scope is confounded with architecture and loss geometry. Router descriptors $0.08/1.0$ encode nominal support and are not measured parameter-count ratios.

On 80 held-out calibration groups, progress RMSE/MAE are $0.0121/0.0092$ and low-progress AUC is 0.937. Learned burden correlates 0.984 with oracle soft burden and 0.836 with oracle low-progress fraction. A deliberately noisy comparator has RMSE 0.0459 and AUC 0.533; an aliased comparator has RMSE 0.0855 and AUC 0.919, exposing distinct ranking and calibration failure modes.

Table 5: Planar-contact results with learned scoring (mean $\pm$ standard error over five independently pretrained policy seeds; four final shifts are averaged within each seed).

Method	Shifted (%)	Nominal (%)	Distance ↓	Burden ↓
keep	25.4 ± 0.8	97.9 ± 1.5	0.253 ± 0.004	2.931 ± 0.024
fixed local	100.0 ± 0.0	100.0 ± 0.0	0.051 ± 0.003	2.437 ± 0.078
fixed global	100.0 ± 0.0	100.0 ± 0.0	0.028 ± 0.002	1.975 ± 0.122
tuned heuristic	99.8 ± 0.1	99.9 ± 0.1	0.035 ± 0.003	2.296 ± 0.107
LOOPADAPT	99.8 ± 0.1	99.9 ± 0.1	0.037 ± 0.007	2.107 ± 0.137

Table 5 reports the per-seed results. The frozen scorer selects global/local/keep on $14/5/1$ groups versus $10/6/4$ for the heuristic. Its finite-evaluation oracle agreement/regret is $60\%/0.0105$ versus $40\%/0.0153$. These estimates condition on one fitted scorer and 20 policy–condition groups. The utility difference is 0.0048 (original percentile interval $[-0.0054, 0.0115]$); success is identical at 99.84%. The mean proxy reduction over the heuristic is 0.189, but conclusions depend on the small-sample interval method. Fixed global has the lowest observed aggregate proxy and highest observed utility. The oracle chooses global/local on 15/5 groups, never keep, so the benefit of abstention is not demonstrated.

Table 6: Exploratory objective/router ablations on the local-grip and compound final shifts. Entries are descriptive means over ten seed–shift cells; the policy seed remains the inferential unit in primary comparisons.

Variant	Success (%)	Burden ↓	Regret ↓	Agree (%)
Task only	100	2.527	0.0181	30
Progress shaping	100	2.534	0.0205	30
Burden only	80	2.348	0.4368	80
Task + burden	100	2.118	0.0034	80
Burden feature only	100	2.327	0.0087	60

Table 6 compares objective and feature configurations. These ablations change candidate objectives and refit the routers; objective variants also use different training rollout seeds, and task-only removes a router feature. Burden-only retains nominal-loss and action-consistency terms. Consequently, the table compares configurations rather than isolating a single causal factor. The burden-only configuration reaches 80% success, confirming the need for the task objective. A paired factorial experiment with matched rollout/router seeds is needed to separate objective, feature, and training-noise effects.

F.3.1 IMPLEMENTED TARGET, INFORMATION ACCESS, AND STATISTICAL AUDIT

Let p_t be learned scalar progress, e_t enclosure, and $\Delta p_t = p_{t+1} - p_t$. Contact optimizes the separate proxy

$$m_t = \sigma(18(p_t - 0.18))\sigma(18(0.88 - p_t)),$$
$$\Psi = \mathbb{E}\sum_t m_t \left[\frac{\mathrm{softplus}(45(0.004 - \Delta p_t))}{45} + 1.7\frac{\mathrm{softplus}(-45\Delta p_t)}{45} + 0.22[e_t - e_{t+1}]_+\right].$$

It penalizes soft stalls, reversals, and enclosure loss. It is not an anchored-cycle count, need not vanish on strictly progressing trajectories, and no identity or gradient-equivalence result links it to Φ^{B}. Evaluation uses the same proxy with simulator-derived progress; the reported learned/oracle correlation is proxy agreement, not agreement with observed cycle counts. Repairability is not computed or used by this contact scorer.

The scorer minimizes Huber loss to targets $V_a = \widehat{U}_a^{(1)} + 0.9\widehat{U}_a^{(2)}$, where both estimates evaluate the same frozen candidate on the same shifted and nominal distributions with different initial-state seeds. Its implemented utility is

$$U_a = s_a - 0.42d_a - 0.06\Psi_a - 0.90[0.90 - s_a^{\mathrm{nom}}]_+ - c_a,$$
$$(c_{\mathrm{keep}}, c_{\mathrm{local}}, c_{\mathrm{global}}) = (0, 0.018, 0.045).$$

The penalty is charged in both evaluations. These constants are action preferences, not measured construction interactions. In expectation $V_a = 1.9\mathbb{E}[U_a]$; the repeated evaluation changes sampling noise but does not represent an adaptation transition or delayed forgetting. There is no Bellman bootstrap, learned state transition, or evolving adapter library in this diagnostic.

Router features include full-state task outcomes and actuation residuals computed by calling the simulator's known action transformation with its shift parameters. Candidate gradients also pass through that transformation and the dynamics. No shift-name label or final candidate outcome enters the router, but this is still simulator-provided information; estimating equivalent residuals and gradients from robot observations remains untested.

The threshold veto requires nominal success at least $\max(0.85, s_{\mathrm{keep}}^{\mathrm{nom}} - 0.05)$ and an improvement in success, distance, or proxy. It lacks confidence bounds and explicit task-failure, action-limit, and KL checks. Only learned/heuristic proposals are vetoed; fixed-scope references and the post-hoc oracle are unmasked. Accordingly, these comparisons do not instantiate the certified-set theorem. Their oracle maximizes noisy finite-evaluation values, not known true values.

Table 7: Exploratory sensitivity analysis from the same saved contact observations. Each difference first averages four shifts within a policy seed ($n = 5$). Paired t intervals use four degrees of freedom; the two-sided sign test enumerates all 2^5 sign assignments. Their assumptions differ, and no multiplicity adjustment is applied.

Comparator	Endpoint	Δ	95% CI	Sign p
Heuristic	Soft burden	-0.1890	$[-0.4051, 0.0270]$	0.1250
Heuristic	Utility	0.0048	$[-0.0090, 0.0187]$	0.3750
Fixed global	Soft burden	0.1324	$[0.0357, 0.2291]$	0.1250
Fixed global	Utility	-0.0083	$[-0.0214, 0.0047]$	0.1250
Fixed local	Soft burden	-0.3302	$[-0.5494, -0.1110]$	0.0625
Fixed local	Utility	0.0120	$[-0.0177, 0.0418]$	0.3125

Table 7 compares the paired inference procedures. The heuristic-minus-LOOPADAPT percentile bootstrap interval $[0.054, 0.325]$ is reproduced from the saved data. The paired t interval $[-0.027, 0.405]$ and sign-randomization $p = 0.125$ show that a robust significance claim is not supported by these five policy seeds. This does not establish equality or absence of an effect. Intervals condition on the single frozen progress/router fit and fixed shift suite, omitting uncertainty from training those models or sampling new families.

