RESPONSE-CONDITIONED OPTIMIZER ROUTING ON ONE IRREVERSIBLE TRAJECTORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Many stochastic optimizers are available for training a model, and it is rarely clear in advance which one to use. Worst-case convergence bounds can be a pessimistic guide; A/B tests need one training run per candidate, which, for a single large model, can cost more than a good choice saves; tests on smaller models can be misleading. We therefore study how to choose the optimizer while the model itself is being trained. Such a run has only one model state: updates cannot be undone, and optimizers that were not chosen cannot be tried from the same state. At each step, a stochastic gradient is revealed first, and a router then chooses which candidate algorithm uses it to produce a new testing point. We compare the expected terminal loss with that of the best fixed optimizer run from the same initial state with the same budget of B gradients and updates, without resets, model copies, or extra queries.

We propose three methods that share one model of the objective, fitted during a persistently exciting calibration phase that explores all decision-relevant directions. PE-Commit chooses before the first post-calibration gradient, PE-Post chooses once after seeing it, and PE-RCR may switch again at later decision points. For unknown strongly convex quadratics and optimizers with contractive affine-linear state updates, we prove explicit finite-sample bounds on the positive part of the expected pseudo-regret of all three methods. The bounds hold in expectation and assume bounded, conditionally mean-zero noise with a fixed covariance, which is stronger than generic minibatch noise; PE-RCR pays an estimated term at every decision point. An exact two-step example shows that scoring by the remaining horizon can be the unique optimal gradient-contingent plan and can strictly beat one-step greedy selection that also sees the gradient. A lower bound shows that no router can identify the better next method uniformly without information along decision-relevant directions.

1 INTRODUCTION

Many stochastic optimization methods are available for training machine learning models: SGD (Robbins & Monro, 1951), momentum methods (Polyak, 1964; Nesterov, 1983), Adam and AdamW (Kingma & Ba, 2015; Loshchilov & Hutter, 2019), and numerous later variants. Each method has its own strengths and weaknesses, the tasks on which they work well overlap, and no method clearly dominates on all tasks (Schmidt et al., 2021). The natural question is which one to choose.

One way is to compare the convergence guarantees of the candidates and to choose the method with the best upper bound; however, such bounds describe the worst case over a class of problems and can be pessimistic for the problem at hand. Another standard way is an A/B test, which trains the model once with each candidate and keeps the best. If the goal is to train one large model, this selection costs more computation than the training itself, and hence more than a good choice can save. Tests on smaller models are cheaper, but they can mislead: the advantage of an optimizer on small models can shrink as the model grows (Wen et al., 2026), and the outcome of a comparison can depend on the random seed (Bouthillier et al., 2021). This leads to our main question: *how can we choose the optimizer while training the actual model?*

The difficulty is that a training run has only one model state. At step q, a stochastic oracle returns a gradient at the current query point, a *router* chooses which method uses it, and the resulting state is

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 only starting point for all later decisions; a method that was not chosen cannot be resumed from a parallel checkpoint. Rules that switch optimizers during a run exist (Keskar & Socher, 2017; Zou et al., 2026; Chen et al., 2026), but to our knowledge none comes with a guarantee relative to the best fixed optimizer under the same budget (Section 2).

The problem is neither classical algorithm selection nor a stationary bandit. All methods minimize the same f, so their standalone minima coincide. The quality of a method is instead the value of continuing with it from the current state (iterate and optimizer memory) for the remaining budget, and trying one method changes every future counterfactual. We therefore compare the terminal loss with that of the best fixed optimizer that uses the same budget from the same initial state.

Because the gradient is revealed before the choice, three information sets arise: a router may commit before seeing the next gradient, commit once after seeing it, or also decide again after later gradients. Choosing after seeing a gradient couples the chosen method with the noise of its update, so the value formula of a precommitted method cannot simply be reused, and drawing a fresh gradient for the decision would change the budget. We compare all three options with one model fit and one budget.

Contributions. Our main contributions are as follows.

1. **Budget-matched formulation.** We compare the router with the best fixed optimizer under the same budget of B gradients and B updates from the same initial state (Eqs. (1)–(2)). Appendix Propositions 4 and 5 show that neither information along decision-relevant directions nor accounting for the full budget, including calibration, can be dropped.
2. **Three learned routers with guarantees.** Our methods share a calibration phase of m predictable, *persistently exciting* (PE) steps, which explore all decision-relevant directions, and one model fit. PE–Commit then chooses before seeing the next gradient g_m, PE–Post chooses once after seeing it, and PE–RCR (receding-commit routing, Algorithm 1) makes $O(\log B)$ decisions, each after seeing the current gradient. Theorem 1 bounds the positive part of the expected pseudo-regret of all three under the full budget, with rate $B^{-1/2}$ up to logarithms. The one-shot methods pay one estimation term and PE–RCR one per decision, so rerouting is not needed for consistency, and we do not claim that it always helps. Appendix Proposition 3, Corollary 4, and Section 4 treat random PE, unknown covariance, and computational cost.
3. **Later gradients can help.** On an exact, stable two-step quadratic, receding-horizon commit with unit blocks is the unique optimal gradient-contingent policy and beats an exact one-step greedy router, which also sees the gradient, by $1/135$ (Proposition 1 and Appendix Proposition 2). After one gradient, the rankings by immediate loss and by remaining-horizon value are strictly reversed, and the same policy beats every predictable policy and every complete schedule chosen after the first gradient. This separates scoring horizons, not receding routing from every reactive rule. Lean 4 verifies these finite separations and the deterministic assembly of the bounds from explicit hypotheses, not the complete stochastic theorem.
4. **Experiments.** On 64 new quadratic tasks with shared optimizer memory, drawn from a previously tuned and then frozen generator after the code and primary comparison were locked, PE–RCR beats PE–Post at all three budgets (Figure 1). On nonlinear problems, a schedule fixed in advance helps, but neither learned router beats a fixed method (Figure 2).

Scope. The guarantees cover strongly convex quadratics with contractive affine-linear optimizer states, not Adam's normalization, frozen private memories, or nonconvex objectives. The noise model fixes one bounded conditional covariance on all reachable histories, which is stronger than generic martingale-difference or minibatch noise with iterate-dependent covariance.

2 Related work

Algorithm selection and optimizer switching. Classical algorithm selection acts before a run; warm-started variants switch once after a prefix, and serial black-box portfolios keep independent runs of their constituents (Rice, 1976; Kerschke et al., 2019; Vermetten et al., 2020; Kostovska et al., 2022; Vermetten et al., 2023; Dinu et al., 2026). Dynamic algorithm configuration can encode repeated state-dependent optimizer choices, so switching within a run is not new in itself (Biedenkapp et al., 2020; Adriaensen et al., 2022); examples include switching by reinforcement learning or bandits,

Table 1: Closest formulations. "Before" and "after" indicate whether the decision precedes or follows the current oracle response, transition, or cost.

Work	Decision timing	State and access	Main guarantee or target
GAPS/BAPS (Lin et al., 2023)	before stage outcome	one contractive trajectory	cumulative policy regret
M-GAPS (Lin et al., 2024)	before stage outcome	one nonlinear trajectory; online model fit	nonconvex local regret
Online Switching Control (Li et al., 2023)	before plant transition	one plant; batched controllers	stability and cumulative policy regret
Greedy Alignment (Lee & Lee, 2025)	after current gradient	one iterate; shadow filter states	alignment; one-step descent bound
DOIT (Chen et al., 2026)	state-driven	optimizer-specific surrogates; reset on switch	empirical dynamic interleaving
This work	once before; once or repeatedly after	one shared state; no resets or copies	terminal loss vs. best fixed optimizer; strict value of later gradients

Adam-to-SGD transitions, guarded learned optimizers, hyper-heuristics with one shared incumbent, and a continuous interpolation of optimizers learned by hypergradients (Getzelman & Balaprakash, 2021; Meidani et al., 2022; Keskar & Socher, 2017; Zou et al., 2026; Prémont-Schwarz et al., 2022; Lehre & Özcan, 2013; Lissovoi et al., 2019; Antipov & Doerr, 2026; Ozkara et al., 2024).

Control and identification on one trajectory. On one contractive trajectory of length T, GAPS gives $O(\sqrt{T})$ convex adaptive-policy regret and a nonconvex local-regret bound, while finite-policy BAPS gives $O(T^{2/3})$ expected static policy regret (Lin et al., 2023). Online Switching Control uses batched Exp3-ISS on an unknown plant for stability and $\widetilde{O}(N^{1/3}T^{2/3}) + \exp(O(M))$ policy regret, where N is the number of candidate controllers and M the number of those that are not exponentially input-to-state stable (Li et al., 2023). The closest learned-control analogue, M-GAPS, couples online model estimation and policy optimization through the prediction error along the visited path (Lin et al., 2024). These methods act before observing outcomes and target cumulative or local regret; we choose which update rule consumes an observed gradient and compare the terminal value with a budget-matched fixed optimizer whose memory is part of the state. Work on memory, stateful master algorithms, and navigation lacks this structure (Zhao et al., 2023; Agarwal et al., 2017; Cutkosky et al., 2021; Al Marjani et al., 2021); meta-optimization transfers information across problems, not within one run (Chen & Hazan, 2023). Adaptive LQR also combines estimation along the trajectory with perturbation of a stable model, but seeks feedback under cumulative cost; its obstruction from uninformative (singular) data parallels ours (Abbasi-Yadkori & Szepesvári, 2011; Mania et al., 2019; Simchowitz & Foster, 2020; Ziemann & Sandberg, 2021).

Selectors acting after the gradient or through surrogates. Greedy Alignment shadow-updates all candidates and selects by normalized gradient alignment, and it relates alignment to one-step descent up to an approximation error (Lee & Lee, 2025); we compare with it in Section 5.2. DOIT uses optimizer-specific surrogates and resets the state at each switch (Chen et al., 2026). Neither bounds the terminal regret against the best fixed optimizer.

To the best of our knowledge, no prior result combines gradients revealed before the action, one augmented state that cannot be reset, a same-budget terminal comparison with the best fixed optimizer, and a learned-routing bound that conditions on the observed gradient (Table 1); ours remains limited to persistently excited unknown quadratics and uniformly contracting affine-linear states.

3 Routing on a single trajectory

Protocol. We have K optimizers, indexed by $[K] = \{1, \dots, K\}$, and a budget of B steps for minimizing f. The full state $z_q \in \mathbb{R}^p$ contains the shared iterate $x_q = Cz_q \in \mathbb{R}^d$ (C is a known readout matrix) and all optimizer memory, such as momentum buffers. At step q, the oracle returns a stochastic gradient g_q at x_q (the *response*), and the router then chooses a method $i_q \in [K]$ that uses g_q to compute z_{q+1}. Each gradient drives exactly one irreversible update, and each candidate memory evolves only through its specified map; hidden parallel trajectories are not allowed.

Formally, on a probability space carrying the oracle noise and any internal randomization, let $\mathcal{F}_q$ be the pre-response history, U_q any randomness the router draws after seeing g_q, and $\mathcal{F}_q^+ = \mathcal{F}_q \vee \sigma(g_q, U_q)$. With $\phi_q := ((Cz_q)^\top, 1)^\top$, we require $\mathcal{F}_q \subseteq \mathcal{F}_q^+ \subseteq \mathcal{F}_{q+1}$, with z_q, ϕ_q $\mathcal{F}_q$-measurable and g_q, i_q, z_{q+1} $\mathcal{F}_q^+$-measurable. A choice is *predictable* if it is made before the current gradient is revealed. Calibration randomization belongs to $\mathcal{F}_q$; our methods break ties by a fixed rule and need no U_q.

Benchmark. For budget B, the value of each fixed optimizer and a best one are

$$J_{i,B}(z_0) = \mathbb{E}[f(CZ_B^{(i)}) \mid Z_0 = z_0], \qquad i_B^\star \in \arg\min_i J_{i,B}(z_0), \tag{1}$$

where $Z^{(i)}$ applies optimizer i after each of the B gradients. The terminal pseudo-regret of the router's trajectory z^{route} is

$$\mathrm{Reg}_B = \mathbb{E}[f(Cz_B^{\mathrm{route}})] - \min_i J_{i,B}(z_0). \tag{2}$$

The minimum is outside the expectation: the benchmark is one fixed optimizer, not an oracle that picks the best fixed path separately for each noise realization. The pseudo-regret may be negative, since switching can beat every fixed method. Expectations are over the oracle noise and any internal randomization; without the latter, a policy is a deterministic function of its observed history.

3.1 Unknown quadratic objectives and linear optimizers

Our positive results concern the model

$$f_{H,b}(x) = \tfrac{1}{2}x^\top Hx + b^\top x + c, \qquad g_q = HCz_q + b + \xi_q, \qquad z_{q+1} = A_{i_q} z_q - D_{i_q} g_q, \tag{3}$$

where ξ_q is the gradient noise; e.g., gradient descent with stepsize η has $C = A_i = I$ and $D_i = \eta I$. The matrices A_i, D_i, C are known, while H, b, and optionally the noise covariance, are learned from the single trajectory. The constant c affects no selection or regret, since both terms in Eq. (2) shift by the same amount; we therefore use the centered objective $\bar{f}_{H,b} = f_{H,b} - c$ in every exact and plug-in score, which is equivalent to setting $c = 0$ in both the true and the fitted model.

Assumption 1 (Noise, stability, and excitation)**.**

(i) Noise. *For every admissible response-before-action policy using the specified update maps and every reachable pre-response history—including the fixed-optimizer histories that define Eq.* (1)*—the gradient noise is a bounded martingale difference with fixed (homoscedastic) conditional covariance:*

$$\mathbb{E}[\xi_q \mid \mathcal{F}_q] = 0, \quad \mathbb{E}[\xi_q \xi_q^\top \mid \mathcal{F}_q] = \Sigma, \quad \|\xi_q\| \le W,$$

and each coordinate is conditionally σ-sub-Gaussian.

(ii) Stability. *For known constants $0 < \mu \le L < \infty$, the unknown pair (H, b) belongs to a known nonempty closed convex class $\mathfrak{T}$ contained in*

$$\left\{(H,b):\ H = H^\top,\ \mu I \preceq H \preceq LI,\ \|b\| \le G,\ \|A_i - D_i HC\|_{\mathrm{op}} \le \rho < 1\ (i \in [K])\right\}.$$

Additional known closed convex restrictions, such as a diagonal Hessian class, are allowed.

(iii) Persistent excitation (PE). *The first $m \ge d+1$ steps follow a predictable calibration policy that uses only portfolio methods. Its observed design is checked against*

$$\sum_{q<m} \phi_q \phi_q^\top \succeq \kappa m I_{d+1}. \tag{4}$$

Write $\mathcal{P}_m(\kappa)$ for the event in Eq. (4)*. In the almost-sure case, $\Pr\{\mathcal{P}_m(\kappa)\} = 1$ at the stated known κ; passing one realized check is not this assumption. Theorem 1 also covers random PE. If the check fails, the algorithm commits to a prespecified fallback $i_{\mathrm{fb}} \in [K]$, which is stable by the same contraction condition.*

Condition (i) constrains the noise on every history that an admissible policy can reach, not only on the one realized by PE–RCR. It is stronger than generic martingale-difference or unbiased minibatch noise, which may have state-, history-, policy-, or batch-dependent conditional covariance and need not be uniformly bounded or sub-Gaussian on all routed and counterfactual histories. Condition (ii) allows nonsymmetric, noncommuting transitions and covers memory that the methods share or transfer. It excludes private memories that are literally frozen while inactive: they add identity blocks to A_i and break uniform Euclidean contraction, which would require a seminorm on the reachable states or a separate finite-horizon sensitivity analysis.

Why directional excitation is needed. Appendix Proposition 4 constructs two Hessians $H_0 = \mathrm{diag}(1, 0.8)$ and $H_1 = \mathrm{diag}(1, 1.2)$ and two methods ($A_i = I$, $D_i = P_i$) with $\|I - P_i H_j\|_{\mathrm{op}} < 0.283$ whose gradients agree along a calibration trajectory confined to one direction, but whose uniquely better next method is reversed. No router can recover the missing curvature from such a history, so some directional information or excitation is necessary for identifying the next method uniformly over this class. This does not show that the full-rank PE condition in Eq. (4) is necessary for every restricted model class.

4 Methods and guarantees

All three methods share calibration, the PE check, and one model fit; they differ only in how they use the fit.

Model fitting. Write $\Theta = [H\ b]$, so that $g_q = \Theta\phi_q + \xi_q$ is a linear regression. After calibration, all three methods fix $\lambda > 0$ and compute the projected ridge estimate

$$V_m = \lambda I + \sum_{q<m} \phi_q \phi_q^\top, \qquad \bar{\Theta}_m = \left(\sum_{q<m} g_q \phi_q^\top \right) V_m^{-1}, \qquad \widehat{\Theta}_m = \Pi_{\mathfrak{T}}(\bar{\Theta}_m), \tag{5}$$

where $\Pi_{\mathfrak{T}}$ is the Euclidean projection onto the convex model class of Assumption 1. Write $\widehat{\Theta}_m = [\widehat{H}\ \widehat{b}]$. With known covariance, $\widehat{\theta} = (\widehat{H}, \widehat{b}, \Sigma)$; the residual-covariance extension uses $\widehat{\theta} = (\widehat{H}, \widehat{b}, \widehat{\Sigma})$.

Value of a fixed continuation. For a candidate i, a model $\theta = (H, b, \Sigma)$, and a state y, let $J^\theta_{i,h}(y)$ be the exact expected *centered* terminal value $\mathbb{E}[\bar{f}_{H,b}(CZ_h)]$ after h further applications of i to gradients not yet observed. Propagating the mean and covariance of the state gives

$$\begin{aligned} F_i(H) &= A_i - D_i H C, \\ \bar{z}_{i,0}(y) &= y, & S_{i,0} &= 0, \\ \bar{z}_{i,k+1}(y) &= F_i(H)\bar{z}_{i,k}(y) - D_i b, \\ S_{i,k+1} &= F_i(H) S_{i,k} F_i(H)^\top + D_i \Sigma D_i^\top, \\ J^\theta_{i,h}(y) &= \tfrac{1}{2}(C\bar{z}_{i,h})^\top H (C\bar{z}_{i,h}) + b^\top C\bar{z}_{i,h} + \tfrac{1}{2}\,\mathrm{tr}(HCS_{i,h}C^\top). \end{aligned} \tag{6}$$

Only the conditional first and second noise moments enter this recursion.

PE–Commit. This deliberately simple baseline selects, at the end of calibration and *before* the gradient g_m is revealed,

$$\widehat{i}_{\mathrm{commit}} \in \arg\min_{i \in [K]} J^{\widehat{\theta}}_{i,B-m}(z_m) \tag{7}$$

and applies $\widehat{i}_{\mathrm{commit}}$ for all remaining updates. PE–Commit versus PE–Post isolates the value of the first post-calibration gradient; PE–Post versus PE–RCR isolates the value of later gradients.

Scores after seeing the gradient. At a decision point, r steps remain and the gradient g at the current state has been observed. The post-response commit score is

$$Q^\theta_{i,r}(z, g) = J^\theta_{i,r-1}(A_i z - D_i g), \qquad \widehat{Q}_{i,r}(z, g) = J^{\widehat{\theta}}_{i,r-1}(A_i z - D_i g). \tag{8}$$

That is, $\widehat{Q}_{i,r}(z, g)$ is the fitted expected centered loss after applying i now, to the observed g, and for the remaining $r - 1$ updates. The same realized g enters every candidate's score and then drives

Inputs. Budget and calibration length $B, m < B$; initial state and readout z_0, C; predictable calibration policy; ridge and PE parameters (λ, κ); portfolio $\{(A_i, D_i)\}_{i=1}^K$; model class $\mathfrak{T}$; known Σ or the residual-covariance estimator; stable fallback i_{fb}; and a fixed tie-breaking rule.

1. **Calibrate.** Run $m < B$ steps (m gradients and m updates) with the predictable calibration policy, which uses only portfolio methods. Check Eq. (4); if it fails, declare "not identified" and apply the prespecified stable method i_{fb} for every remaining update.
2. **Fit.** Compute $(\widehat{H}, \widehat{b})$ by Eq. (5); use the known Σ, or the residual covariance estimator in Appendix A.5.
3. **Route.** Set $r = B - m$. At the start of each block, observe g, choose $\widehat{i} \in \arg\min_i \widehat{Q}_{i,r}(z, g)$, and use $\widehat{i}$ for $\ell(r) = \max\{1, \lfloor r/2 \rfloor\}$ consecutive updates, including the update driven by the observed g.
4. Decrease r by $\ell(r)$ and repeat Step 3 until the budget is exhausted.

Output. The final state z_B, equivalently the final iterate $x_B = Cz_B$.

Algorithm 1: PE–RCR: learned routing on one trajectory

the selected update. The score conditions on g rather than averaging over it, which would be wrong because the selected method is not independent of its own noise. All finite argmins use a fixed tie-breaking rule.

PE–Post and PE–RCR. PE–Post observes the first post-calibration gradient g_m, selects

$$\widehat{i}_{\mathrm{post}}(g_m) \in \operatorname*{arg\,min}_{i \in [K]} \widehat{Q}_{i,B-m}(z_m, g_m), \tag{9}$$

and applies this method for all $B - m$ remaining updates, including the one driven by g_m. Its choice is $\mathcal{F}_m \vee \sigma(g_m)$-measurable (hence $\mathcal{F}_m^+$-measurable) rather than predictable with respect to g_m, but it makes no later decision. PE–RCR (Algorithm 1) reuses the same score at the start of every block, with block lengths that halve.

PE–RCR makes at most $N_R \le \lceil \log_2 R \rceil + 1$ routing decisions for $R = B - m$. Within a block its actions are predictable; at a block start, Eq. (8) accounts exactly for the noise of the gradient that drives the selected update.

Computational cost. With dense streaming algebra, fitting and checking PE cost $O(md^2 + d^3 + \mathrm{time}_\Pi)$ time and $O(d^2 + \mathrm{mem}_\Pi)$ workspace, where $\mathrm{time}_\Pi, \mathrm{mem}_\Pi$ are the cost and workspace of one projection. Direct PE–RCR scoring costs $O(KRp^3)$; exact multi-step summaries built by repeated doubling reduce this to $O(Kp^3 \log(1+R))$ time and $O(Kp^2 \log(1+R))$ cache, and the updates add $O(B(p^2 + pd))$. PE–Commit and PE–Post score one horizon and stream with $O(p^2 + K)$ extra workspace. The oracle complexity is always exactly B gradients and B updates (Appendix A).

4.1 Main result

Let $\zeta_0 = \|z_0\|$, $c_D = \max_i \|D_i\|_{\mathrm{op}}$, $c_C = \|C\|_{\mathrm{op}}$, and define the known state and loss bounds

$$\begin{aligned} \zeta_{\mathrm{cal}} &= \zeta_0 + \frac{c_D(G+W)}{1-\rho}, & \zeta_{\mathrm{fix}} &= \zeta_0 + \frac{c_D(G+W)}{1-\rho}, \\ X &= c_C \zeta_{\mathrm{cal}}, & \Lambda &= \tfrac{1}{2} L X^2 + GX. \end{aligned} \tag{10}$$

(The two state bounds coincide here but differ when calibration may inject extra inputs; see Appendix A.) Put $\Phi = \sqrt{X^2 + 1}$, $\bar{G} = \sqrt{L^2 + G^2}$, and

$$\begin{aligned} \beta_m(\delta) &= \sigma \sqrt{2 \log \frac{d}{\delta} + (d+1) \log\left(1 + \frac{m\Phi^2}{\lambda(d+1)}\right)} + \sqrt{\lambda}\bar{G}, \\ e_m(\delta) &= \frac{\sqrt{d}\,\beta_m(\delta)}{\sqrt{\lambda + \kappa m}}. \end{aligned} \tag{11}$$

With probability at least $1 - \delta$, $e_m(\delta)$ bounds the joint error $\|[\widehat{H} - H \;\; \widehat{b} - b]\|_F$, and hence every score error is at most $\Gamma e_m(\delta)$, where $\Gamma = \Gamma_H + \Gamma_b$ is an explicit constant independent of the horizon (Appendix Eq. (45) and Lemma 3). Since a router may beat every fixed optimizer, we bound the

positive part $[u]_+ = \max\{u, 0\}$. As Eq. (2) already averages over all randomness, the bounds below are deterministic bounds on the expected pseudo-regret, not high-probability bounds on the realized regret; confidence failure enters only through $2\Lambda\delta$.

Theorem 1 (Deterministic finite-sample bounds on expected pseudo-regret)**.** *Let Assumption 1 hold with known Σ, let $m < B$ and $0 < \delta < 1$, and define $\Psi_B(\delta) = (LX+G)c_C(\zeta_{\mathrm{cal}}+\zeta_{\mathrm{fix}})\rho^{B-m}+2\Lambda\delta$. Then the expected pseudo-regrets of PE–Commit, PE–Post, and PE–RCR (Algorithm 1) satisfy the deterministic inequalities*

$$\boxed{\begin{aligned}[\mathrm{Reg}_B^{\mathrm{commit}}]_+ &\le \Psi_B(\delta) + 2\Gamma e_m(\delta),\\ [\mathrm{Reg}_B^{\mathrm{post}}]_+ &\le \Psi_B(\delta) + 2\Gamma e_m(\delta),\\ \mathrm{Reg}_B \le [\mathrm{Reg}_B]_+ &\le \Psi_B(\delta) + 2N_{B-m}\Gamma e_m(\delta).\end{aligned}} \tag{12}$$

All three methods use exactly B gradients and B updates, with no idle steps (holds), resets, model copies, or counterfactual rollouts. The display assumes $\Pr\{\mathcal{P}_m(\kappa)\} = 1$. If instead $\Pr\{\mathcal{P}_m(\kappa)\} \ge 1 - \delta_{\mathrm{PE}}$ for a given $\delta_{\mathrm{PE}} \in [0, 1]$ at the known κ, and a failed check commits to the stable fallback i_{fb} for the remaining updates, each right-hand side gains $2\Lambda\delta_{\mathrm{PE}}$. An estimated residual covariance adds the term in Appendix A.5. Full proofs are Appendix Theorems 2, 3, and 4.

Discussion. In $\Psi_B(\delta)$, the first term is the calibration cost, forgotten at the rate ρ^{B-m}, and $2\Lambda\delta$ covers confidence failure; each fitted decision costs $2\Gamma e_m(\delta)$. The theorem is a hierarchy of safety guarantees, not a performance ranking. With $m = \lfloor B/2 \rfloor$, $\delta = B^{-2}$, and fixed $\kappa > 0$, all three positive parts are $\widetilde{O}(B^{-1/2})+O(\rho^{B/2})$, where $\widetilde{O}$ hides logarithms; in particular, $\limsup_B \mathrm{Reg}_B \le 0$, not two-sided convergence. PE–RCR pays N_{B-m} estimation terms instead of one in exchange for access to later gradients, so the theorem alone gives no reason to reroute. The margin transfer below is a sufficient condition for an exact-model advantage to carry over to the learned policies, and Proposition 1 is an exact instance of strict non-myopic value.

Random PE. Appendix Proposition 3 uses round-robin calibration, $E_q = 1 + (q \bmod K)$ (method E_q at step q). If, for a known $\kappa_0 > 0$ and burn-in s, $\mathbb{E}[\phi_q\phi_q^\top \mid \mathcal{F}_{q-1}] \succeq \kappa_0 I$ almost surely for $q = s, \dots, m-1$, then Eq. (4) holds with $\kappa = \kappa_0(m-s)/(2m)$ except with probability at most $(d+1)\exp\{-\kappa_0^2(m-s)/(32\Phi^4)\}$, where $\|\phi_q\| \le \Phi$. A conditional small-ball condition with known $\pi_0, \alpha > 0$, $\Pr(|u^\top\phi_q| \ge \alpha \mid \mathcal{F}_{q-1}) \ge \pi_0$ for all unit u and those q on one common probability-one event, implies this premise with $\kappa_0 = \pi_0\alpha^2$. Neither stability nor nonzero noise implies it; without directional excitation, the lower bound of Section 3 applies.

Margin transfer. Let exact PE–RCR and PE–Post use the true model instead of the fitted one (with the same PE fallback), and put $\Delta_B = \mathbb{E}[f(CZ_B^{\mathrm{post},\star}) - f(CZ_B^{\mathrm{rcr},\star})]$. If, jointly with a PE pass, a score error above $\Gamma e_m(\delta)$ has probability at most δ and an exact PE–RCR score gap of at most $2\Gamma e_m(\delta)$ at some decision point has probability at most δ_{M}, then $\mathbb{E}[f(Cz_B^{\mathrm{rcr}}) - f(Cz_B^{\mathrm{post}})] \le -\Delta_B + 4\Lambda(\delta+\delta_{\mathrm{M}})$, with no independence required. Learned PE–RCR is thus strictly better when the right-hand side is negative. We do not estimate δ_{M}, so this does not certify Figure 1; Appendix Corollary 1 gives the coupling, events, proof, and an always-routing variant.

Proposition 1 (Strict non-myopic value)**.** *There exist a one-dimensional strongly convex quadratic, two strictly contractive gradient maps, and bounded i.i.d. zero-mean noise such that, for $B = 2$, exact receding-horizon commit with unit blocks is the unique optimum over the 64 outcome-distinct deterministic response-contingent plans and*

$$\underbrace{\frac{143}{4320}}_{\textit{receding}} < \underbrace{\frac{35}{864}}_{\textit{response-aware one-step greedy}} < \underbrace{\frac{67}{864}}_{\textit{predictable minimum = PE–Post-style commit minimum}}.$$

After the low first gradient, the immediate loss uniquely prefers the aggressive map, whereas the two-update commit score uniquely prefers the conservative map; the respective ranking gaps are $20/81$ and $19/81$. The smallest receding score gap over both decision points is $10/81$, so any uniform score error below $5/81$ preserves the full route.

Appendix Proposition 2 enumerates the policy classes, derives both score rules, and identifies the unique optimum. The result separates receding-horizon commit scoring from exact one-step greedy

selection and from policies that stop adapting after the first gradient; it does not claim that the optimal state–gradient policy lacks a reactive representation, and it is separate from the consistency bound. Appendix Remark 2 lists what Lean 4 verifies (these exact finite claims and the deterministic assembly of the theorem) and what is proved by hand (concentration, the selected-noise lemma, PE, contraction, and the continuous stochastic model).

Proof sketch. Let $M_r^\theta(z) = \min_i J_{i,r}^\theta(z)$. At a block start, conditionally on the observed g, plug-in minimization gives

$$Q_{\widehat{i},r}^\theta(z,g) \le \min_i Q_{i,r}^\theta(z,g) + 2\Gamma e_m(\delta).$$

After the block, $M_{r-\ell}^\theta$ is at most the value of continuing with the selected optimizer, and averaging over g uses only the valid direction $\mathbb{E}_g \min_i Q_{i,r} \le \min_i \mathbb{E}_g Q_{i,r} = M_r$. Telescoping over the N_{B-m} blocks gives the statistical term, and comparing the calibration state with the B-step best fixed trajectory through the Lipschitz bound for fixed continuations (Appendix Lemma 5) gives the exponentially forgotten calibration term. PE–Post applies the block lemma once with $r = \ell = B-m$; since M_0 is the terminal centered loss, this gives exactly one $2\Gamma e_m(\delta)$ term. PE–Commit instead applies the fixed-score plug-in inequality once to $J_{i,B-m}(z_m)$. Both avoid the factor N_{B-m}, although only PE–Post conditions on g_m. Appendix A gives full proofs, including the one-trajectory confidence bound and the covariance perturbation.

5 Experiments

Each step (a "primitive" in the figure labels) makes one common oracle call and one update of the actual state; candidate losses and counterfactual paths are computed only for reporting. Paired noise realizations are averaged within each task, and intervals are taken across independent tasks. Appendix B gives the task generators, budget accounting, and validation scripts.