F.4 PLANAR CONTACT WITH FAMILY-LABEL ROUTING

We retain an earlier family-label candidate-path diagnostic in the same simulator. It uses a movable object, two soft finger contacts, inertia, damping, friction/slip, and a shelf lip. The policy controls gripper acceleration and aperture. A behaviorally cloned policy is nominally qualified before adaptation; a gated two-unit gripper residual defines the local candidate and a frozen-backbone affine output adapter defines the global candidate. The four held-out families are benign, local grip degradation, diffuse planar actuation corruption, and their compound. This is deliberately a lightweight differentiable simulator: it is not visual control, Isaac Sim, or hardware evidence.

Development, qualification, routing evidence, shadow validation, and final evaluation use disjoint seed sets. All five base-policy seeds pass a predeclared $\geq 90\%$ nominal qualification threshold before the final protocol is run. Per base policy and method, the executable record contains 1,600 candidate-construction or matched-frozen rollouts, 800 protected-nominal rollouts, 192 routing-evidence rollouts, 192 shifted plus 192 nominal shadow rollouts, and 384 shifted plus 384 nominal fresh final rollouts. LOOPADAPT independently constructs only the candidate proposed before candidate outcomes are observed; keep executes a frozen-policy matched spend. The auditor confirms all $5 \times 4 \times 4 = 80$ final cells, all 80 construction records, finite metrics, identical interaction counts, and inclusion of failures.

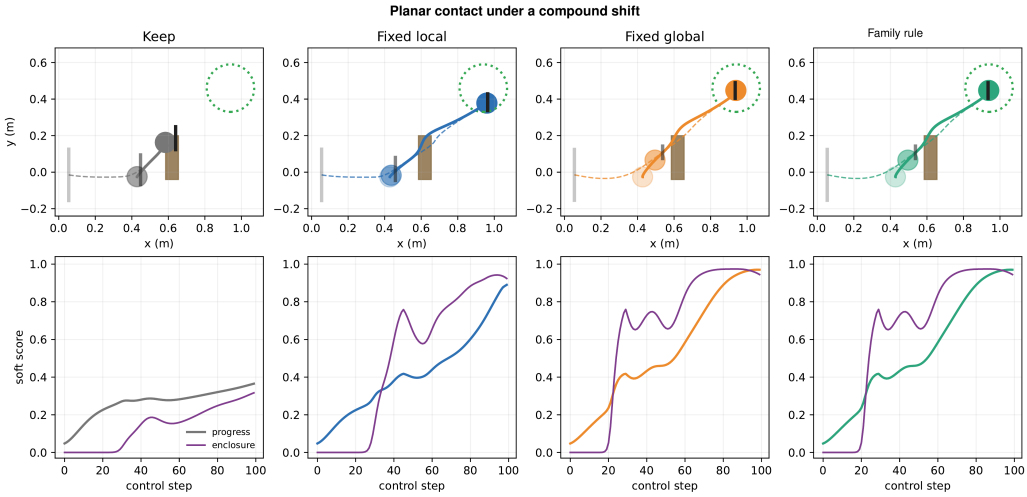


Figure 11: **Planar soft-contact simulation frames.** Representative held-out compound-shift trajectories show the object path (solid), gripper path (dashed), shelf lip, and goal region; the lower panels show progress and enclosure. The Family rule column is the predeclared heuristic.

Figure 11 shows the compound-shift rollout used in the main qualitative comparison. The unadapted object remains near the shelf lip; adapted trajectories advance toward the goal as progress and enclosure change.

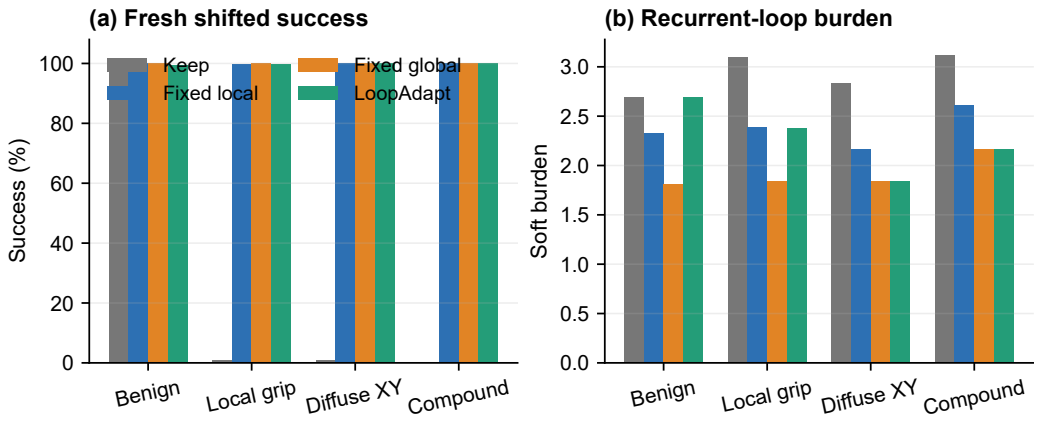


Figure 12: Success and soft stagnation/reversal proxy in the family-label contact experiment; five policy seeds per condition.

Figure 12 compares the family-label rule with fixed scopes. Successful methods can have different proxy values, so terminal success alone does not determine the utility ranking.

Table 8: Planar soft-contact final results (mean ± standard error over five base-policy seeds).

Shift	Method	Shifted (%)	Nominal (%)	Loop burden
benign	keep	99.4 ± 0.4	99.4 ± 0.4	2.686 ± 0.025
benign	fixed local	100.0 ± 0.0	100.0 ± 0.0	2.322 ± 0.075
benign	fixed global	100.0 ± 0.0	100.0 ± 0.0	1.806 ± 0.081
benign	LoopAdapt	99.4 ± 0.4	99.4 ± 0.4	2.686 ± 0.025
local grip	keep	0.7 ± 0.7	99.4 ± 0.4	3.098 ± 0.017
local grip	fixed local	99.7 ± 0.3	100.0 ± 0.0	2.388 ± 0.029
local grip	fixed global	100.0 ± 0.0	100.0 ± 0.0	1.836 ± 0.087
local grip	LoopAdapt	99.7 ± 0.3	100.0 ± 0.0	2.379 ± 0.026
diffuse xy	keep	0.7 ± 0.5	99.4 ± 0.4	2.832 ± 0.033
diffuse xy	fixed local	100.0 ± 0.0	100.0 ± 0.0	2.164 ± 0.067
diffuse xy	fixed global	100.0 ± 0.0	100.0 ± 0.0	1.844 ± 0.089
diffuse xy	LoopAdapt	100.0 ± 0.0	100.0 ± 0.0	1.844 ± 0.089
compound	keep	0.0 ± 0.0	99.4 ± 0.4	3.121 ± 0.005
compound	fixed local	99.9 ± 0.1	100.0 ± 0.0	2.605 ± 0.019
compound	fixed global	100.0 ± 0.0	100.0 ± 0.0	2.166 ± 0.070
compound	LoopAdapt	100.0 ± 0.0	100.0 ± 0.0	2.166 ± 0.070

Table 8 reports outcomes by shift family. The predeclared family-label scope rule commits keep on all benign seeds, the local candidate on all local-grip seeds, and the global candidate on all diffuse and compound seeds. On the three non-benign families, this gives 99.7%, 100%, and 100% fresh shifted success with 100% protected nominal success; keep reaches 0.73%, 0.73%, and 0%. No proposed non-keep update is rejected in this final run. The rule reads the predefined shift family and is not an output of the learned Q-router. Against a post-hoc utility oracle evaluated on the finite rollout set over the three constructed actions, agreement is 45%, mean regret is 0.0101, and maximum regret is 0.0448: successful candidates are often near-tied, and the burden/cost terms can favor a different scope. This experiment tests scoped construction, nominal retention, the threshold veto, and commitment. It does not establish router superiority, autonomous shift identification, local-scope necessity, interaction savings, or robot transfer.