5.1 Quadratic tasks with shared optimizer memory

Do later gradients help when noncommuting methods pass genuine memory to each other? Within the setting of Theorem 1, four noncommuting methods read and overwrite one memory in $z = (x, v) \in \mathbb{R}^4$, with no dormant blocks or shadow copies; the known contraction bound is $0.99201 < 0.995$, and every run uses exactly B gradients and B updates. PE–Post and PE–RCR share calibration, fit, and their first decision state, gradient, scores, and action. The generator and methods were tuned earlier and then frozen; before this new draw, an internal lock fixed the unchanged code, fresh seeds, the primary comparison, and its all-budget success criterion. This within-generator experiment was neither designed independently nor registered externally, and it makes no population claim. On 64 new tasks with 16 paired noise realizations each, replay confirms that PE–Post and PE–RCR coincide at the first decision point in all 3,072 cases. PE–RCR minus PE–Post at $B = 64, 128, 256$ is $-1.151[-1.285, -1.017] \times 10^{-4}$, $-2.616[-3.019, -2.213] \times 10^{-5}$, and $-2.049[-2.432, -1.666] \times 10^{-5}$ (mean and 95% interval): all upper endpoints are negative, meeting the pre-specified criterion, and 64/64, 63/64, and 62/64 task means favor PE–RCR. Appendix Sections B.2–B.3 give diagnostics, secondary comparisons, and a post hoc analysis of the scoring horizon.

5.2 Beyond the theory: nonlinear problems with private state

We test 24 rotated, rippled Rosenbrock functions and 24 small tanh networks on generated XOR data, with SGD, two heavy-ball methods, Nesterov, and AdamW. Inactive private states stay frozen, except for Greedy Alignment's three shadow filters. Six noise realizations per task select the fixed comparator, and three disjoint paired ones form each task mean. The oracle returns one common noisy value–gradient pair per step, and every run uses B oracle calls and B updates without querying candidate points.

A single switch at a fixed time, designed only on separate pilot tasks, transfers to the new tasks: Nesterov-to-SGD at $0.75B$ improves Rosenbrock by -5.93×10^{-5} $[-7.90, -3.95] \times 10^{-5}$, and AdamW-to-heavy-ball at $0.65B$ improves the neural loss by -0.00889 $[-0.01266, -0.00512]$. This is evidence for a fixed (open-loop) schedule, not for a learned router. Compared post hoc only with

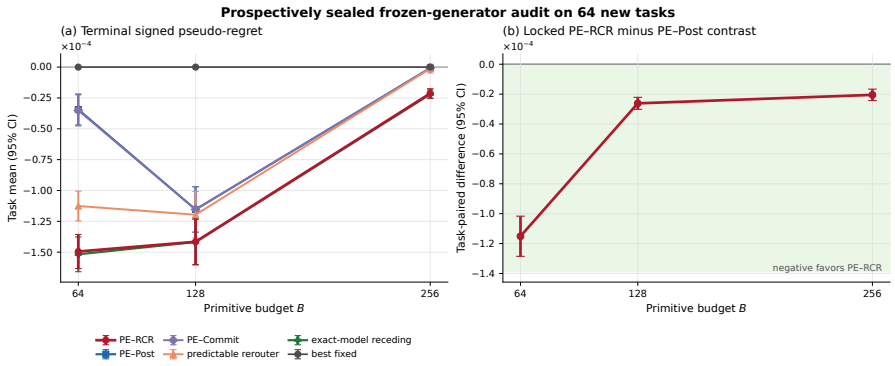

Figure 1: Pre-specified experiment on 64 new tasks from the tuned and then frozen generator (not an independent confirmation). Left: terminal pseudo-regret. Right: the pre-specified difference PE–RCR minus PE–Post; their shared first action isolates later decisions. Bars: 95% intervals across tasks, each averaged over 16 paired noise realizations; negative favors PE–RCR.

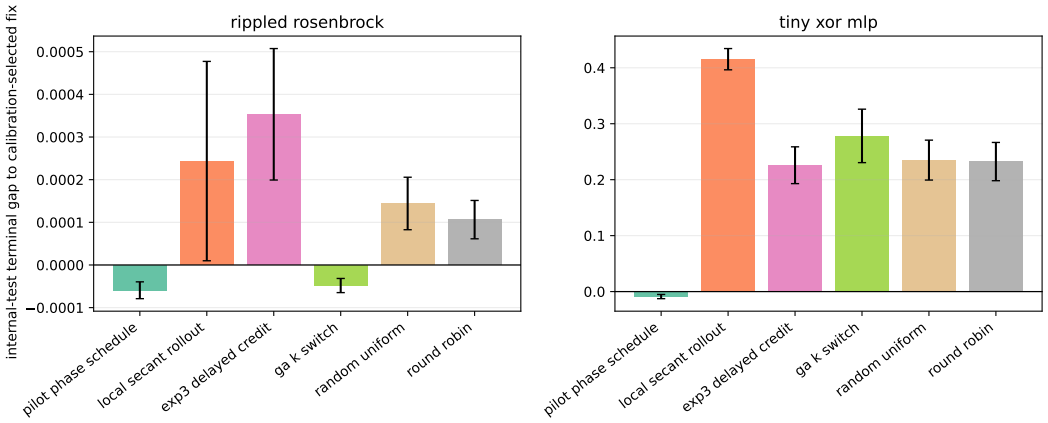

Figure 2: Terminal gap to the fixed method selected on calibration runs, on 24 new test tasks per family. Bars: 95% intervals across tasks, each averaged over three paired noise realizations; negative favors the policy. The pilot schedule is fixed in advance. For context, the Greedy Alignment bar uses the common five-method comparator; Appendix B.4 restricts it to its own action set.

its own three momentum methods, Greedy Alignment has gap $-1.082[-1.396, -0.768] \times 10^{-4}$ on Rosenbrock but $0.2503[0.2105, 0.2901]$ on the neural tasks (Appendix B.4). Two learned routers for the full portfolio, a local-secant rollout and delayed-credit EXP3, do not improve: their unadjusted descriptive intervals lie above zero. With memory set to zero, the best method differs between the initial and switching points on 24/24 Rosenbrock and 16/24 neural tasks, but this diagnostic discards the current memory, and the neural winner intervals overlap. Exploiting such phases with a learned router outside the stable linear setting thus remains open. Oracle budgets match but local computation does not: local-secant simulates $5B$ updates, and Greedy Alignment maintains three filters.

The appendix reports further experiments, including negative ones, all with reproduction commands, independent validation scripts, and checksums.

6 Limitations and Conclusion

The guarantees require contractive affine-linear quadratic dynamics, persistent excitation, and bounded noise with a fixed covariance. A failed PE check only leads to the fallback, the constants are vacuous, and the projection is not counted in the oracle budget. The evidence is internal and synthetic: the generator was tuned, the tasks are small, pilot tasks set the schedule, both learned

routers fail on nonlinear problems, and the best fixed optimizer is a weaker benchmark than the best switching policy. The theory and the pre-specified experiment therefore establish the value of later gradients only within a limited scope, not practical dominance or population-level evidence. Within that scope, we give a budget-matched guarantee and an exact case in which later gradients and the scoring horizon change an irreversible optimizer choice.

AI USE STATEMENT

The authors used Claude for paper editing and language polishing. OpenAI GPT Sol was used for initial draft preparation: initial idea brainstorming and an initial search for related papers, as well as proof verification via Lean and drafting full proofs from proofs ideas provided by authors. The authors reviewed all AI-assisted work and take responsibility for the final text, claims, code, and artifacts.

REPRODUCIBILITY STATEMENT

The problem setup, algorithms, assumptions, and computational cost are stated in the main paper. The appendix gives complete proofs, the exact finite constructions, experimental protocols, checks of the estimators and of the budget accounting, and additional results, including negative ones. The anonymized supplement contains the experiment code and independent validation scripts, pinned Python dependencies, exact reproduction commands, and the Lean 4/mathlib verification described in the paper.

REFERENCES

Yasin Abbasi-Yadkori and Csaba Szepesvári. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, volume 19 of *Proceedings of Machine Learning Research*, pp. 1–26, 2011. URL https://proceedings.mlr.press/v19/abbasi-yadkori11a.html.

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Online least squares estimation with self-normalized processes: An application to bandit problems. *arXiv preprint arXiv:1102.2670*, 2011. URL https://arxiv.org/abs/1102.2670.

Steven Adriaensen, André Biedenkapp, Gresa Shala, Noor Awad, Theresa Eimer, Marius Lindauer, and Frank Hutter. Automated dynamic algorithm configuration. *Journal of Artificial Intelligence Research*, 75:1633–1699, 2022. doi: 10.1613/jair.1.13922. URL https://doi.org/10.1613/jair.1.13922.

Alekh Agarwal, Haipeng Luo, Behnam Neyshabur, and Robert E. Schapire. Corralling a band of bandit algorithms. In *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, pp. 12–38, 2017. URL https://proceedings.mlr.press/v65/agarwal17b.html.

Aymen Al Marjani, Aurélien Garivier, and Alexandre Proutiere. Navigating to the best policy in markov decision processes. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/d9896106ca98d3d05b8cbdf4fd8b13a1-Abstract.html.

Denis Antipov and Carola Doerr. When switching algorithms helps: A theoretical study of online algorithm selection. In *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2026. doi: 10.1145/3795095.3805156. URL https://doi.org/10.1145/3795095.3805156.

André Biedenkapp, H. Furkan Bozkurt, Theresa Eimer, Frank Hutter, and Marius Lindauer. Dynamic algorithm configuration: Foundation of a new meta-algorithmic framework. In *ECAI 2020*, pp. 427–434, 2020. doi: 10.3233/FAIA200122. URL https://doi.org/10.3233/FAIA200122.

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Naz Sepah, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Dmitriy Serdyuk, Tal Arbel, Chris Pal, Gaël Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In *Proceedings of Machine Learning and Systems*, volume 3, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/hash/0184b0cd3cfb185989f858a1d9f5c1eb-Abstract.html.

Xinyi Chen and Elad Hazan. Online control for meta-optimization. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/745b7e084d5ca5afc07fb454ab2be522-Abstract-Conference.html.

Yile Chen, Zeyi Wen, Jian Chen, and Jin Huang. Towards dynamic interleaving optimizers. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=AII8ADdDHt. ICLR 2026 Poster.

Ashok Cutkosky, Christoph Dann, Abhimanyu Das, Claudio Gentile, Aldo Pacchiano, and Manish Purohit. Dynamic balancing for model selection in bandits and RL. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2276–2285, 2021. URL https://proceedings.mlr.press/v139/cutkosky21a.html.

Catalin-Viorel Dinu, Diederick Vermetten, and Carola Doerr. How sequential algorithm portfolios can benefit black box optimization. *arXiv preprint arXiv:2601.16896*, 2026. URL https://arxiv.org/abs/2601.16896. Version 1, submitted 23 January 2026.

Grant Getzelman and Prasanna Balaprakash. Learning to switch optimizers for quadratic programming. In *Proceedings of the 13th Asian Conference on Machine Learning*, volume 157 of *Proceedings of Machine Learning Research*, pp. 1553–1568. PMLR, 2021. URL https://proceedings.mlr.press/v157/getzelman21a.html.

Pascal Kerschke, Holger H. Hoos, Frank Neumann, and Heike Trautmann. Automated algorithm selection: Survey and perspectives. *Evolutionary Computation*, 27(1):3–45, 2019. doi: 10.1162/evco_a_00242. URL https://doi.org/10.1162/evco_a_00242.

Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from Adam to SGD. *arXiv preprint arXiv:1712.07628*, 2017. URL https://arxiv.org/abs/1712.07628.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The Third International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Ana Kostovska, Anja Jankovic, Diederick Vermetten, Jacob de Nobel, Hao Wang, Tome Eftimov, and Carola Doerr. Per-run algorithm selection with warm-starting using trajectory-based features. In *Parallel Problem Solving from Nature – PPSN XVII*, volume 13398 of *Lecture Notes in Computer Science*, pp. 46–60, 2022. doi: 10.1007/978-3-031-14714-2_4. URL https://doi.org/10.1007/978-3-031-14714-2_4.

Jaerin Lee and Kyoung Mu Lee. Greedy alignment principle for optimizer selection. *arXiv preprint arXiv:2512.06370*, 2025. URL https://arxiv.org/abs/2512.06370. Version 3, revised 7 May 2026.

Per Kristian Lehre and Ender Özcan. A runtime analysis of simple hyper-heuristics: To mix or not to mix operators. In *Proceedings of the 12th ACM Workshop on Foundations of Genetic Algorithms*, pp. 97–104, 2013. doi: 10.1145/2460239.2460249. URL https://doi.org/10.1145/2460239.2460249.

Yingying Li, James A. Preiss, Na Li, Yiheng Lin, Adam Wierman, and Jeff S. Shamma. Online switching control with stability and regret guarantees. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 1138–1151. PMLR, 2023. URL https://proceedings.mlr.press/v211/li23a.html.

Yiheng Lin, James A. Preiss, Emile Anand, Yingying Li, Yisong Yue, and Adam Wierman. Online adaptive policy selection in time-varying systems: No-regret via contractive perturbations. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53508–53521, 2023. doi: 10.52202/075280-2328. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a7a7180fe7f82ff98eee0827c5e9c141-Abstract-Conference.html. Full version includes the finite-policy BAPS analysis in Appendix G.

Yiheng Lin, James A. Preiss, Fengze Xie, Emile Anand, Soon-Jo Chung, Yisong Yue, and Adam Wierman. Online policy optimization in unknown nonlinear systems. In *Proceedings of the Thirty-Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pp. 3475–3522. PMLR, 2024. URL https://proceedings.mlr.press/v247/lin24a.html.

Andrei Lissovoi, Pietro S. Oliveto, and John Alasdair Warwicker. On the time complexity of algorithm selection hyper-heuristics for multimodal optimisation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 2322–2329, 2019. doi: 10.1609/aaai.v33i01.33012322. URL https://doi.org/10.1609/aaai.v33i01.33012322.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *The Seventh International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Horia Mania, Stephen Tu, and Benjamin Recht. Certainty equivalence is efficient for linear quadratic control. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/5dbc8390f17e019d300d5a162c3ce3bc-Abstract.html.

Kazem Meidani, Seyedali Mirjalili, and Amir Barati Farimani. MAB-OS: Multi-armed bandits metaheuristic optimizer selection. *Applied Soft Computing*, 128:109452, 2022. doi: 10.1016/j.asoc.2022.109452. URL https://doi.org/10.1016/j.asoc.2022.109452.

Yurii Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Kaan Ozkara, Can Karakus, Parameswaran Raman, Mingyi Hong, Shoham Sabach, Branislav Kveton, and Volkan Cevher. MADA: Meta-adaptive optimizers through hyper-gradient descent. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 38983–39008. PMLR, 2024. URL https://proceedings.mlr.press/v235/ozkara24a.html.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. doi: 10.1016/0041-5553(64)90137-5.

Isabeau Prémont-Schwarz, Jaroslav Vítků, and Jan Feyereisl. A simple guard for learned optimizers. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17910–17925. PMLR, 2022. URL https://proceedings.mlr.press/v162/premont-schwarz22a.html.

John R. Rice. The algorithm selection problem. In *Advances in Computers*, volume 15, pp. 65–118. Elsevier, 1976. doi: 10.1016/S0065-2458(08)60520-3. URL https://doi.org/10.1016/S0065-2458(08)60520-3.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951. doi: 10.1214/aoms/1177729586.

Robin M. Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley – benchmarking deep learning optimizers. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9367–9376. PMLR, 2021. URL https://proceedings.mlr.press/v139/schmidt21a.html.

Max Simchowitz and Dylan Foster. Naive exploration is optimal for online LQR. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8937–8948, 2020. URL https://proceedings.mlr.press/v119/simchowitz20a.html.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. doi: 10.1007/s10208-011-9099-z. URL https://arxiv.org/abs/1004.4389.

Diederick Vermetten, Hao Wang, Carola Doerr, and Thomas Bäck. Towards dynamic algorithm selection for numerical black-box optimization: Investigating BBOB as a use case. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 654–662, 2020. doi: 10.1145/3377930.3390189. URL https://doi.org/10.1145/3377930.3390189.

Diederick Vermetten, Hao Wang, Kevin Sim, and Emma Hart. To switch or not to switch: Predicting the benefit of switching between algorithms based on trajectory features. In *Applications of Evolutionary Computation*, pp. 335–350. Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-30229-9_22. URL https://arxiv.org/abs/2302.09075.

Kaiyue Wen, David Leo Wright Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2J51qUZ0iG.

Peng Zhao, Yu-Hu Yan, Yu-Xiang Wang, and Zhi-Hua Zhou. Non-stationary online learning with memory and non-stochastic control. *Journal of Machine Learning Research*, 24(206):1–70, 2023. URL https://jmlr.org/papers/v24/22-0218.html.

Ingvar Ziemann and Henrik Sandberg. On uninformative optimal policies in adaptive LQR with unknown B-matrix. In *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, volume 144 of *Proceedings of Machine Learning Research*, pp. 213–226, 2021. URL https://proceedings.mlr.press/v144/ziemann21a.html.

Weidong Zou, Yuanqing Xia, Bineng Zhong, and Weipeng Cao. AdaSwitch: Adapting switch from Adam to SGDM by exponential function. *Applied Soft Computing*, 189:114459, 2026. doi: 10.1016/j.asoc.2025.114459. URL https://doi.org/10.1016/j.asoc.2025.114459.

CONTENTS OF THE APPENDIX

A Full specification and proofs for PE–Commit, PE–Post, and PE–RCR **15**

A.1 Model, budget accounting, and target 15

A.2 Estimation, scores, and the three learned rules 17

A.3 Confidence and score sensitivity 20

A.4 Potential decrease and pseudo-regret under the full budget 22

A.5 Extension to an unknown noise covariance 28

A.6 Necessity results and limitations 29

B Experimental details **30**

B.1 Common protocol: budget, target quantity, and reporting 30

B.2 Pre-specified experiment with shared optimizer memory 31

B.3 Post hoc analysis of the scoring horizon 32

B.4 Nonlinear problems with private optimizer state 32

B.5 Pre-specified confirmation on random tasks within the theory 34

B.6 Single-task check of PE–RCR under random excitation 35

B.7 Small datasets with a split-sample comparator 36

B.8 Code, data, and independent validation 37

A FULL SPECIFICATION AND PROOFS FOR PE–COMMIT, PE–POST, AND PE–RCR

This appendix states the complete assumptions, specifies the algorithms in full, and proves our main result on learned routing. Oracle calls and updates are counted against one budget, and the best fixed comparator uses the same budget. At each decision point, the proofs condition on the gradient used to select the next optimizer; since this gradient also drives the selected update, its noise is not treated as an independent draw.

A.1 MODEL, BUDGET ACCOUNTING, AND TARGET

Let $z_q \in \mathbb{R}^p$ be the full state of the actual trajectory and let $x_q = Cz_q \in \mathbb{R}^d$ be the common query point, where $C \in \mathbb{R}^{d\times p}$ is known. The state is common to all methods and can be augmented by a shared momentum or other linear memory; there is no optimizer state outside z_q. Up to an irrelevant additive constant, the unknown objective is

$$f_{H,b}(x) = \frac{1}{2}x^\top Hx + b^\top x + c, \qquad H = H^\top \succ 0. \tag{13}$$

We work on a probability space that carries the oracle noise and any specified internal randomization of the calibration policy or the router. At step q, let $\mathcal{F}_q$ be the pre-response history, let U_q be any random bits that the router draws after observing the gradient, and define $\mathcal{F}_q^+ = \mathcal{F}_q \vee \sigma(g_q, U_q)$. The oracle returns the stochastic gradient

$$g_q = HCz_q + b + \xi_q. \tag{14}$$

After observing g_q, the router chooses a method $i_q \in [K]$ and applies the corresponding linear optimizer map

$$z_{q+1} = A_{i_q} z_q - D_{i_q} g_q. \tag{15}$$

The filtrations are nested, $\mathcal{F}_q \subseteq \mathcal{F}_q^+ \subseteq \mathcal{F}_{q+1}$: z_q and $\phi_q := ((Cz_q)^\top, 1)^\top$ are $\mathcal{F}_q$-measurable, while g_q, i_q, and z_{q+1} are $\mathcal{F}_q^+$-measurable. Random bits that randomize a predictable calibration index are

drawn before g_q is revealed and are included in $\mathcal{F}_q$. The three fitted rules break ties by a fixed rule, so they need no post-response randomization U_q. Hence the response-before-action timing holds literally. All matrices $A_i \in \mathbb{R}^{p\times p}$, $D_i \in \mathbb{R}^{p\times d}$, and C are known, and the update map is specified for every coordinate of z_q. In particular, a portfolio whose separate memories stay frozen while dormant must encode these identity blocks in A_i; see Remark 8 for the consequence.

Every step $q = 0, \dots, B-1$ makes one oracle call and one update of the actual state. Hence both the number of oracle calls and the number of updates equal B. We use no frozen model copy, hold (a step without an update), reset, parallel shadow trajectory, or validation call outside the budget. The terminal value $f(Cz_B)$ is a mathematical evaluation and is not fed back to the router. If an implementation needs an oracle call to measure it, this extra call counts as oracle call $B+1$, for the router and every comparator alike.

The value of a fixed optimizer and the terminal pseudo-regret are

$$J_{i,r}^{H,b}(z) = \mathbb{E}_{H,b}[f_{H,b}(CZ_r) - c \mid Z_0 = z, i_0 = \cdots = i_{r-1} = i], \tag{16}$$

$$\mathrm{Reg}_B = \mathbb{E}_{H,b}[f_{H,b}(Cz_B^{\mathrm{route}}) - c] - \min_{i\in[K]} J_{i,B}^{H,b}(z_0). \tag{17}$$

The benchmark is a minimum of expectations, not the expectation of a minimum taken separately for each noise realization. The pseudo-regret is signed: an adaptive router can outperform every fixed optimizer. We write $[u]_+ = \max\{u, 0\}$ for the positive part, which appears in the one-sided learning guarantees. Every expectation is taken over the oracle noise and any specified internal randomization of the calibration policy or the router; without internal randomization, the policy is deterministic given its observed history. Consequently, every pseudo-regret theorem below is a deterministic inequality between expected terminal values. The proofs use high-probability estimation events and charge their failures to deterministic penalty terms that scale with the range of the loss; they do not give high-probability bounds on the realized pathwise regret.

A.1.1 Standing assumptions

We deliberately state the following assumptions before the theorems.

(A1) Uniformly over every admissible response-before-action policy using the specified update maps and every reachable pre-response history, including histories on the fixed counterfactual continuations in Eq. (17), the noise is a homoscedastic martingale difference:

$$\mathbb{E}[\xi_q \mid \mathcal{F}_q] = 0, \qquad \mathbb{E}[\xi_q\xi_q^\top \mid \mathcal{F}_q] = \Sigma, \qquad \|\xi_q\|_2 \le W \quad \text{a.s.}$$

Every scalar coordinate is conditionally σ-sub-Gaussian under the same uniform quantifiers. Thus (A1) specifies the oracle kernel on every history used by the proofs, not only on the history realized by PE–RCR. The assumption is stronger than a generic martingale-difference or unbiased minibatch-gradient model: such models may have history-dependent conditional covariance and need not provide uniform boundedness or sub-Gaussian control on all counterfactual histories. The covariance Σ is initially treated as known; Corollary 4 removes this assumption.

(A2) For known constants $0 < \mu \le L < \infty$, the unknown pair belongs to a known nonempty closed convex class

$$\emptyset \ne \mathfrak{T} \subseteq \mathfrak{T}_0 := \left\{ (H,b) : \begin{array}{l} H = H^\top, \quad \mu I \preceq H \preceq LI, \quad \|b\|_2 \le G, \\ \|A_i - D_i H C\|_{\mathrm{op}} \le \rho < 1 \quad \text{for every } i \end{array} \right\}. \tag{18}$$

Write $c_D = \max_i \|D_i\|_{\mathrm{op}}$, $c_C = \|C\|_{\mathrm{op}}$, and assume $\|z_0\|_2 \le \zeta_0$ (the main text takes $\zeta_0 = \|z_0\|_2$). Convexity follows because spectral-norm sublevel sets of affine maps of H are convex. Thus $\mathfrak{T}_0$ is closed and convex; $\mathfrak{T}$ may impose additional known closed convex restrictions, such as a diagonal Hessian class.

(A3) The first $m \ge d+1$ actions follow a prespecified calibration policy that uses only portfolio methods. Each index $E_q \in [K]$ is $\mathcal{F}_q$-measurable, that is, fixed before g_q is revealed, and

$$z_{q+1} = A_{E_q} z_q - D_{E_q} g_q, \qquad q < m. \tag{19}$$

For the $\mathcal{F}_q$-measurable $\phi_q = ((Cz_q)^\top, 1)^\top \in \mathbb{R}^{d+1}$, the observed design is checked against

$$\sum_{q=0}^{m-1} \phi_q \phi_q^\top \succeq \kappa m I_{d+1}. \tag{20}$$

Write $\mathcal{P}_m(\kappa)$ for the event in Eq. (20). For the almost-sure PE theorems, $\Pr\{\mathcal{P}_m(\kappa)\} = 1$ at the stated known κ; passing one realized check is not this premise. The algorithm checks this condition. If the check fails, the algorithm reports "not identified" and applies a prespecified portfolio method $i_{\mathrm{fb}} \in [K]$ for every remaining update. This fallback is stable by (A2). Without an a priori bound on the probability that PE fails, a finite collection of passed checks alone implies no vanishing guarantee for the positive part of the pseudo-regret. Proposition 4 shows that some directional information or excitation is necessary for identifying the next method uniformly over the stated class; it does not claim that this exact full-rank PE condition is necessary in every restricted class.

Assumptions (A1)–(A2) give the pathwise bounds

$$\zeta_{\mathrm{cal}} = \zeta_0 + \frac{c_D(G+W)}{1-\rho}, \qquad \zeta_{\mathrm{fix}} = \zeta_0 + \frac{c_D(G+W)}{1-\rho}, \tag{21}$$

$$\zeta = \zeta_{\mathrm{cal}}, \qquad X = c_C \zeta, \qquad \Lambda = \frac{1}{2} L X^2 + G X. \tag{22}$$

Indeed, every candidate next state satisfies

$$\|A_i z - D_i g\| = \|(A_i - D_i H C) z - D_i b - D_i \xi\| \le \rho\zeta + c_D(G+W) \le \zeta$$

for every $i \in [K]$ and every $\|z\| \le \zeta$, even when the selected index is chosen after seeing ξ. Consequently, $\|z_q\| \le \zeta$ on the routed trajectory, $\|Z_q^{(i)}\| \le \zeta_{\mathrm{fix}}$ on every fixed-optimizer trajectory, and $|f_{H,b}(Cz) - c| \le \Lambda$ whenever $\|z\| \le \zeta$.

Remark 1 (Optional probing is an extension, not part of the basic model). If an application explicitly allows a predictable additive calibration input w_q, $\|w_q\| \le \bar{w}$, the same arguments hold after replacing ζ_{cal} by $\zeta_0 + \{c_D(G+W) + \bar{w}\}/(1-\rho)$. PE–RCR itself neither requires nor silently introduces such an input: throughout the main theorem, $\bar{w} = 0$ and calibration uses only methods from the given portfolio.

A.2 Estimation, scores, and the three learned rules

Define $\Theta = [H\ b] \in \mathbb{R}^{d \times (d+1)}$, so that $g_q = \Theta\phi_q + \xi_q$ is a linear regression model. Fix $\lambda > 0$. After the m-th calibration gradient, we form the projected ridge estimate

$$V_m = \lambda I_{d+1} + \sum_{q<m} \phi_q \phi_q^\top, \tag{23}$$

$$\bar{\Theta}_m = \left(\sum_{q<m} g_q \phi_q^\top \right) V_m^{-1}, \tag{24}$$

$$(\widehat{H}, \widehat{b}) = \operatorname*{arg\,min}_{(H', b') \in \mathfrak{T}} \|[H'\ b'] - \bar{\Theta}_m\|_F. \tag{25}$$

We write $\widehat{\Theta}_m = \widehat{\Theta} = [\widehat{H}\ \widehat{b}]$. The last step is a Euclidean projection onto a convex set. For the base theorem with known covariance, we set $\theta = (H, b, \Sigma)$ and $\widehat{\theta} = (\widehat{H}, \widehat{b}, \Sigma)$. The residual-covariance extension replaces the last component of $\widehat{\theta}$ by $\widehat{\Sigma}$. When Σ is fixed and known, we abbreviate the superscripts (H, b, Σ) to (H, b). All values J below are centered, that is, computed for $f - c$, so the same additive constant is set to zero in both the true and the fitted scores. This convention changes neither any argmin nor the pseudo-regret above. For the known row-norm bound on the true parameter matrix used below, one may take $\bar{G} = \sqrt{L^2 + G^2}$, because every row of the Hessian has norm at most L.

For a model $\theta = (H, b, \Sigma)$, define $F_i(H) = A_i - D_i H C$. Starting from $y \in \mathbb{R}^p$, the expected value of committing to i for the next h gradients, none of which has been observed yet, follows from the

mean and covariance recursion

$$\bar{z}^{\theta}_{i,0}(y) = y, \qquad S^{\theta}_{i,0} = 0, \tag{26}$$

$$\bar{z}^{\theta}_{i,k+1}(y) = F_i(H)\bar{z}^{\theta}_{i,k}(y) - D_i b, \tag{27}$$

$$S^{\theta}_{i,k+1} = F_i(H) S^{\theta}_{i,k} F_i(H)^{\top} + D_i \Sigma D_i^{\top}, \tag{28}$$

$$J^{\theta}_{i,h}(y) = \frac{1}{2}(C\bar{z}^{\theta}_{i,h})^{\top} H (C\bar{z}^{\theta}_{i,h}) + b^{\top} C \bar{z}^{\theta}_{i,h} + \frac{1}{2}\operatorname{tr}\big(HCS^{\theta}_{i,h}C^{\top}\big). \tag{29}$$

Only the first and second conditional moments of the noise enter this recursion.

Lemma 1 (Exact value of a fixed continuation)**.** *Under (A1), Eq.* (29) *equals the conditional expected centered terminal objective of h consecutive applications of method i from state y. The conclusion needs neither Gaussian noise nor temporal independence beyond the displayed conditional first and second moments. In particular, the conditional value is history invariant: any two prior histories ending at the same full state y have the same $J^{H,b}_{i,h}(y)$.*

Proof. For the fixed method, we have

$$Z_{k+1} = F_i(H) Z_k - D_i b - D_i \xi_k.$$

Since the noise has conditional mean zero, this gives the stated mean recursion. If $\widetilde{Z}_k = Z_k - \bar{z}_{i,k}$, then

$$\widetilde{Z}_{k+1} = F_i(H)\widetilde{Z}_k - D_i \xi_k.$$

The cross terms have zero conditional expectation because $\widetilde{Z}_k$ and the fixed action are measurable with respect to the history before ξ_k. Hence the constant conditional covariance in (A1) gives the covariance recursion in Eq. (29). Finally,

$$\mathbb{E}[(CZ_h)^{\top} H (CZ_h)] = (C\bar{z}_{i,h})^{\top} H (C\bar{z}_{i,h}) + \operatorname{tr}(HCS_{i,h}C^{\top}),$$

and the linear term averages to $b^{\top} C \bar{z}_{i,h}$, proving the formula. □

Consider a decision point (the start of a block) at which $r \geq 1$ steps remain, the state is z, and the common gradient g has just been observed. The exact and plug-in *post-response commit scores* are

$$Q^{\theta}_{i,r}(z,g) = J^{\theta}_{i,r-1}(A_i z - D_i g), \qquad \widehat{Q}_{i,r}(z,g) = J^{\widehat{\theta}}_{i,r-1}(A_i z - D_i g). \tag{30}$$

Both scores hold the observed g fixed, including its realized noise, which both influences the selection and drives the selected update. Hence they are not unconditional open-loop scores into which a data-dependent action is inserted afterward. All finite argmins use a fixed tie-breaking rule.