The following experiments examine scalar burden estimation, metric repairability, and synthetic routing. They use the aggregate formulation in Appendix C; the common-descent study is reported separately in Appendix E.1.

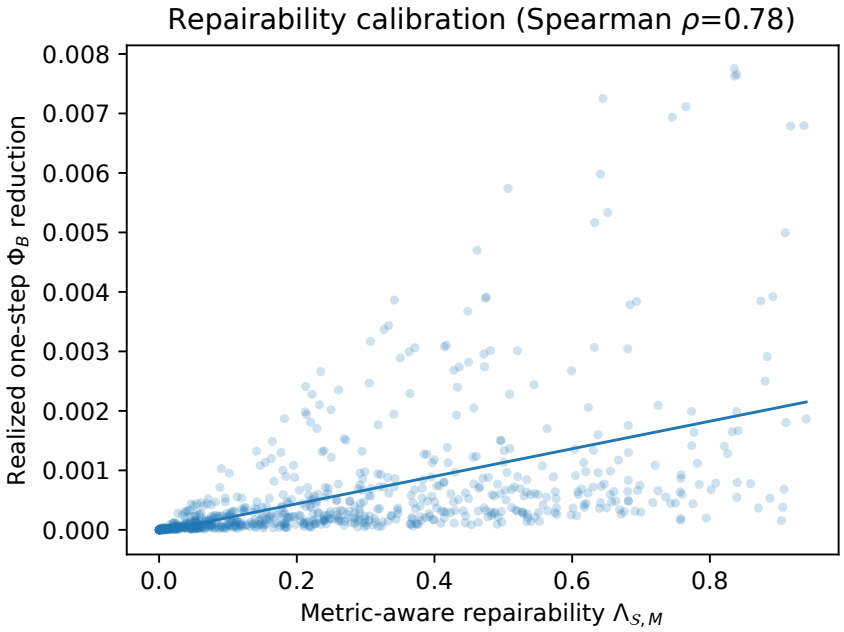


Figure 13: Metric-aware repairability versus realized one-step burden reduction over random restricted subspaces. The main controlled setting gives Spearman $\rho = 0.784$ versus 0.699 for Euclidean repairability.

Figure 13 compares the scalar metric-aware score with realized one-step descent. Their association supports its use as a local diagnostic, while the scatter indicates that the score does not fully predict finite-step outcomes.

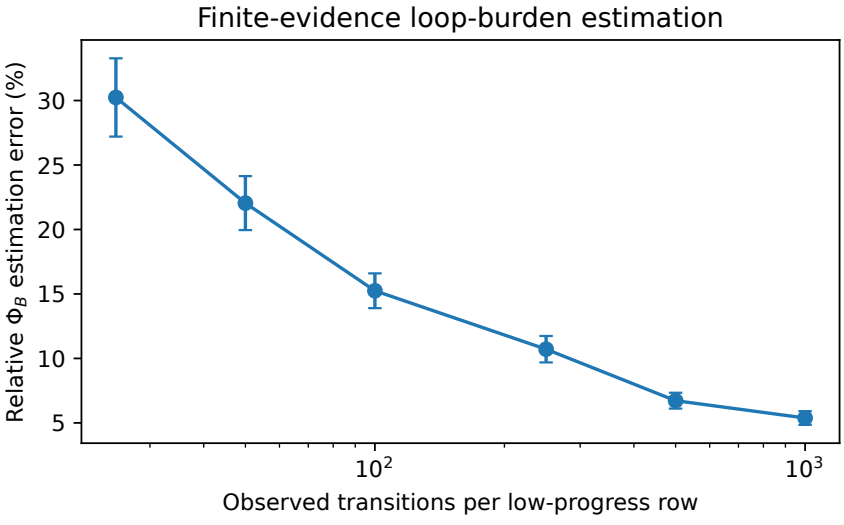


Figure 14: Finite-evidence burden estimation. Error decreases with observed transitions per low-progress row but remains non-negligible at small sample sizes, motivating abstention and kernel-conditioning diagnostics.

Figure 14 shows decreasing estimation error as more transitions are observed per row. The remaining error motivates accounting for uncertainty before selecting an update.

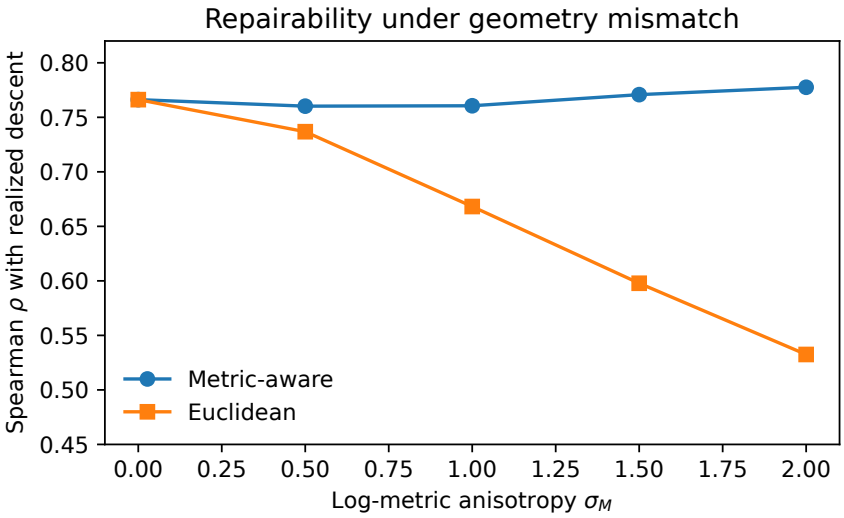


Figure 15: Repairability sensitivity to update-geometry anisotropy. Metric-aware and Euclidean scores coincide at $\sigma_M = 0$; the Euclidean calibration deteriorates as the trust-region geometry becomes more anisotropic.

Figure 15 shows why candidate scopes need a common metric: Euclidean repairability becomes less aligned with realized descent as the update geometry becomes anisotropic.

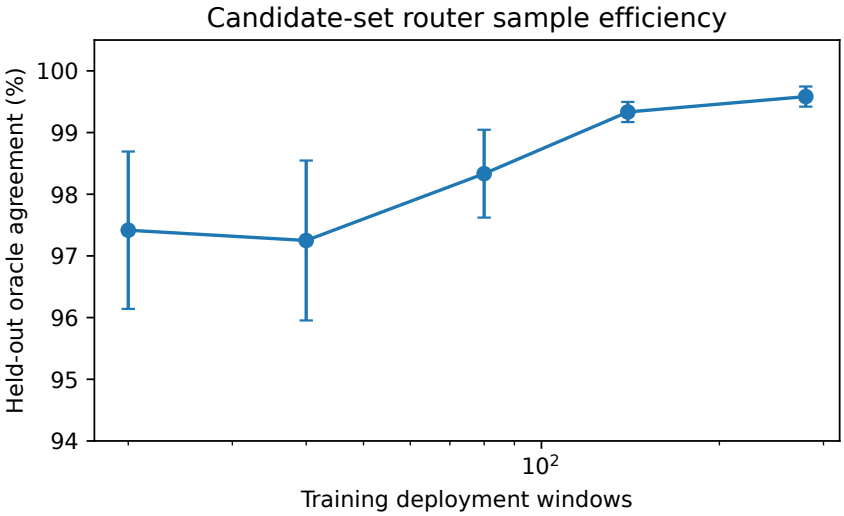


Figure 16: Synthetic routing accuracy versus training-window count under a fixed generator.

Figure 16 measures the data requirement of the synthetic routing task. More training windows improve oracle agreement under this generator; robot interaction requirements are not measured by this curve.

Table 9: Controlled OOD routing regret (lower is better), averaged over five independently trained routers. None of these OOD families appears in the training generator.

OOD family	Learned LH	Rule	Myopic	Immediate
Mixed local+diffuse	0.0095	0.0725	0.0694	0.0782
Severity extrapolation	0.0022	0.1233	0.3079	0.2838
Recurrence extrapolation	0.0012	0.0790	0.3118	0.2344

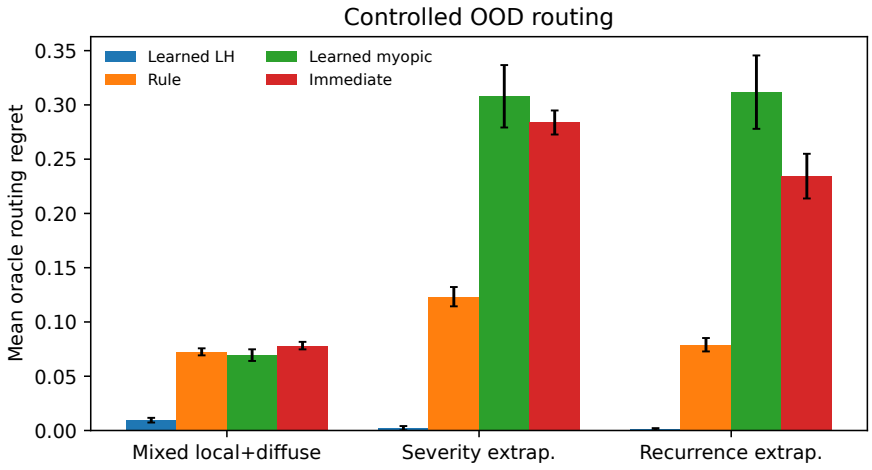


Figure 17: Controlled OOD routing regret over mixed local+diffuse, severity-extrapolation, and recurrence-extrapolation families absent from the training generator. The mixed family is the hardest and exposes the largest remaining generalization gap.

Figure 17 and Table 9 test held-out generator families. Mixed local and diffuse shifts produce the largest regret, indicating that combining familiar shift types is harder than the other extrapolations considered.

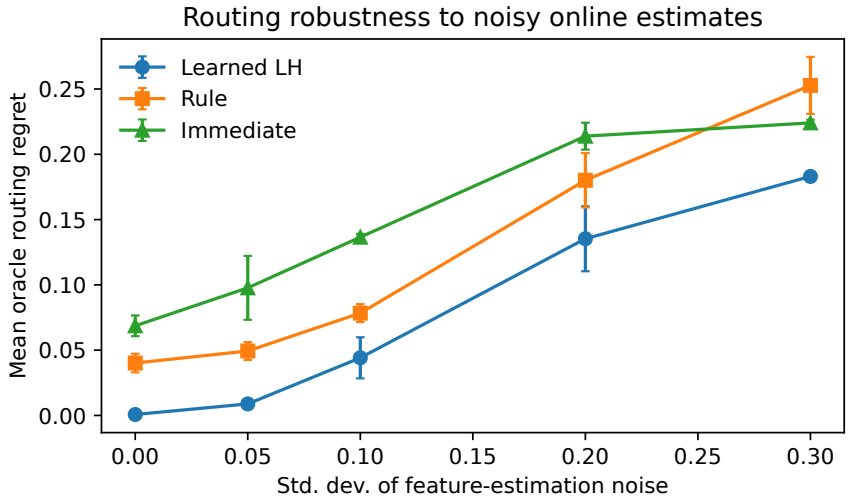


Figure 18: Sensitivity to noisy online features. Noise is added to structural state and continuous candidate features while oracle action values remain fixed. All selectors degrade; the learned long-horizon router retains lower regret than the rule across the tested range.

Table 10: Repeated-inspection certification with an anytime Hoeffding lower confidence sequence for bounded utility increments. At zero true gain, acceptance is a false positive; for positive gain, the final column reports median stopping time among accepted streams.

True mean	Ever accepted (%)	Median t
0.00	0.00	–
0.10	3.65	1206
0.20	97.15	751
0.30	100.00	294
0.40	100.00	154

Figure 18 perturbs input features while keeping oracle values fixed. Regret rises with noise for all methods, although the long-horizon router retains lower regret than the rule over this range.

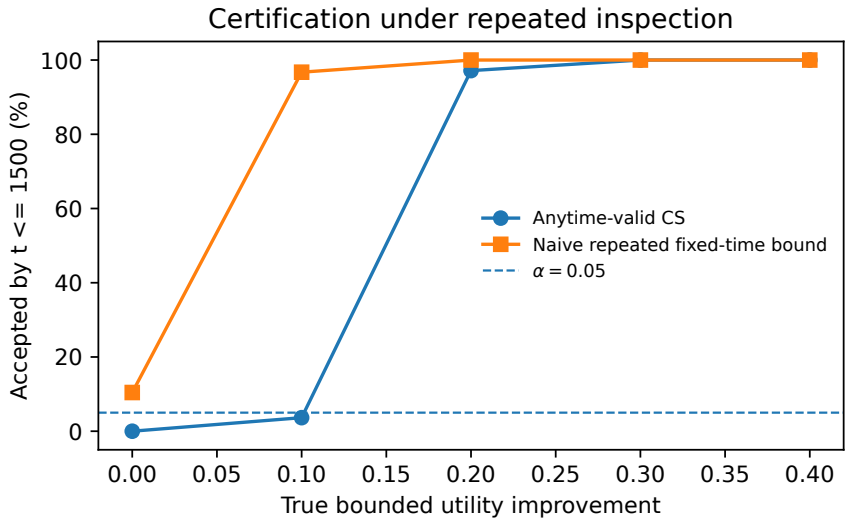


Figure 19: Repeated-inspection certification. The anytime-valid reference bound controls false acceptance in the null simulation but is substantially more conservative than repeatedly reusing a fixed-time bound; the latter is faster but invalid under arbitrary peeking.