PE–Commit baseline (predictable one-shot commit). After the m calibration steps shared by all three rules, let $R = B - m$ and choose

$$\widehat{i}_{\mathrm{commit}} \in \operatorname*{arg\,min}_{i \in [K]} J^{\widehat{\theta}}_{i,R}(z_m) \tag{31}$$

using the $\mathcal{F}_m$-measurable state and fit. This choice is made before g_m is observed. PE–Commit then applies $\widehat{i}_{\mathrm{commit}}$ for all remaining R updates. This baseline pays exactly the same calibration (opportunity) cost as PE–RCR and differs only in its routing rule.

PE–Post rule (post-response one-shot commit). After the same calibration phase and fit, PE–Post observes the next gradient g_m at the state z_m, chooses

$$\widehat{i}_{\mathrm{post}}(g_m) \in \operatorname*{arg\,min}_{i \in [K]} \widehat{Q}_{i,R}(z_m, g_m), \qquad R = B - m, \tag{32}$$

and applies $\widehat{i}_{\mathrm{post}}(g_m)$ for all R remaining updates, including the first update, which is driven by g_m. The fitted model and z_m are $\mathcal{F}_m$-measurable, whereas the choice is $\mathcal{F}_m \vee \sigma(g_m)$-measurable and hence $\mathcal{F}_m^{+}$-measurable. There is no later rerouting and no additional oracle call. Thus PE–Post differs from PE–Commit only in that it uses g_m, and from PE–RCR in that it commits after this single decision point.

PE–RCR algorithm (persistently excited receding-commit routing). Fix $m < B$, typically $m = \lfloor B/2 \rfloor$.

1. *Calibrate and fit.* Make exactly m oracle calls and m updates of the actual state with the calibration policy in Eq. (19). Fit $(\widehat{H}, \widehat{b})$ by Eq. (25).
2. *Route.* Set $r = B - m$. At the start of a block, query the oracle once at the current state, observe g, and choose
$$\widehat{i} \in \arg\min_{i \in [K]} \widehat{Q}_{i,r}(z, g).$$
Apply this optimizer for $\ell(r) = \max\{1, \lfloor r/2 \rfloor\}$ consecutive updates, including the update driven by the already observed g. The later gradients in the block have not been observed yet, but the action for them is already fixed.
3. *Repeat.* Replace r by $r - \ell(r)$ and start a new block unless $r = 0$.

For $R = B - m$, the number N_R of post-response routing decisions satisfies

$$N_R \le \lceil \log_2 R \rceil + 1. \tag{33}$$

Indeed, a remaining budget $r > 1$ becomes $r - \lfloor r/2 \rfloor = \lceil r/2 \rceil$, and the final block, with $r = 1$, ends the run. Hence there are at most $\lceil \log_2 R \rceil$ halvings plus one final block. Every routing score is computed after the gradient at the same decision point is observed and before this gradient is used in the update. No extra oracle call is hidden in the score.

A.2.1 Computational cost

Write $\mathrm{time}_\Pi(d, p, K)$ for the arithmetic cost and $\mathrm{mem}_\Pi(d, p, K)$ for the workspace of one projection onto $\mathfrak{T}$. Accumulating V_m and $\sum_{q<m} g_q \phi_q^\top$ in a single pass costs $O(md^2)$ arithmetic and $O(d^2)$ memory; factoring the $(d+1)$-dimensional ridge system, solving it for its d right-hand sides, and checking PE cost $O(d^3)$. Thus regression and projection require

$$O(md^2 + d^3 + \mathrm{time}_\Pi) \quad \text{arithmetic}, \qquad O(d^2 + \mathrm{mem}_\Pi) \quad \text{workspace}. \tag{34}$$

If Σ is unknown, we also accumulate $\sum_{q<m} g_q g_q^\top$; after fitting, the residual covariance can then be computed from these three second-order sufficient statistics, with the same $O(md^2)$ order and without storing the trajectory. We leave the projection cost as a parameter because the general spectral stability constraints may require a conic solver. Constructing the fitted matrices F_i and $D_i \Sigma D_i^\top$ for all candidates costs $O(d^2 p + K(pd^2 + p^2 d))$ dense operations.

For one candidate, one step of the mean–covariance recursion costs $O(p^3)$, dominated by $F_i S F_i^\top$, and needs $O(p^2)$ working memory. Hence a direct score with horizon h costs $O(hp^3)$. If $r_0 = R$ and $r_{j+1} = \lceil r_j/2 \rceil$, then $\sum_j (r_j - 1) = O(R)$, so direct scoring at every decision point costs

$$O(KRp^3) \quad \text{arithmetic} \tag{35}$$

and can process the candidates one at a time with $O(p^2 + K)$ extra workspace.

An exact acceleration by repeated doubling is also possible. For a fixed candidate, we summarize h future steps by $(\mathsf{D}_h, \mathsf{u}_h, \mathsf{V}_h)$, where the resulting mean and covariance are $\mathsf{D}_h y + \mathsf{u}_h$ and V_h. The one-step summary is $(F_i, -D_i b, D_i \Sigma D_i^\top)$, and summaries compose as

$$(\mathsf{D}_2, \mathsf{u}_2, \mathsf{V}_2) \circ (\mathsf{D}_1, \mathsf{u}_1, \mathsf{V}_1) = \big(\mathsf{D}_2 \mathsf{D}_1,\ \mathsf{D}_2 \mathsf{u}_1 + \mathsf{u}_2,\ \mathsf{D}_2 \mathsf{V}_1 \mathsf{D}_2^\top + \mathsf{V}_2\big). \tag{36}$$

Let $h_j = r_j - 1$ be the continuation horizon needed at decision point j. Because $r_{j+1} = \lceil r_j/2 \rceil$,

$$h_{j+1} = \left\lfloor \frac{h_j}{2} \right\rfloor. \tag{37}$$

Starting from the zero-step (identity) summary, we build the needed summaries in reverse order. From the cached h_{j+1}-step summary, one self-composition gives the $2h_{j+1}$-step summary and, when h_j is odd, one further composition with the one-step summary gives horizon h_j. Thus all needed horizons require only $O(\log(1 + R))$ compositions per candidate, instead of a fresh binary expansion

at every decision point. We cache each summary together with the scalar contribution of its terminal covariance to the score. With this exact acceleration, the cost of PE–RCR is

$$\begin{aligned} \text{time} &= O(Kp^3 \log(1+R)) + O(Kp^2 \log(1+R)), \\ \text{cache} &= O(Kp^2 \log(1+R)). \end{aligned} \tag{38}$$

Updates of the actual state add $O(B(p^2+pd))$ arithmetic. Neither implementation calls the oracle while scoring, so the oracle complexity is exactly B gradients and B applied updates. These workspace bounds exclude the storage of the fixed, known portfolio, which is $O(K(p^2+pd)+dp)$ in dense form.

PE–Commit and PE–Post each need only one horizon, namely R and $R-1$, respectively. They can therefore process the candidates one at a time and evaluate the exact dyadic compositions in $O(Kp^3 \log(1+R))$ arithmetic with $O(p^2+K)$ extra workspace (or use the direct $O(KRp^3)$ recursion). They share the costs above of fitting, constructing the transitions, updating the actual state, storing the portfolio, and calling the oracle. Accordingly, all three accelerated rules need the same asymptotic time to construct the summaries. Each one-shot rule makes only K score evaluations and processes candidates in a streaming fashion, whereas PE–RCR makes $KN_R = O(K \log(1+R))$ evaluations and caches the full chain of needed horizons. PE–Post's selection uses the already counted gradient g_m, so it does not change the budget of B oracle calls and B updates.

A.3 Confidence and score sensitivity

Lemma 2 (Ridge confidence bound on one trajectory)**.** *Suppose that every row of the true Θ has Euclidean norm at most $\bar{G}$. For $0 < \delta < 1$, define*

$$\Phi = \sqrt{X^2+1}, \tag{39}$$

$$\beta_m(\delta) = \sigma\sqrt{2\log\frac{d}{\delta} + (d+1)\log\left(1 + \frac{m\Phi^2}{\lambda(d+1)}\right)} + \sqrt{\lambda}\bar{G}, \tag{40}$$

$$e_m(\delta) = \frac{\sqrt{d}\,\beta_m(\delta)}{\sqrt{\lambda+\kappa m}}. \tag{41}$$

Under (A1)–(A3), with probability at least $1-\delta$,

$$\|[\widehat{H} - H \;\; \widehat{b} - b]\|_F = \|\widehat{\Theta}_m - \Theta\|_F \le e_m(\delta), \quad \textit{and hence} \quad \|\widehat{H} - H\|_{\mathrm{op}},\ \|\widehat{b} - b\|_2 \le e_m(\delta). \tag{42}$$

This event is $\mathcal{F}_m$-measurable.

Proof. Fix an output coordinate j. The self-normalized inequality for vector-valued martingales with conditionally sub-Gaussian noise (Abbasi-Yadkori et al., 2011, Theorem 3), followed by a union bound over $j \in [d]$, gives

$$\|\bar{\Theta}_{m,j:} - \Theta_{j:}\|_{V_m} \le \sigma\sqrt{2\log\left(\frac{d}{\delta}\frac{\det(V_m)^{1/2}}{\lambda^{(d+1)/2}}\right)} + \sqrt{\lambda}\bar{G}.$$

The ridge term comes from the identity $\bar{\Theta}_m - \Theta = (\sum_{q<m} \xi_q \phi_q^\top - \lambda\Theta)V_m^{-1}$. Since $\|\phi_q\| \le \Phi$, the arithmetic–geometric mean inequality gives

$$\frac{\det V_m}{\lambda^{d+1}} \le \left(1 + \frac{m\Phi^2}{\lambda(d+1)}\right)^{d+1}.$$

The PE condition in Eq. (20) gives $\lambda_{\min}(V_m) \ge \lambda + \kappa m$. Converting each row's self-normalized norm into the Euclidean norm and summing the squared row bounds yields $\|\bar{\Theta}_m - \Theta\|_F \le e_m(\delta)$. The Euclidean projection onto a closed convex set containing Θ cannot increase the Frobenius distance to Θ; this proves the first inequality in Eq. (42). Finally, both the operator norm of the Hessian block and the Euclidean norm of the last column are at most the Frobenius norm of the full error matrix. □

The next lemma is the uniform perturbation bound needed by the router: it controls how much the commit scores change when the model is perturbed. Its constants do not grow with the commit horizon, because every admissible closed-loop map is uniformly contractive. Define

$$\bar{\zeta} = \zeta + \frac{c_D G}{1-\rho}, \qquad \mathcal{V} = \frac{c_D^2 \operatorname{tr}(\Sigma)}{1-\rho^2}, \tag{43}$$

$$\Gamma_H = \frac{c_D c_C^2 \bar{\zeta}(L c_C \bar{\zeta} + G)}{1-\rho} + \frac{c_C^2}{2}(\bar{\zeta}^2 + \mathcal{V}) + \frac{L\rho c_D c_C^3 \mathcal{V}}{1-\rho^2}, \tag{44}$$

$$\Gamma_b = \frac{c_D c_C (L c_C \bar{\zeta} + G)}{1-\rho} + c_C \bar{\zeta}, \qquad \Gamma = \Gamma_H + \Gamma_b. \tag{45}$$

Lemma 3 (Uniform sensitivity of fixed and post-response commit scores)**.** *Let $\theta = (H, b)$ and $\theta' = (H', b')$ both belong to $\mathfrak{T}$, with the same Σ, and suppose $\|H - H'\|_{\mathrm{op}} \le e_H$ and $\|b - b'\|_2 \le e_b$. For every method i, every $h \ge 0$, and every $\|y\| \le \zeta$,*

$$|J_{i,h}^{\theta}(y) - J_{i,h}^{\theta'}(y)| \le \Gamma_H e_H + \Gamma_b e_b. \tag{46}$$

Consequently, for every $r \ge 1$ and every observed pair (z, g) whose actual next state $y = A_i z - D_i g$ satisfies $\|y\| \le \zeta$,

$$|Q_{i,r}^{\theta}(z, g) - Q_{i,r}^{\theta'}(z, g)| \le \Gamma_H e_H + \Gamma_b e_b. \tag{47}$$

In particular, on the event in Lemma 2, all scores used by PE–Post and PE–RCR have error at most

$$\varepsilon_m(\delta) = \Gamma e_m(\delta). \tag{48}$$

Proof. Fix i, write $F = A_i - D_i H C$ and $F' = A_i - D_i H' C$, and start both fixed continuations from the same state $y = A_i z - D_i g$. Their means and covariances follow the recursion in Eq. (29). By uniform contraction, for all k,

$$\|\bar{z}_k\|, \|\bar{z}'_k\| \le \bar{\zeta}, \qquad \operatorname{tr}(S_k), \operatorname{tr}(S'_k) \le \mathcal{V}.$$

Let $\chi_k = \|\bar{z}_k - \bar{z}'_k\|$. Since $\|F - F'\| \le c_D c_C e_H$,

$$\chi_{k+1} \le \rho \chi_k + c_D c_C \bar{\zeta} e_H + c_D e_b, \qquad \chi_0 = 0,$$

and hence

$$\chi_k \le \frac{c_D(c_C \bar{\zeta} e_H + e_b)}{1-\rho}. \tag{49}$$

For $\chi_k^S = \|S_k - S'_k\|_*$, we add and subtract $F S'_k F^\top$. Since the covariance term $D_i \Sigma D_i^\top$ of the future noise is the same in both models,

$$\chi_{k+1}^S \le \rho^2 \chi_k^S + 2\rho c_D c_C \mathcal{V} e_H, \qquad \chi_0^S = 0.$$

Therefore

$$\chi_k^S \le \frac{2\rho c_D c_C \mathcal{V}}{1-\rho^2} e_H. \tag{50}$$

For the terminal mean terms, we add and subtract the value at the primed mean under the unprimed H, b; the Cauchy–Schwarz inequality then gives

$$\left| \frac{1}{2}(C\bar{z})^\top H (C\bar{z}) + b^\top C \bar{z} - \frac{1}{2}(C\bar{z}')^\top H' (C\bar{z}') - (b')^\top C \bar{z}' \right|$$
$$\le (L c_C \bar{\zeta} + G) c_C \chi_k + \frac{1}{2} c_C^2 \bar{\zeta}^2 e_H + c_C \bar{\zeta} e_b.$$

For the covariance terms,

$$\frac{1}{2} \left| \operatorname{tr}(H C S_k C^\top) - \operatorname{tr}(H' C S'_k C^\top) \right| \le \frac{1}{2} L c_C^2 \chi_k^S + \frac{1}{2} c_C^2 \mathcal{V} e_H.$$

Substituting Eqs. (49)–(50) and collecting the coefficients of e_H, e_b gives Eq. (46) with the constants in Eq. (45). Taking $y = A_i z - D_i g$ and $h = r - 1$ gives Eq. (47). The argument includes $h = 0$, so it also covers a block that consists of a single step. The observed current gradient is never averaged over or reused in this perturbation comparison. □

A.4 POTENTIAL DECREASE AND PSEUDO-REGRET UNDER THE FULL BUDGET

We define the best commit potential

$$M_r^\theta(z) = \min_{i\in[K]} J_{i,r}^\theta(z), \qquad M_0^\theta(z) = f_{H,b}(Cz) - c. \tag{51}$$

Note that this is not the unrestricted Bellman value; it is exactly the quantity needed to telescope over the receding-commit decisions.

Lemma 4 (One receding-commit block)**.** *At a block start, let $r \geq 1$ steps remain and let the block length be $1 \leq \ell \leq r$. Suppose that all plug-in commit scores have absolute error at most ε. If the router observes g, selects a plug-in score minimizer $\widehat{i}$, and then uses $\widehat{i}$ for ℓ updates, then*

$$\mathbb{E}\big[M_{r-\ell}^\theta(Z_\ell) \mid Z_0 = z\big] \leq M_r^\theta(z) + 2\varepsilon. \tag{52}$$

The inequality holds conditionally on any complete history up to the block start that ends in state z; the expectation is over the current gradient and the $\ell - 1$ subsequent gradients.

Proof. We first condition on the observed current gradient g. If method i is used for the block, then

$$M_{r-\ell}^\theta(Z_\ell) \leq J_{i,r-\ell}^\theta(Z_\ell).$$

The tower property and the fixed conditional first and second moments in (A1) give

$$\mathbb{E}\big[J_{i,r-\ell}^\theta(Z_\ell) \mid Z_0 = z, g_0 = g\big] = J_{i,r-1}^\theta(A_i z - D_i g) = Q_{i,r}^\theta(z, g). \tag{53}$$

Indeed, the left side applies i for the remaining $\ell - 1$ updates of the actual block and for the hypothetical commit continuation of $r - \ell$ updates, hence to all $r - 1$ future gradients (not yet observed) after the fixed first update.

Uniform score error and plug-in minimization imply, pointwise in g,

$$Q_{\widehat{i},r}^\theta(z, g) \leq \min_i Q_{i,r}^\theta(z, g) + 2\varepsilon.$$

Finally, we average over the current common gradient. Since the expectation of a minimum is at most the minimum of the expectations,

$$\mathbb{E}_g \min_i Q_{i,r}^\theta(z, g) \leq \min_i \mathbb{E}_g Q_{i,r}^\theta(z, g) = \min_i J_{i,r}^\theta(z) = M_r^\theta(z).$$

Combining these displays proves Eq. (52). Note that the same current noise may both determine the action and drive its update: we condition on it exactly and average only over the subsequent noise. □

Lemma 5 (Fixed continuations forget their initial state)**.** *For any i, $R \geq 0$, and $\|z\|, \|z'\| \leq \zeta$,*

$$\left|J_{i,R}^{H,b}(z) - J_{i,R}^{H,b}(z')\right| \leq (LX + G)c_C\rho^R\|z - z'\|. \tag{54}$$

Proof. For a fixed method and a fixed parameter, the two terminal means satisfy

$$\bar{z}_{i,R}(z) - \bar{z}_{i,R}(z') = F_i(H)^R(z - z'),$$

while their covariance matrices coincide, because the covariance recursion starts from zero and does not depend on the deterministic initial state. Hence the covariance terms in Eq. (29) cancel exactly. The means of both terminal query points lie in the ball of radius X, on which $x \mapsto f_{H,b}(x) - c$ is $(LX + G)$-Lipschitz. Moreover,

$$\|CF_i(H)^R(z - z')\| \leq c_C\rho^R\|z - z'\|.$$

Combining these facts proves Eq. (54). The argument uses no coupling of the future noise and no assumption on its support. □

Theorem 2 (Predictable one-shot plug-in commit)**.** *Suppose that (A1)–(A3) hold, let $R = B - m$, and run PE–Commit as in Eq.* (31)*. Denote its pseudo-regret under the full budget by* $\mathrm{Reg}_B^{\mathrm{commit}}$*. For every $0 < \delta < 1$,*

$$\boxed{\mathrm{Reg}_B^{\mathrm{commit}} \le [\mathrm{Reg}_B^{\mathrm{commit}}]_+ \le (LX + G)c_C(\zeta_{\mathrm{cal}} + \zeta_{\mathrm{fix}})\rho^R + 2\Gamma e_m(\delta) + 2\Lambda\delta.} \tag{55}$$

The choice is predictable with respect to the gradient g_m, and the complete procedure still uses B gradients and B updates. If instead, for a given $\delta_{\mathrm{PE}} \in [0,1]$, $\Pr\{\mathcal{P}_m(\kappa)\} \ge 1 - \delta_{\mathrm{PE}}$ at the known κ, and a failed check applies the prespecified stable portfolio method i_{fb} for every remaining update, then the right side gains the deterministic term $2\Lambda\delta_{\mathrm{PE}}$.

Proof. Let $\mathcal{E}_m \in \mathcal{F}_m$ be the confidence event from Lemma 2. Conditionally on $\mathcal{F}_m$, the remaining expected terminal value of PE–Commit is $J^{\theta}_{\widehat{i}_{\mathrm{commit}},R}(z_m)$. On $\mathcal{E}_m$, Lemma 3 and plug-in minimization give

$$J^{\theta}_{\widehat{i}_{\mathrm{commit}},R}(z_m) \le J^{\widehat{\theta}}_{\widehat{i}_{\mathrm{commit}},R}(z_m) + \Gamma e_m(\delta) \le \min_i J^{\theta}_{i,R}(z_m) + 2\Gamma e_m(\delta) = M^{\theta}_R(z_m) + 2\Gamma e_m(\delta).$$

On $\mathcal{E}_m^c$, both terms lie in $[-\Lambda, \Lambda]$. Taking expectations therefore yields

$$\mathbb{E}[f_{H,b}(Cz_B^{\mathrm{commit}}) - c] \le \mathbb{E}[M^{\theta}_R(z_m)] + 2\Gamma e_m(\delta) + 2\Lambda\delta.$$

For $i_B^\star \in \arg\min_i J^{H,b}_{i,B}(z_0)$, we have pointwise $M^{\theta}_R(z_m) \le J^{H,b}_{i_B^\star,R}(z_m)$. We apply Lemma 5 to z_m and the state $Z_m^\star$ of the best fixed method after m updates, and then use $\|z_m\| \le \zeta_{\mathrm{cal}}$, $\|Z_m^\star\| \le \zeta_{\mathrm{fix}}$, and the tower property. This bounds the remaining difference to $J^{H,b}_{i_B^\star,B}(z_0)$ by the calibration (opportunity) cost displayed in Eq. (55). Subtracting the value of the best fixed method proves the claim. Intersecting with a random PE event gives the additional deterministic term by the same 2Λ range argument. □

Theorem 3 (Fitted post-response one-shot commit)**.** *Suppose that (A1)–(A3) hold, let $R = B - m$, and run PE–Post as in Eq.* (32)*. Denote its pseudo-regret under the full budget by* $\mathrm{Reg}_B^{\mathrm{post}}$*. For every $0 < \delta < 1$,*

$$\boxed{\mathrm{Reg}_B^{\mathrm{post}} \le [\mathrm{Reg}_B^{\mathrm{post}}]_+ \le (LX + G)c_C(\zeta_{\mathrm{cal}} + \zeta_{\mathrm{fix}})\rho^R + 2\Gamma e_m(\delta) + 2\Lambda\delta.} \tag{56}$$

The selection uses g_m, the same gradient that drives its first applied update, and the complete procedure uses exactly B gradients and B updates. If instead, for a given $\delta_{\mathrm{PE}} \in [0,1]$, $\Pr\{\mathcal{P}_m(\kappa)\} \ge 1 - \delta_{\mathrm{PE}}$ at the known κ, and a failed check applies the prespecified stable portfolio method i_{fb} for every remaining update, then the right side gains the deterministic term $2\Lambda\delta_{\mathrm{PE}}$.

Proof. Let $\mathcal{E}_m \in \mathcal{F}_m$ be the confidence event from Lemma 2. On every complete calibration history in $\mathcal{E}_m$, all post-response scores in Eq. (32) have error at most $\Gamma e_m(\delta)$, uniformly over the newly observed gradient g_m and over all candidates. We apply Lemma 4 once with $r = \ell = R$. Since $M_0^\theta(z) = f_{H,b}(Cz) - c$, it gives

$$\mathbb{E}[f_{H,b}(Cz_B^{\mathrm{post}}) - c \mid \mathcal{F}_m] \le M^{\theta}_R(z_m) + 2\Gamma e_m(\delta) \qquad \text{on } \mathcal{E}_m.$$

On $\mathcal{E}_m^c$, the terminal and commit values lie in $[-\Lambda, \Lambda]$. Taking expectations therefore adds only $2\Lambda\delta$. Fix $i_B^\star \in \arg\min_i J^{H,b}_{i,B}(z_0)$, and let $Z_m^\star$ be its state after m updates. Pointwise, $M^{\theta}_R(z_m) \le J^{H,b}_{i_B^\star,R}(z_m)$, while history invariance and the tower property give $J^{H,b}_{i_B^\star,B}(z_0) = \mathbb{E}[J^{H,b}_{i_B^\star,R}(Z_m^\star)]$. Lemma 5, together with $\|z_m\| \le \zeta_{\mathrm{cal}}$ and $\|Z_m^\star\| \le \zeta_{\mathrm{fix}}$, therefore bounds their expected difference by $(LX + G)c_C(\zeta_{\mathrm{cal}} + \zeta_{\mathrm{fix}})\rho^R$. It remains to subtract the value of the best fixed method under the full budget. Intersecting with a random PE event adds $2\Lambda\delta_{\mathrm{PE}}$ by the same range argument. □

Theorem 4 (End-to-end learned routing under one budget)**.** *Suppose that (A1)–(A3) hold and that PE–RCR is run with $m < B$ calibration updates. Let $R = B - m$, let N_R be bounded as in Eq.* (33)*, and let $i_B^\star \in \arg\min_i J^{H,b}_{i,B}(z_0)$. Then, for every $0 < \delta < 1$,*

$$\boxed{\mathrm{Reg}_B \le \big[\mathrm{Reg}_B\big]_+ \le (LX + G)c_C(\zeta_{\mathrm{cal}} + \zeta_{\mathrm{fix}})\rho^R + 2N_R\Gamma e_m(\delta) + 2\Lambda\delta.} \tag{57}$$

All terms refer to the common oracle budget B. In particular, calibration is not compared with a fixed method run for only $B - m$ updates. At every decision point, the realized noise of the observed gradient, on which the selection depends, is validly reused in the chosen update, because the router estimates its conditional commit value, not an unconditional schedule value. The displayed inequality assumes almost-sure PE, $\Pr\{\mathcal{P}_m(\kappa)\} = 1$. If instead, for a given $\delta_{\mathrm{PE}} \in [0,1]$, $\Pr\{\mathcal{P}_m(\kappa)\} \geq 1 - \delta_{\mathrm{PE}}$ at the known κ, and a failed check applies the prespecified stable portfolio method i_{fb} for every remaining update, then the right side gains the deterministic term $2\Lambda\delta_{\mathrm{PE}}$.

Proof. Let $\mathcal{E}_m \in \mathcal{F}_m$ be the confidence event in Lemma 2. On $\mathcal{E}_m$, we apply Lemmas 3 and 4 to each of the N_R blocks; telescoping yields, conditionally on the complete calibration history,

$$\mathbb{E}[f_{H,b}(Cz_B^{\mathrm{route}}) - c \mid \mathcal{F}_m] \leq M_R^{H,b}(z_m) + 2N_R\Gamma e_m(\delta). \tag{58}$$

On $\mathcal{E}_m^c$, both the terminal value of the routed trajectory and every commit value, after removal of the common constant c, lie in $[-\Lambda, \Lambda]$. Therefore

$$\mathbb{E}[f_{H,b}(Cz_B^{\mathrm{route}}) - c] \leq \mathbb{E}[M_R^{H,b}(z_m)] + 2N_R\Gamma e_m(\delta) + 2\Lambda\delta. \tag{59}$$

Pointwise, $M_R^{H,b}(z_m) \leq J_{i_B^\star,R}^{H,b}(z_m)$. Let $Z_m^\star$ denote the state after the first m updates along the path of the best fixed method $i_B^\star$. History invariance and the tower property give

$$J_{i_B^\star,B}^{H,b}(z_0) = \mathbb{E}[J_{i_B^\star,R}^{H,b}(Z_m^\star)].$$

We place z_m and $Z_m^\star$ on any product probability space. Lemma 5, followed by the pathwise bounds $\|z_m\| \leq \zeta_{\mathrm{cal}}$ and $\|Z_m^\star\| \leq \zeta_{\mathrm{fix}}$, gives

$$\mathbb{E}[J_{i_B^\star,R}^{H,b}(z_m)] - J_{i_B^\star,B}^{H,b}(z_0) \leq (LX + G)c_C(\zeta_{\mathrm{cal}} + \zeta_{\mathrm{fix}})\rho^R. \tag{60}$$

Combining Eqs. (59)–(60) with the definition of $i_B^\star$ proves the displayed upper bound on the signed pseudo-regret. Its right side is nonnegative, so it also bounds $[\mathrm{Reg}_B]_+$; this proves the chain of inequalities in Eq. (57). □

Corollary 1 (Finite-sample transfer of an exact routing advantage)**.** *Let $\mathcal{P} = \mathcal{P}_m(\kappa)$ be the event that the PE check passes, and couple the learned PE–RCR and PE–Post policies to exact-model counterparts that use the same calibration, noise, block schedule, tie-breaking rule, and fallback. On $\mathcal{P}$, the exact counterparts replace only $\widehat{\theta}$ by θ; on $\mathcal{P}^c$, all four policies use the same fallback. Along the exact PE–RCR path, let $\mathcal{C}$ be the event that every candidate score at every decision point is within $\varepsilon = \Gamma e_m(\delta)$ of its true score, and let $\mathcal{M}$ be the event that every true minimizer is unique and its gap to every other candidate exceeds 2ε. Suppose*

$$\Pr(\mathcal{P} \cap \mathcal{C}^c) \leq \delta, \qquad \Pr(\mathcal{P} \cap \mathcal{M}^c) \leq \delta_{\mathrm{M}}.$$

No independence is required. If

$$\Delta_B = \mathbb{E}\big[f(CZ_B^{\mathrm{post},\star}) - f(CZ_B^{\mathrm{rcr},\star})\big],$$

then

$$\mathbb{E}\big[f(Cz_B^{\mathrm{rcr}}) - f(Cz_B^{\mathrm{post}})\big] \leq -\Delta_B + 4\Lambda(\delta + \delta_{\mathrm{M}}). \tag{61}$$

Thus $\Delta_B > 4\Lambda(\delta + \delta_{\mathrm{M}})$ suffices for a strict advantage of learned PE–RCR over learned PE–Post. If instead the exact counterparts always route, rather than using the same fallback, and $\Pr(\mathcal{P}^c) \leq \delta_{\mathrm{PE}}$, then the right side becomes $-\Delta_B + 2\Lambda\delta_{\mathrm{PE}} + 4\Lambda(\delta + \delta_{\mathrm{M}})$, where Δ_B is defined for these always-routing exact policies.

Proof. If true scores Q_i, fitted scores $\widehat{Q}_i$, and a true minimizer $i^\star$ satisfy $|Q_i - \widehat{Q}_i| \leq \varepsilon$ and $Q_j - Q_{i^\star} > 2\varepsilon$ for every $j \neq i^\star$, then minimizing the fitted scores forces $\widehat{i} = i^\star$. Applying this at the first decision point and coupling the same action, state, gradient, and block noises gives equality at the next decision point; induction recovers the complete exact PE–RCR path. Since exact PE–Post and exact PE–RCR share the first score, the same argument also recovers PE–Post and its committed continuation.