Figure 19 and Table 10 quantify the acceptance-delay cost of repeated-inspection validity. Small positive gains are seldom accepted within the finite horizon, while the null simulation records no false acceptance for the time-uniform bound.

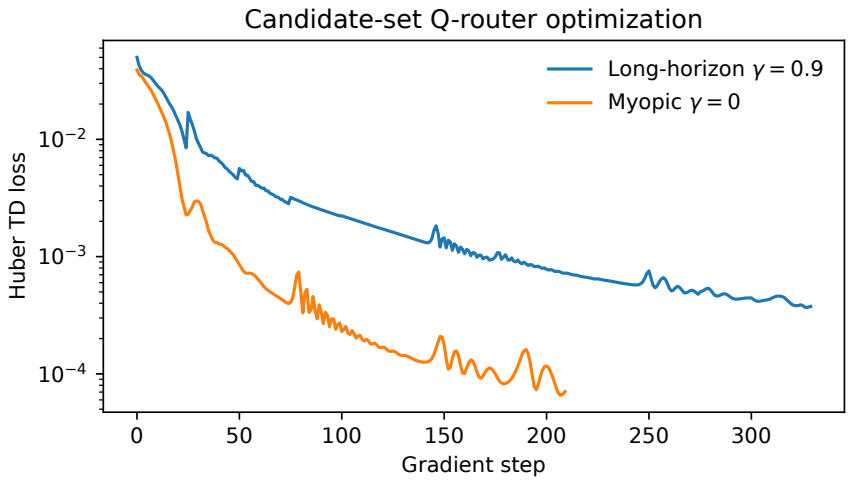


Figure 20: Representative fitted-Q optimization curves. Training loss is a diagnostic only; held-out oracle regret is the routing endpoint.

Figure 20 reports fitted-Q optimization for the long-horizon and myopic models. These losses diagnose training; held-out regret provides the comparison of decision quality.

F.5 Variable candidate sets and certified-set ranking

The main controlled router sees three actions. To test the variable-set interface separately, five routers trained on limited data (40 root windows, 50 fitted-Q steps) are evaluated after adding valid but dominated local proposals. We use a smaller training set than in the main routing experiment to measure sensitivity to candidate-set size.

Table 11: Variable-candidate and certification-mask stress test. Five deliberately data-starved routers are trained on 40 root windows for 50 FQI steps; results are mean ±95% CI across seeds. This is a mechanism stress test, not the main routing result.

Extra local decoys	Agreement (%)	Regret	Decoy selected (%)
0	96.2±2.3	0.0105±0.0087	0.0±0.0
2	95.5±2.6	0.0126±0.0099	0.7±0.6
4	95.2±3.2	0.0132±0.0112	1.0±1.3
8	94.3±4.1	0.0157±0.0149	1.8±2.2
16	93.0±5.5	0.0186±0.0176	3.2±3.6
Random certification masks: restricted-set oracle agreement 98.2±1.2%, regret 0.0045±0.0036.			

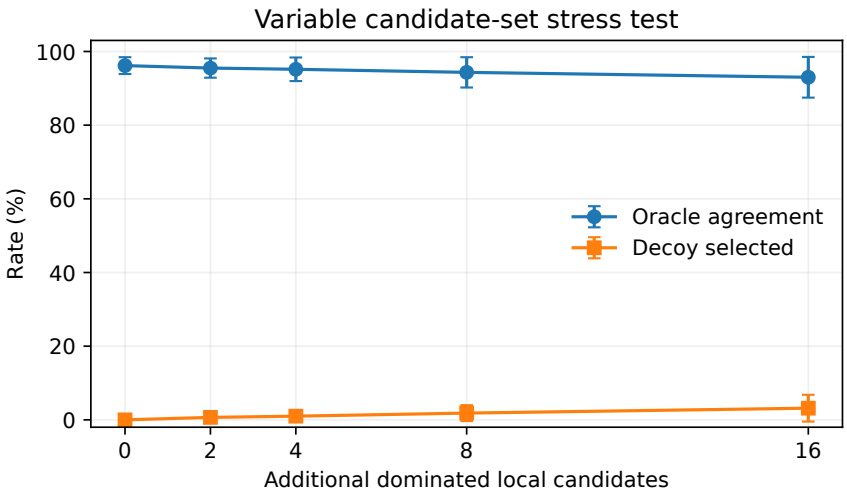


Figure 21: Oracle agreement and dominated-candidate selection as the candidate set grows.

Figure 21 and Table 11 show a gradual decline in oracle agreement as dominated proposals are added. The nonzero selection rate for these proposals identifies errors that padding invariance alone cannot prevent.

F.6 Paired statistics and conditioning diagnostic

Table 12: Paired robustness statistics for executed studies. Intervals are percentile bootstrap 95% CIs over the paired experimental unit; p is a one-sided Wilcoxon signed-rank test in the stated favorable direction. The five-seed PointMass test is included only as a small reference check.

Comparison	n	Mean difference	95% CI	p
Local Φ_B descent − unaware	96	0.0664	[0.0571, 0.0763]	$< 10^{-4}$
Rule regret − learned LH regret	10	0.0423	[0.0380, 0.0471]	0.00098
Myopic regret − learned LH regret	10	0.0861	[0.0814, 0.0915]	0.00098
PointMass no-Φ_B − with-Φ_B	5	0.0389	[0.0353, 0.0422]	0.031

Table 12 reports paired comparisons for kernel control, synthetic routing, and PointMass. Paired inference uses the experimental unit that was independently randomized: kernel instance for burden descent, router-training seed for routing, and pretrained-policy seed for PointMass. We do not treat the hundreds of held-out windows inside one trained router as independent seeds.

The perturbation theory highlights $\|N\|$ as a sensitivity factor, but it need not be a sufficient empirical error predictor because support allocation and edge probabilities also matter. At 100 transitions per row over 128 exact kernels, the observed Spearman correlation between $\log \|N\|_2$ and relative Φ^{B} estimation error is only 0.164; condition-number-only abstention gives little error reduction. This result motivates checking support and confidence width together with conditioning.

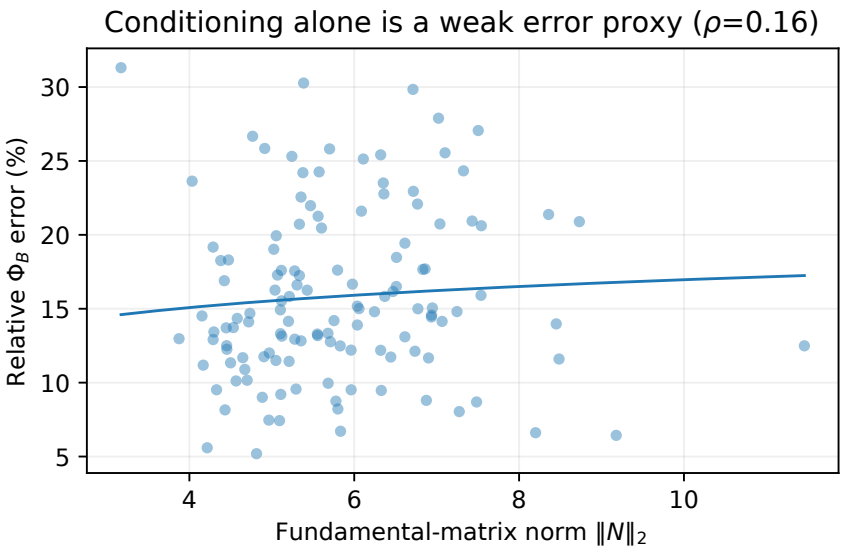


Figure 22: Fundamental-matrix norm and relative burden-estimation error across 128 kernels.

Figure 22 shows a weak relation between matrix norm and estimation error. Conditioning therefore needs to be considered alongside edge support and sampling uncertainty.

G Routing Ablations

We vary the routing horizon, input features, confidence level, and candidate-set size to examine the source of the routing behavior and its computational cost. Each experiment uses the controlled adaptation generator and its stated replication unit.

G.1 Intervening on the routing horizon

The main routing result compares a long-horizon learner with a separately trained myopic model. To isolate temporal horizon from architecture and dataset construction, we additionally hold the replay transitions, network architecture, optimizer, and test oracle fixed while varying only the Bellman discount used during fitted-Q learning. The evaluation oracle remains $\gamma_R^\star = 0.9$.

Table 13: Training-discount intervention: mean regret ± approximate normal 95% interval half-width across three router fits, evaluated at oracle discount 0.9.

γ_R	Overall regret ↓	Recurring regret ↓
0.00	0.0585±0.0490	0.1576±0.1061
0.25	0.0321±0.0343	0.0900±0.0817
0.50	0.0216±0.0231	0.0463±0.0627
0.75	0.0175±0.0143	0.0251±0.0394
0.90	0.0158±0.0132	0.0144±0.0282
0.98	0.0175±0.0136	0.0144±0.0282

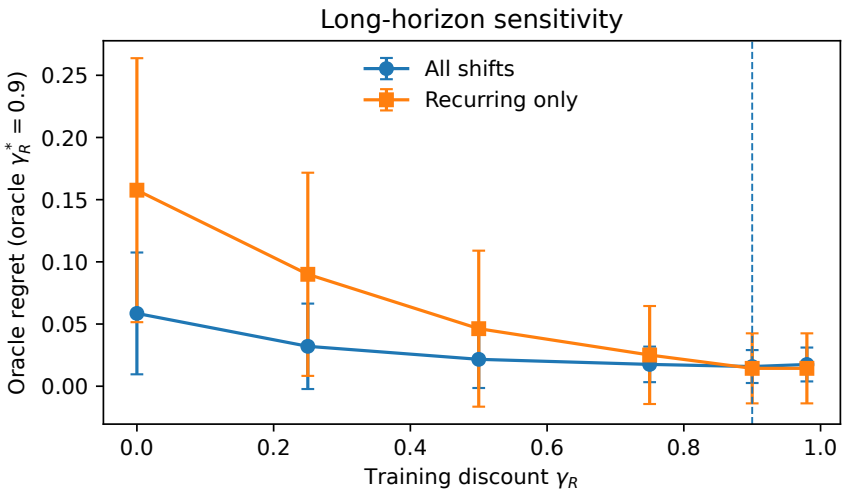


Figure 23: Causal horizon intervention in the controlled routing generator. Increasing the training discount sharply reduces recurring-condition regret until the objective-matched range near $\gamma_R = 0.9$; pushing to 0.98 provides no further gain. Error bars are 95% CIs across three independently trained low-data routers.

Figure 23 and Table 13 vary the training discount with data and evaluation fixed. Regret decreases toward the evaluation discount 0.9; increasing it to 0.98 brings no further mean improvement.

Across ten independently generated test streams (300 windows each), the immediate-gain action differs from the long-horizon oracle on $16.1 \pm 1.3\%$ of windows. Disagreement occurs in $37.2 \pm 3.5\%$ of localized shifts and $26.7 \pm 3.4\%$ of recurring shifts, compared with $0.4 \pm 0.6\%$ of diffuse shifts and no benign-novelty cases. Of 482 pooled disagreement windows, 479 are global→local reversals from immediate to long-horizon choice. These pooled windows describe the routing differences; the ten independent streams remain the units of statistical analysis.

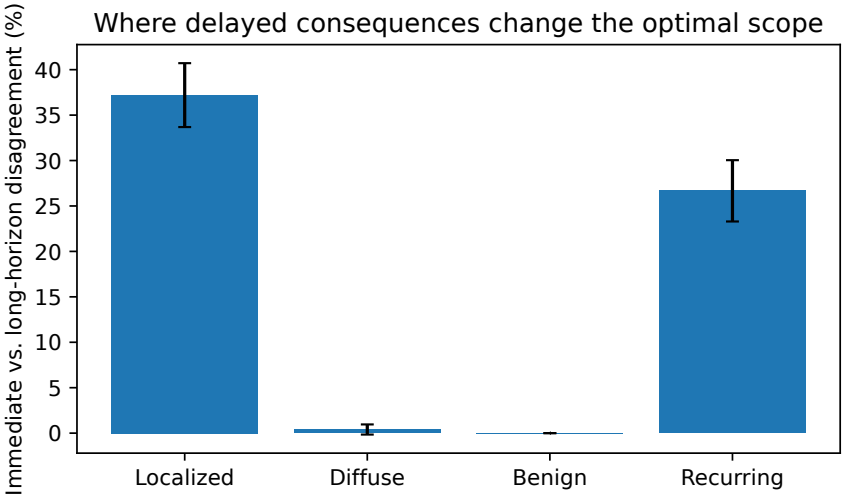


Figure 24: Where delayed consequences change the optimal scope. The immediate and long-horizon decisions almost always agree on diffuse and benign cases; the gap is concentrated in localized and recurring conditions where retention debt or adapter reuse changes future value.

Figure 24 locates the disagreement between immediate and long-horizon decisions. Localized and recurring conditions account for nearly all changes in the preferred update scope.

G.2 Router feature ablations

We train three routers with 40 root windows per seed and 40 fitted-Q steps. Removing the burden scalar alone slightly reduces mean regret, whereas removing retention or the combined structural features (burden, repairability, recurrence) increases it. The broad confidence intervals limit comparisons among individual features. These results indicate redundancy among router inputs; the effect of Φ^{B} on candidate learning is evaluated separately through the objective ablation.

Table 14: Low-data router feature intervention. The broad CIs reflect only three independently trained routers and should be read as a diagnostic of redundancy, not a ranking of feature importance.