Write Υ and $\Upsilon^\star$ for the learned and exact differences of terminal values, PE–RCR minus PE–Post, after subtracting the common objective constant. On $\mathcal{P}^c$, both differences are zero because all four

policies use the same fallback on PE failure; on $\mathcal{P} \cap \mathcal{C} \cap \mathcal{M}$, they are equal. Elsewhere, $|f - c| \le \Lambda$ implies $\Upsilon - \Upsilon^\star \le 4\Lambda$. The union bound proves Eq. (61). For always-routing exact policies, the learned difference is zero on $\mathcal{P}^c$, while the exact difference is at least -2Λ; this gives the stated $2\Lambda\delta_{\mathrm{PE}}$ term. The Lean verification checks the deterministic steps of this argument, namely the strict argmin, the path induction, and the finite-expectation assembly, from these explicit event hypotheses. □

Proposition 2 (Strict non-myopic value)**.** *Consider the known one-dimensional model*

$$f(x) = \tfrac{1}{2}x^2, \qquad x_0 = 1, \qquad x_{q+1} = x_q - \eta_{i_q}(x_q + \xi_q), \qquad \eta_{\mathsf{A}} = \tfrac{3}{2}, \quad \eta_{\mathsf{C}} = \tfrac{1}{2}.$$

Both closed-loop factors have modulus $1/2$*. Let* $B = 2$*, and let the noise be i.i.d. with*

$$\Pr(\xi_q = -4/9) = 3/5, \qquad \Pr(\xi_q = 2/3) = 2/5.$$

Exact receding commit with unit blocks is the unique minimizer among the 64 *outcome-distinct deterministic response-contingent plans in the reduced representation over reachable histories. The exact response-aware one-step greedy rule instead minimizes the immediate post-update loss at each decision point, and*

$$\underbrace{\mathbb{E}f(x_2^{\mathrm{rcr}})}_{143/4320} < \underbrace{\mathbb{E}f(x_2^{\mathrm{greedy}})}_{35/864}, \qquad \mathbb{E}f(x_2^{\mathrm{greedy}}) - \mathbb{E}f(x_2^{\mathrm{rcr}}) = \frac{1}{135}. \tag{62}$$

After the low first gradient, the conservative action is worse immediately by $20/81$*, but the aggressive action is worse by* $19/81$ *under the true two-update same-method commit score. Receding commit also beats every predictable policy and every complete two-action schedule fixed after only the first gradient; their minimum values are* $67/864$ *and* $35/864$*, respectively. The minimum value of same-method post-response commit is* $67/864$*. The smallest true gap between receding scores across both decision points is* $10/81$*; consequently, a uniform score radius below* $5/81$ *preserves every receding action. These conclusions also hold under arbitrary internal randomization that is independent of the oracle.*

Proof. The noise is centered and bounded, and the two possible gradient values are distinct at every state, so $g_q - x_q$ reveals the realized noise label. At the first decision point, direct substitution gives

	immediate loss		two-update commit score	
ξ_0	A	C	A	C
$-4/9$	$1/72$	$169/648$	$97/288$	$265/2592$
$2/3$	$9/8$	$1/72$	$59/96$	$35/864$

Thus, after the low gradient, one-step greedy uniquely chooses A, whereas receding commit uniquely chooses C; subtraction gives the two ranking gaps in the proposition. After the high gradient, both choose C.

Backward minimization over the second gradient gives the conditional terminal values and the optimal final actions

	$i_0 = \mathsf{A}$		$i_0 = \mathsf{C}$	
ξ_0	value	(i_1^-, i_1^+)	value	(i_1^-, i_1^+)
$-4/9$	$35/864$	(C, C)	$73/2592$	(A, C)
$2/3$	$83/864$	(C, A)	$35/864$	(C, C)

All comparisons are strict. Hence the unique contingent optimum chooses C after either first gradient and then chooses A only on the low–low history; averaging its branch values gives $(3/5)(73/2592) + (2/5)(35/864) = 143/4320$. One-step greedy chooses A, C after the low and high first gradients, respectively, and C on every second history, so its value is $35/864$. Direct subtraction over the four reachable final histories gives a minimum gap of $10/81$ at the last decision point; together with the minimum gap of $19/81$ at the first decision point, this proves the stated robustness of the receding actions to uniform score errors.

Exact enumeration of cases gives the minima of the remaining policy classes:

policy class	distinct plans	minimum	unique minimizing plan
both-response contingent	64	$143/4320$	$(\mathsf{C},\mathsf{C};\mathsf{A},\mathsf{C},\mathsf{C},\mathsf{C})$
predictable	8	$67/864$	$(\mathsf{C};\mathsf{C},\mathsf{C})$
schedule fixed after response 1	16	$35/864$	$(\mathsf{AC},\mathsf{CC})$
same-method post-response commit	4	$67/864$	(C,C)

Here, entries are listed in low-then-high gradient order. Any behavioral randomization is a distribution over complete deterministic plans sampled at the start; its value is a convex combination of plan values and cannot improve on a class minimum. This exact construction with a known model uses no calibration and does not change the positive-part learning rate. □

Remark 2 (Machine-checked verification). The Lean 4/mathlib project `proofs/lean_optimizer_routing/` (with pinned versions) verifies the following. For Proposition 2, it verifies the exact rational values and the strict reversal of the immediate-loss and commit-score rankings, derives both response-aware score rules, proves that the local argmins are unique, and identifies the global optimum and proves its uniqueness. It lower-bounds the values of all deterministic policies in four classes: 64 outcome-distinct response-contingent plans, 8 predictable plans, 16 schedules chosen after the first gradient, and 4 post-response commit plans. It also proves that the gradient reveals the realized noise and covers arbitrary real mixtures of complete plans. For Corollary 1, it checks the deterministic selection inequality with a margin of two score radii, pathwise recovery under uniform strict margins, and the transfer to finite expectations, including the separate coefficient of the shared fallback. Further modules verify that the signed bound implies the positive-part bound and prove that a finite-support mean-zero noise law with $\|\xi\|^2 \le W^2$ has $\mathrm{tr}(\Sigma) \le W^2$. Finally, the project checks the deterministic assembly of Theorems 2, 3, and 4 under the full budget from explicit hypotheses on the score error, block descent, calibration cost, confidence failure, and random PE, including the score-radius factor (one versus N_R) and the separate $2\Lambda\delta_{\mathrm{PE}}$ term. The project contains no `sorry`, `admit`, or custom axioms.

Concentration, the conditional block lemma with selected noise, persistent excitation, contraction of fixed continuations, the probabilities of the score-margin events, and the continuous stochastic model are proved by hand in this appendix; the Lean assembly takes them as explicit hypotheses. Thus the formal verification covers exact finite separations and the deterministic assembly of the theorems from explicit hypotheses, not the complete stochastic theorem end to end.

Remark 3 (Random PE event). Fix a known κ and a given $\delta_{\mathrm{PE}} \in [0,1]$. Suppose that $\Pr\{\mathcal{P}_m(\kappa)\} \ge 1-\delta_{\mathrm{PE}}$ instead of almost-sure PE, and that a failed check applies the prespecified stable portfolio method i_{fb} for every remaining update. Then the same proof applies on the intersection of the PE event and the ridge confidence event, and the right-hand sides of Eqs. (55), (56), and (57) gain the deterministic term $2\Lambda\delta_{\mathrm{PE}}$. Thus we never treat a realized Gram eigenvalue as a deterministic design guarantee. Under random PE, a vanishing positive part additionally requires $\delta_{\mathrm{PE}} = \delta_{\mathrm{PE}}(B) \to 0$.

Proposition 3 (Round-robin calibration yields random PE)**.** *During calibration, fix the predictable round-robin schedule over the portfolio, $E_q = 1 + (q \bmod K)$. Let $1 \le s < m$, $n = m - s$, and let $\kappa_0 > 0$ be a known constant. Suppose that for every $q = s, \dots, m-1$,*

$$\mathbb{E}[\phi_q \phi_q^\top \mid \mathcal{F}_{q-1}] \succeq \kappa_0 I_{d+1} \quad \text{a.s.} \tag{63}$$

Then

$$\Pr\left\{ \sum_{q<m} \phi_q \phi_q^\top \not\succeq \frac{\kappa_0 n}{2} I_{d+1} \right\} \le \delta_{\mathrm{PE}}^{\mathrm{rr}}(m) := (d+1)\exp\left\{ -\frac{\kappa_0^2 n}{32\Phi^4} \right\}. \tag{64}$$

Thus (A3) holds with $\kappa = \kappa_0(m-s)/(2m)$, except on this event. If $m \ge 2s$, one may use the fixed $\kappa = \kappa_0/4$. In particular, for fixed s and $m = \lfloor B/2 \rfloor$, $\delta_{\mathrm{PE}}^{\mathrm{rr}}(m)$ decays exponentially in B.

A sufficient condition for Eq. (63) *is the conditional small-ball condition: for known $\pi_0, \alpha > 0$, simultaneously every unit $u \in \mathbb{R}^{d+1}$ and every $q = s, \dots, m-1$ satisfy*

$$\Pr\{|u^\top \phi_q| \ge \alpha \mid \mathcal{F}_{q-1}\} \ge \pi_0 \quad \text{a.s.} \tag{65}$$

Then one may take $\kappa_0 = \pi_0 \alpha^2$. Here "simultaneously" means that one common probability-one event supports the inequality for every displayed q and every unit u, rather than a different null set

for each u. For the conditional-design conclusion, it is enough to impose the small-ball inequality on one common event for a fixed countable dense subset of the unit sphere: the induced quadratic-form bound extends to all unit directions by continuity.

Proof. Let $\mathsf{X}_q = \phi_q \phi_q^\top$ and $\overline{\mathsf{X}}_q = \mathbb{E}[\mathsf{X}_q \mid \mathcal{F}_{q-1}]$. Since $\|\phi_q\| \le \Phi$, both matrices lie between 0 and $\Phi^2 I$. Hence $Y_q = \mathsf{X}_q - \overline{\mathsf{X}}_q$ is a self-adjoint martingale difference with $\|Y_q\|_{\mathrm{op}} \le \Phi^2$ and $Y_q^2 \preceq \Phi^4 I$. The lower-tail form of the matrix Azuma inequality (Tropp, 2012, Theorem 7.1) therefore gives, for every $t > 0$,

$$\Pr\left\{\lambda_{\min}\left(\sum_{q=s}^{m-1} Y_q\right) \le -t\right\} \le (d+1)\exp\left\{-\frac{t^2}{8n\Phi^4}\right\}.$$

On the complementary event with $t = \kappa_0 n/2$, Eq. (63) implies

$$\sum_{q<m} \mathsf{X}_q \succeq \sum_{q=s}^{m-1} \mathsf{X}_q = \sum_{q=s}^{m-1} \overline{\mathsf{X}}_q + \sum_{q=s}^{m-1} Y_q \succeq \frac{\kappa_0 n}{2} I.$$

This proves Eq. (64) and the claims about κ. Finally, on the common event in Eq. (65), every unit vector u (or every one in the fixed dense subset) satisfies

$$u^\top \overline{\mathsf{X}}_q u = \mathbb{E}[(u^\top \phi_q)^2 \mid \mathcal{F}_{q-1}] \ge \pi_0 \alpha^2.$$

The quadratic form $u \mapsto u^\top \overline{\mathsf{X}}_q u$ is continuous. Hence the dense-subset formulation extends this inequality to the full unit sphere, which is equivalent to the conditional matrix lower bound on the same event. □

Remark 4 (Random PE is assumed, not obtained for free). Proposition 3 turns a model-specific conditional design or small-ball condition into an explicit and decaying δ_{PE}. Round robin alone implies neither condition. If all visited regressors miss a direction, Proposition 4 still rules out uniform recovery in that direction. The proposition thus gives concrete conditions for random PE without weakening this identifiability limit.

Corollary 2 (Explicit rate)**.** *Take $m = \lfloor B/2 \rfloor$, $B \ge 2(d+1)$, and $\delta = B^{-2}$. With all structural constants fixed and the PE constant $\kappa > 0$,*

$$\left[\mathrm{Reg}_B\right]_+ = O\left(\Gamma\sigma\sqrt{\frac{d\{(d+1)\log B + \log(dB)\}}{\kappa B}}\,\log B + \rho^{B/2} + B^{-2}\right), \tag{66}$$

with the additional ridge-bias contribution $O(\Gamma\sqrt{d\lambda}\bar{G}\log B/\sqrt{\kappa B})$. Thus the positive part of the learned router's pseudo-regret under the full budget vanishes. Equivalently, $\limsup_{B\to\infty} \mathrm{Reg}_B \le 0$; the theorem does not claim that a negative signed pseudo-regret converges to zero. The router makes $O(\log B)$ routing decisions on the single realized trajectory. Under Proposition 3 with fixed s, take $\kappa = \kappa_0/4$ for all sufficiently large B and add $2\Lambda\delta_{\mathrm{PE}}^{\mathrm{rr}}(m) = \exp\{-\Omega(B)\}$; the same asymptotic conclusion then holds without almost-sure PE.

Proof. Substitute Eq. (41) and $N_R \le \lceil \log_2 R \rceil + 1$ into Theorem 4. Under (A2), all state and sensitivity constants are independent of the horizon, and $R \ge B/2$. □

Corollary 3 (Sharper safety rate for the one-shot rules)**.** *Under the choices of Corollary 2, for each $a \in \{\mathrm{commit}, \mathrm{post}\}$,*

$$[\mathrm{Reg}_B^a]_+ = O\left(\Gamma\sigma\sqrt{\frac{d\{(d+1)\log B + \log(dB)\}}{\kappa B}} + \frac{\Gamma\sqrt{d\lambda}\bar{G}}{\sqrt{\kappa B}} + \rho^{B/2} + B^{-2}\right). \tag{67}$$

Under Proposition 3 with fixed burn-in, an additional $2\Lambda\delta_{\mathrm{PE}}^{\mathrm{rr}}(m) = \exp\{-\Omega(B)\}$ term appears. Thus both the predictable and post-response one-shot rules have a vanishing positive-part guarantee and avoid PE–RCR's $N_R = O(\log B)$ multiplier.

Proof. Substitute Eq. (41) into Theorems 2 and 3; unlike Theorem 4, each makes only one plug-in selection. □

Remark 5 (Numerical scale of the bound in the released $B = 320$ experiment). Consider the known constants of the single-task check of PE–RCR under random excitation (Appendix B.6), and take $m = R = 160$, $\delta = B^{-2}$, $\lambda = 10^{-6}$, and $\kappa = 4 \times 10^{-4}$. Direct substitution gives

$$\zeta = 6.7218833, \qquad \bar{\zeta} = 9.9361690, \qquad \Gamma = 1119.0914,$$
$$\beta_m = 1.1365223, \quad e_m = 6.3533029, \quad N_R = 9.$$

The calibration-cost, statistical, and ridge-confidence failure terms in Eq. (57) are, respectively,

$$0.00189318, \qquad 127978.67594224, \qquad 0.00106921,$$

so the right-hand side under almost-sure PE is 127978.67890462. This finite-budget value is valid but vacuous: a separate pathwise range argument already gives $[\mathrm{Reg}_{320}]_+ \le 2\Lambda = 109.48699821$. The large gap is due to the conservative uniform sensitivity constant and the small prespecified κ; we do not use the bound to interpret the empirical mean. Under random PE, Remark 3 adds the term $2\Lambda\delta_{\mathrm{PE}}$, which we do not report: observing no PE failures in a finite sample does not bound the probability δ_{PE}.