Variant	Agreement (%) ↑	Regret ↓
Full	95.00±4.42	0.0158±0.0132
w/o burden	97.50±1.22	0.0078±0.0038
w/o repairability	95.83±6.34	0.0142±0.0222
w/o retention	91.04±3.56	0.0352±0.0085
w/o recurrence	95.42±4.32	0.0152±0.0140
w/o cost	95.00±3.08	0.0178±0.0081
w/o structural trio	88.33±14.57	0.0537±0.0731

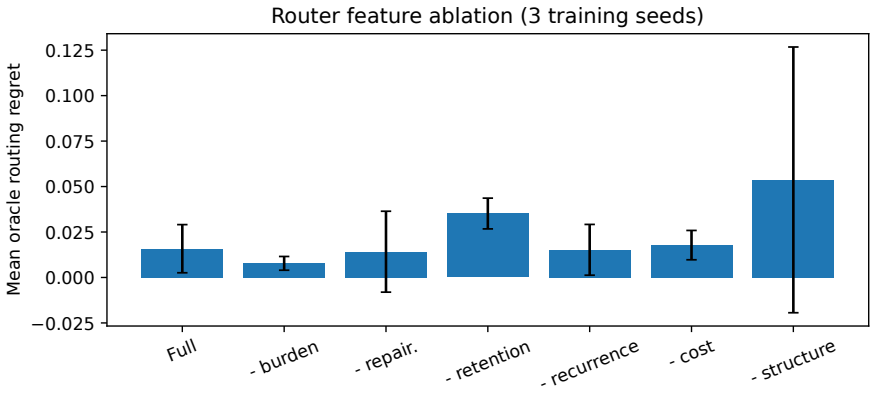


Figure 25: Secondary router feature ablation. Removing all three structural cues or the retention-risk feature is damaging in this low-data regime; burden alone is redundant with the remaining correlated features.

Figure 25 and Table 14 show correlated information among router inputs. Removing burden alone has little effect, while removing retention or the combined structural features increases mean regret.

G.3 Confidence level and inference time

The main shield experiment fixes $\alpha = 0.05$. A confidence-level sweep over $\alpha \in \{0.10, 0.05, 0.02, 0.01\}$, with 2,500 bounded streams per effect and 1,500 possible inspection times, observes zero null false acceptances at every tested level. For true normalized gain 0.20, the median acceptance time increases from 724 samples at $\alpha = 0.10$ to 841 at 0.01; the corresponding acceptance probability by the horizon decreases from 98.36% to 96.0%. This sweep measures the dependence of acceptance delay on the allocated statistical error.

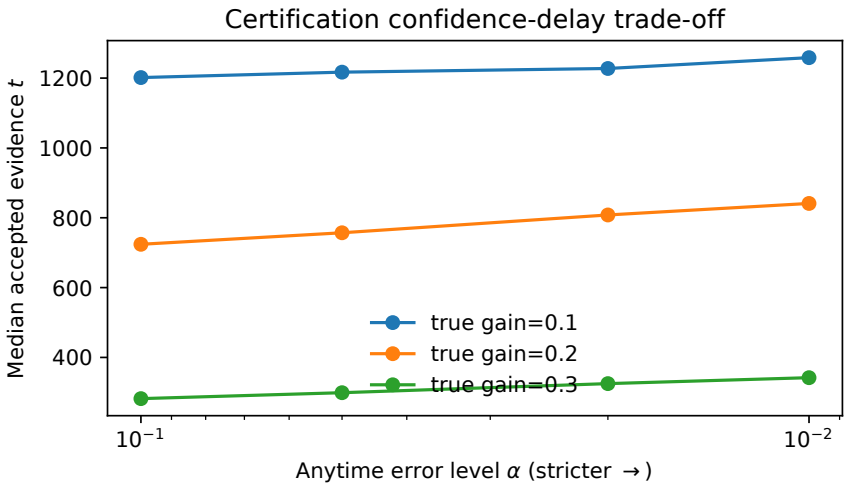


Figure 26: Confidence–delay trade-off for the anytime-valid reference shield. Stricter error levels require more evidence, especially for small positive effects.

Figure 26 shows the additional observations required by smaller error budgets. Acceptance times depend on both effect size and confidence level.

The reference candidate-set router contains 21,697 trainable parameters. On the recorded CPU environment, median single-observation forward latency rises from roughly 0.41 ms for 3 valid candidates to 0.59 ms for 128. These numbers are hardware-specific and do not include candidate construction or validation; without a matching end-to-end timing measurement, they do not determine the fraction of total cost spent on routing.

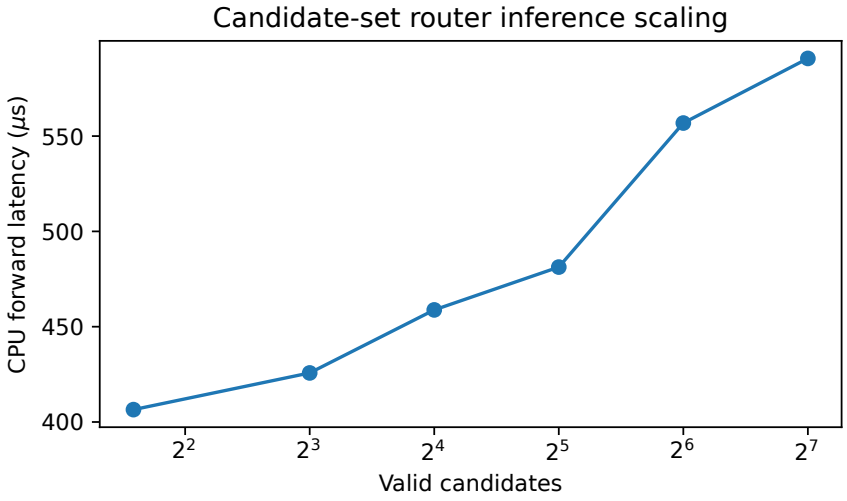


Figure 27: Hardware-specific CPU forward latency as the number of valid candidates grows. Candidate generation and environment interaction are intentionally excluded.

Figure 27 measures only candidate-set forward evaluation on the recorded CPU. Its moderate growth with set size does not include the cost of producing or validating candidates.

H ADAPTER ABLATIONS

We examine adapter rank, objective weight, reuse, and routing uncertainty. The PointMass sweeps use three pretrained-policy seeds and a 20-step, batch-16 local-optimization budget so that rank and burden-weight interventions share one fixed low-compute schedule. They use three policies, separately from the five-policy main comparison.

H.1 ADAPTER RANK

Table 15: Low-compute localized-shift rank sweep. Success saturates at rank 2, while recurrent burden continues to improve with additional local capacity.

Rank	Parameters	Success (%)	$\Phi_B \downarrow$
1	8	0.0	0.7241 ± 0.0825
2	16	100.0	0.2931 ± 0.0225
4	32	100.0	0.1999 ± 0.0091
8	64	100.0	0.1268 ± 0.0052
16	128	100.0	0.0719 ± 0.0020

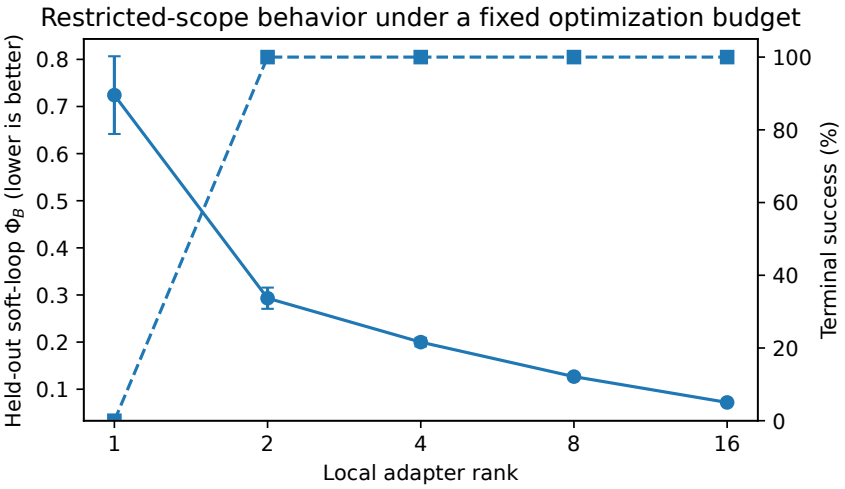


Figure 28: Restricted-scope capacity after terminal success saturation. Under the fixed 20-step budget, rank 1 does not recover while ranks 2–16 do, and Φ^{B} falls from 0.2931 ± 0.0225 at rank 2 to 0.0719 ± 0.0020 at rank 16.

Figure 28 shows that ranks 2–16 reach the same terminal success while retaining different loop potentials. The structural endpoint distinguishes their behavior after binary success saturates.

Table 15 reports the corresponding parameter counts and outcomes. The trainable residual contains only $8r$ parameters in this reference. Rank-1 failure under the matched budget is a candidate-generation outcome, not a routing error, but it is not proof of representational impossibility under longer optimization. Among scopes that do recover, binary task success no longer distinguishes them while the structural objective still resolves recurrent inefficiency. This supports exposing scope capacity and repairability to candidate construction and routing.

H.2 SENSITIVITY TO THE BURDEN WEIGHT

With rank fixed to 8, all tested $\lambda_B \in \{0, 0.5, 1.5, 3, 6\}$ reach 100% terminal success under the same reduced optimization schedule. Mean soft-loop burden decreases from 0.1534 ± 0.0031 at $\lambda_B = 0$ to 0.1100 ± 0.0120 at $\lambda_B = 6$. The observed stability applies to this tested weight range and PointMass setting.

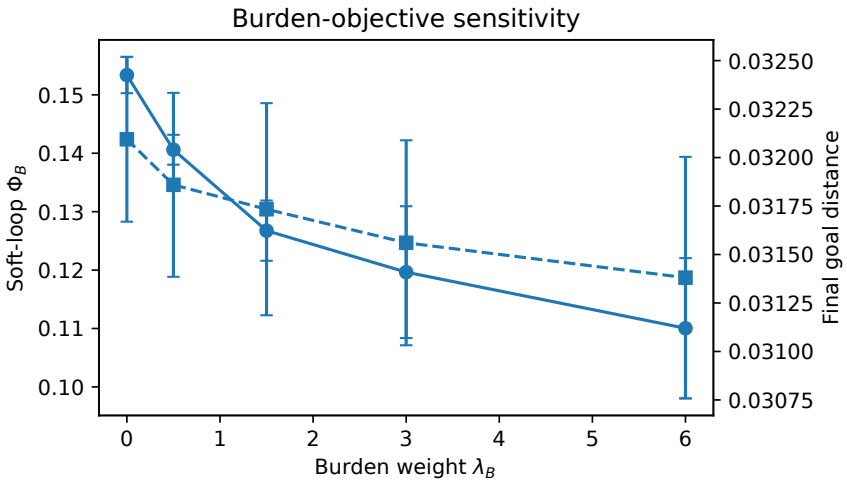


Figure 29: Burden-weight sensitivity. The recurrent objective improves Φ^{B} throughout the tested range without changing terminal success in this low-compute PointMass diagnostic.

Figure 29 varies the burden weight at fixed rank and optimization budget. The tested weights reduce loop potential without changing success, with variation across the three policies.

H.3 Adapter reuse under changing severity

A rank-8 adapter is trained once at localized severity 1.0 and later evaluated with no additional gradient update. Table 16 reports the result. The stored adapter succeeds at severities 0.6, 0.8, and 1.0, including the training-severity recurrence where the base checkpoint fails. It fails at 1.2 and 1.4. Hence persistent adapter memory can create future value, but retrieval is not itself a deployment decision: a recalled adapter remains a candidate that must be re-ranked and certified under current evidence.

Table 16: Zero-additional-optimization reuse of a local adapter trained at severity 1.0.

Severity	Keep success (%)	Reuse success (%)	Reuse Φ_B
0.6	100.0	100.0	0.0383
0.8	100.0	100.0	0.0621
1.0	0.0	100.0	0.1268
1.2	0.0	0.0	1.0047
1.4	0.0	0.0	0.9193

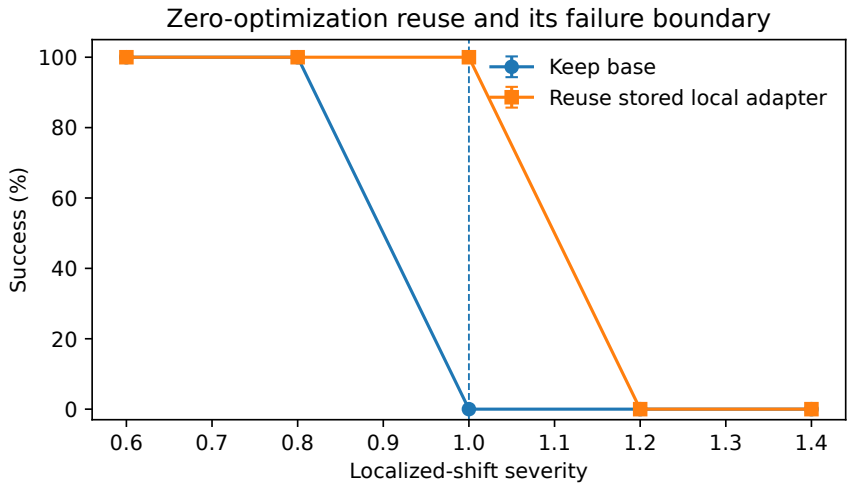


Figure 30: Stored-adapter success across localized-shift severity, with no additional optimization.

Figure 30 shows successful reuse at severities up to the training value and failure at the two larger shifts. Stored adapters therefore need validation under the current condition.

H.4 Value margins and routing errors

We train three routers on limited data using 80 root windows and 40 fitted-Q steps and compare predicted candidate values against the known controlled adaptation values. Candidate-value Spearman agreement is 0.877 ± 0.052, action agreement is $92.4 \pm 5.1\%$, and mean regret is 0.0242 ± 0.0176. Routing errors are concentrated in small top-two predicted-Q margins: the lowest-margin quintile contains $78.1 \pm 21.5\%$ of all errors, has $72.8 \pm 14.6\%$ action agreement and 0.0875 ± 0.0544 regret, whereas the highest-margin quintile has no observed error in these three runs.

This diagnostic is intentionally not used to replace the shield. A confident but misspecified router can still be wrong, and safety/retention constraints require candidate-specific evidence with repeated-inspection validity.