A.5 Extension to an unknown noise covariance

The scores require the covariance of the future noise. We show that it can be learned from the same calibration trajectory, without a frozen model copy. Define the residuals $\widehat{\xi}_q = g_q - \widehat{\Theta}\phi_q$ and the residual covariance estimator

$$\widehat{\Sigma}_m = \frac{1}{m} \sum_{q<m} \widehat{\xi}_q \widehat{\xi}_q^\top. \tag{68}$$

Corollary 4 (Unknown homoscedastic covariance)**.** *Use confidence level $\delta/2$ for both the parameter regression and the covariance concentration. Let $e = e_m(\delta/2)$, and define*

$$e_\Sigma = d \left[W^2 \sqrt{\frac{32 \log(4d/\delta)}{m}} + 2W\Phi e + \Phi^2 e^2 \right]. \tag{69}$$

With probability at least $1 - \delta$, $\|\widehat{\Sigma}_m - \Sigma\|_* \le e_\Sigma$. If the plug-in scores use $\widehat{\Sigma}_m$, Theorems 2, 3, and 4 remain true after replacing every occurrence of $\Gamma e_m(\delta)$ by

$$\Gamma_H e + \Gamma_b e + \Gamma_\Sigma e_\Sigma, \qquad \Gamma_\Sigma = \frac{L c_C^2 c_D^2}{2(1-\rho^2)}, \tag{70}$$

and replacing $\mathcal{V}$ in Γ_H by the computable bound

$$\mathcal{V}_{\mathrm{pub}} = \frac{c_D^2 (W^2 + e_\Sigma)}{1 - \rho^2}. \tag{71}$$

Indeed, $\mathrm{tr}\,\Sigma = \mathbb{E}[\|\xi_q\|^2 \mid \mathcal{F}_q] \le W^2$ *under (A1). The resulting positive-part bound still vanishes at a root-m rate up to dimension and logarithmic factors; the replacement appears once for PE–Commit and PE–Post, and at each decision point for PE–RCR.*

Proof. Let $a_q = (\Theta - \widehat{\Theta})\phi_q$. On the full Frobenius parameter event in Eq. (42), $\|a_q\| \le \|\Theta - \widehat{\Theta}\|_{\mathrm{op}} \|\phi_q\| \le \|\Theta - \widehat{\Theta}\|_F \Phi \le e\Phi$ and $\widehat{\xi}_q = \xi_q + a_q$. The self-adjoint martingale differences $Y_q = \xi_q \xi_q^\top - \Sigma$ satisfy $Y_q^2 \preceq 4W^4 I$. The matrix Azuma inequality (Tropp, 2012, Theorem 7.1), applied to both signs, gives, with probability at least $1 - \delta/2$,

$$\left\| \frac{1}{m} \sum_{q<m} Y_q \right\|_{\mathrm{op}} \le W^2 \sqrt{\frac{32 \log(4d/\delta)}{m}}.$$

Expanding $(\xi_q + a_q)(\xi_q + a_q)^\top - \xi_q \xi_q^\top$ and using $\|\xi_q\| \le W$ adds at most $2W\Phi e + \Phi^2 e^2$ in operator norm. For a d-dimensional symmetric matrix, the nuclear norm is at most d times the operator norm, proving Eq. (69).

In the covariance comparison in Lemma 3, the recursion acquires the additional term $D_i(\Sigma - \widehat{\Sigma}_m)D_i^\top$. Therefore

$$\chi_k^S \le \frac{2\rho c_D c_C \mathcal{V} e_H + c_D^2 e_\Sigma}{1-\rho^2}.$$

Only the last summand is new, and its terminal contribution is $Lc_C^2 c_D^2 e_\Sigma/[2(1-\rho^2)]$. The enlarged $\mathcal{V}$ bounds the covariance traces of both models; for a computable guarantee, use the known $\mathcal{V}_{\mathrm{pub}}$ from Eq. (71). The proof of the main theorem is otherwise unchanged. □

A.6 Necessity results and limitations

We first show that neither directional information nor accounting for the calibration cost can be dropped, and then discuss the limitations.

Proposition 4 (A missing decision-relevant direction obstructs identification)**.** *Without excitation outside the span of past query points, even noiseless quadratic observations need not identify the best next optimizer. In particular, there are two positive-definite Hessians that produce identical complete calibration histories but opposite unique best one-step methods, while both methods satisfy the uniform stability assumption (A2) under both Hessians. Thus some decision-relevant directional information or excitation is necessary for identifying the next method uniformly over this class. The claim is not that the particular full-rank PE condition (A3) is necessary for every restricted model class.*

Proof. Take $p = d = 2$, $C = I$, $b = 0$, no noise, and

$$H_0 = \mathrm{diag}(1, 0.8), \qquad H_1 = \mathrm{diag}(1, 1.2).$$

For both candidate methods, let the known state matrix be $A_i = I$, and set

$$D_{\mathsf{A}} = P_{\mathsf{A}} = \begin{pmatrix} 1 & 0 \\ -0.2 & 1 \end{pmatrix}, \qquad D_{\mathsf{B}} = P_{\mathsf{B}} = \begin{pmatrix} 0.8 & 0 \\ 0 & 1 \end{pmatrix}.$$

For $H_a = \mathrm{diag}(1, a)$, the two closed-loop matrices are

$$F_{\mathsf{A}}(H_a) = I - P_{\mathsf{A}} H_a = \begin{pmatrix} 0 & 0 \\ 0.2 & 1-a \end{pmatrix}, \qquad F_{\mathsf{B}}(H_a) = I - P_{\mathsf{B}} H_a = \begin{pmatrix} 0.2 & 0 \\ 0 & 1-a \end{pmatrix}.$$

At $a \in \{0.8, 1.2\}$,

$$\|F_{\mathsf{A}}(H_a)\|_{\mathrm{op}} = \sqrt{0.08} < 0.283, \qquad \|F_{\mathsf{B}}(H_a)\|_{\mathrm{op}} = 0.2.$$

Thus both models and both methods lie in the known class of (A2) with $\mu = 0.8$, $L = 1.2$, and $\rho = 0.29$.

Now initialize at $x_0 = t_0 \mathbf{e}_1$, $t_0 \neq 0$, and use method B during calibration. Under either Hessian,

$$x_q = 0.2^q t_0 \mathbf{e}_1, \qquad g_q = H_a x_q = 0.2^q t_0 \mathbf{e}_1.$$

Hence the entire calibration history is identical, and the augmented feature Gram matrix has no information in the $\mathbf{e}_2$ direction. At any resulting state $x = t\mathbf{e}_1$, method A moves to $0.2t\mathbf{e}_2$, while method B moves to $0.2t\mathbf{e}_1$. Their one-step losses are

$$f_{H_a}(x_{\mathsf{A}}^+) = 0.02at^2, \qquad f_{H_a}(x_{\mathsf{B}}^+) = 0.02t^2.$$

Method A is uniquely best under H_0, and method B is uniquely best under H_1, with gap $0.004t^2$ in either model. Since the data laws are identical, any randomized selector has worst-case identification error at least $1/2$ and minimax value regret at least $0.002t^2 > 0$. □

Proposition 5 (The calibration cost cannot be omitted)**.** *A guarantee that charges only the $B - m$ post-calibration updates can be arbitrarily different from the comparator that uses the full budget B, even in a one-dimensional noiseless quadratic.*

Proof. Let $f(x) = x^2/2$, $x_0 \neq 0$, $B = 3$, and $m = 2$. Take two stable gradient maps, $T_{\mathsf{A}}(x) = 3x/4$ and $T_{\mathsf{B}}(x) = x/2$. The predictable calibration prefix applies A twice, after which the exact post-calibration choice applies B; its terminal loss is $81x_0^2/2048$. The best fixed method applies B

three times and has loss $x_0^2/128$. Thus the full-budget regret is $65x_0^2/2048$, arbitrarily large as $|x_0|$ grows, even though the post-calibration selection error is zero. The two distinct calibration states also make the one-dimensional affine-feature Gram matrix positive definite. The exponentially forgotten calibration cost in Eq. (60) is precisely the term that prevents this accounting error when $B - m > 0$. □

Remark 6 (Selected noise is handled, not assumed away). At a decision point, the choice may depend arbitrarily on the observed g_q, and the same g_q drives the selected update. The proof remains valid because $Q_{i,r}(z, g)$ conditions on g before comparing actions. By contrast, suppose that one selects an action using g and then evaluates it with an unconditional covariance formula. Then terms such as $\mathbb{E}[D_{i(g)}\xi\xi^\top D_{i(g)}^\top]$ cannot be replaced by a mixture of $D_i\Sigma D_i^\top$, because the selected index and the quadratic noise are correlated. Fresh noise is needed only for the gradients that follow the first update of a block; the first update itself is handled by conditioning on the observed gradient.

Remark 7 (What the theorem does and does not cover). The theorem gives a learned router with a guarantee under the full budget and observable PE; it does not solve the unstructured problem.

- It learns H, b (and, by Corollary 4, Σ) on the same irreversible trajectory. It makes adaptive post-response method choices at $O(\log B)$ decision points.
- It does not require a hold action (a step without an update), frozen batches, counterfactual rollouts, or known current gradients.
- It requires a verifiable persistently exciting calibration policy and a known stable model class. Without decision-relevant directional information, Proposition 4 rules out uniform identification of the next method over the stated class. Without contraction of the fixed continuations, calibration damage need not be forgotten, and the vanishing full-budget bound on the positive part can fail.
- The affine-linear optimizer state must be Markov sufficient, that is, it must contain all optimizer memory. Adam's nonlinear coordinatewise normalization is not represented by Eq. (15) without a further nonlinear perturbation analysis.

Remark 8 (Limitation: dormant memories). Suppose that z stacks separate memories and that an unselected memory is literally frozen. Then the corresponding action matrix has an identity block, and the uniform Euclidean contraction condition in Eq. (18) fails. The potential-descent Lemma 4 still holds exactly, but the horizon-independent sensitivity constants and the exponentially vanishing calibration cost do not follow as written. They can be recovered only with a problem-specific seminorm on the reachable states, a convention for transferring or resetting memory, or a direct uniform finite-horizon sensitivity bound. This is a genuine limitation of the scope, which should not be hidden by calling the primal iterate alone the state.

B Experimental details

B.1 Common protocol: budget, target quantity, and reporting

Each oracle call produces one applied update of the single actual trajectory. Candidate-point losses, full-objective values, test values, and realized counterfactual paths are computed by the evaluation code only for reporting and are hidden from routers. Previews of candidate states and shadow memories are available only to policies whose protocol explicitly uses them. The primary target quantity is the terminal pseudo-regret

$$\mathbb{E}[\mathcal{L}_B^{\mathrm{route}}] - \min_{i\in[K]} \mathbb{E}[\mathcal{L}_B^{(i)}],$$

where $\mathcal{L}_B^{\mathrm{route}}$ and $\mathcal{L}_B^{(i)}$ are the terminal losses of the router and of fixed method i after B steps. The minimum is outside the expectation. We reserve *seedwise-oracle gap* for the stronger pathwise diagnostic $\mathbb{E}[\mathcal{L}_B^{\mathrm{route}} - \min_i \mathcal{L}_B^{(i)}]$. Unless a protocol states otherwise, all confidence intervals below are two-sided Student intervals over the prespecified units of analysis (paired before aggregation when applicable). For instance, the single-task check of PE–RCR under random excitation (Appendix B.6) uses paired normal Monte Carlo intervals with multiplier 1.95996398454. Archived figure legends call the exact-model receding control "oracle postresponse"; the text and Figure 1 use the name "exact-model receding".

B.2 PRE-SPECIFIED EXPERIMENT WITH SHARED OPTIMIZER MEMORY

Tasks and methods. The tasks satisfy the affine-linear structure, stability, and budget accounting of the main theorem, with $d = 2$, $p = 4$, and one shared memory $v \in \mathbb{R}^2$; the PE assumption is checked empirically for this task generator rather than proved almost surely. Candidate i uses

$$v^+ = \omega_i v + \psi_i g, \qquad x^+ = x - P_i g - \tau_i v, \tag{72}$$

so every method reads and overwrites the same actual memory. Equivalently, $z^+ = A_i z - D_i g$ with the known augmented matrices induced by Eq. (72). There is no private dormant state, shadow update, reset, or model copy. The four SPD preconditioners and memory coefficients are known for each task and yield noncommuting full 4×4 closed-loop maps. The unknown objective is centered, with a diagonal Hessian in the known class $0.8I \preceq H \preceq 1.2I$; $b = 0$ and the task covariance are known. The projected estimator symmetrizes the raw Hessian, zeroes its off-diagonal entries, and clips both diagonal entries to $[0.8, 1.2]$. Checks at the endpoints of this convex class, combined with the perturbation $H - I$, give the verified uniform bound

$$\sup_{i,H} \|A_i - D_i H C\|_2 \le 0.992007183 < \rho = 0.995.$$

Calibration uses predictable round robin for the first $m = \lfloor B/2 \rfloor$ steps, and the specified PE fallback is the first portfolio method (`M1` in the released code). The comparator is the smallest exact expected B-step loss among the four fixed methods. The controls share the calibration prefix (and the fitted ones also share the estimator): fitted PE–Commit, fitted PE–Post, predictable blockwise rerouting, the same-prefix fixed continuation (calibration prefix, then the best fixed method), true-model predictable commit, and exact-model receding routing.

Protocol. We tuned the task generator and the methods during development, using only seeds reserved for design, and then froze them; we rejected an initial design in which PE–RCR lost to PE–Commit, and the released design history records the subsequent tuning. An earlier sample from the frozen generator, analyzed during development, is not reported here; its code, design history, and results are released under `experiments/pe_rcr_stateful_inclass_v1/`. The present experiment tests the value of later gradients (PE–RCR minus PE–Post) prospectively. Without further tuning or scientific modification, it reuses the frozen task generator, known model class, four-method portfolio, projected ridge fit, calibration, scores, tie-breaking rule, PE fallback, and budget accounting. Before any full task or outcome was generated, an internal lock (code hashes recorded before the draw) fixed PE–Post as the primary control, fresh task and noise seeds, the budgets, the target quantity, the task-level interval, and the success criterion across all three budgets. An independent check of the seed ranges and an explicit clearance after the lock preceded the full draw. This internally locked experiment within one task generator is not an independently designed scientific confirmation; it has neither an external registry nor a trusted timestamp and is not a neutral population study.

PE–Post shares PE–RCR's calibration, projected fit, first post-calibration gradient, fitted conditional scores, tie-breaking rule, and first action, then applies that action to all remaining updates; only later decisions can differ. The new sample contains 64 independently generated tasks, 16 paired noise realizations per task and budget, and $B \in \{64, 128, 256\}$. The unit of analysis is the task: we first average each paired difference over its 16 noise realizations and then form a two-sided Student 95% interval across tasks. The prespecified success criterion requires negative upper endpoints for PE–RCR minus PE–Post at all three budgets simultaneously. The standardized paired task effect (the mean task-level difference divided by its standard deviation across tasks) and all other comparisons are descriptive. Exact replay confirms that both methods have the same state, gradient, scores, and action at the first decision point in all 3,072 task–budget–noise combinations.

At budgets $64, 128, 256$, the mean differences PE–RCR minus PE–Post are -1.15109×10^{-4}, -2.61636×10^{-5}, and -2.04920×10^{-5}, with task-level 95% intervals $[-1.28543, -1.01676] \times 10^{-4}$, $[-3.01931, -2.21341] \times 10^{-5}$, and $[-2.43211, -1.66629] \times 10^{-5}$. Thus the pre-specified all-budget criterion is met. PE–RCR has the lower task mean on 64/64, 63/64, and 62/64 tasks, respectively; the descriptive standardized effects are -2.1405, -1.6219, and -1.3368. Figure 1 shows the terminal values and the pre-specified primary difference.

The pre-specified secondary comparisons narrow down the mechanism. PE–Post minus PE–Commit is 6.998×10^{-7} $[-2.004, 3.403] \times 10^{-6}$ at $B = 64$ and exactly zero at $B = 128, 256$. PE–RCR minus the predictable rerouter has means and intervals $-3.689[-4.153, -3.226] \times 10^{-5}$,

$-2.190[-2.535, -1.846] \times 10^{-5}$, and $-2.009[-2.300, -1.718] \times 10^{-5}$. Against exact-model receding routing, the gap of PE–RCR is small but not zero at every budget: at $B = 64$ it is $2.274[0.522, 4.026] \times 10^{-6}$, so PE–RCR is detectably worse, whereas the intervals at $B = 128, 256$ contain zero.

An independent validation script, which does not import the experiment code, regenerates all tasks and noise realizations, refits all 3,072 models, and replays 36,864 terminal paths and 73,728 routing decisions. It checks the lock, summaries, and hashes, verifies that exactly 5,505,024 oracle calls and as many actual updates were made, and reports PASS. This independent software replay is not an independent scientific confirmation. All 64 designated PE draws pass at each budget, but this finite-panel diagnostic is not an almost-sure PE theorem. The locked PE–Post code omits the specified fallback for a PE failure; since all 3,072 checks pass, this unreachable, counterfactual branch cannot affect the panel. The complete lock, released code and data, and replay report are under `experiments/pe_rcr_post_confirmatory_v1/` and `results/pe_rcr_post_confirmatory_v1/`. The experiment remains a small synthetic test with $d = 2, p = 4$ and known covariance; it does not establish performance for nonlinear or production optimizers.

B.3 Post hoc analysis of the scoring horizon

After seeing the outcomes of the pre-specified experiment, we replayed the same frozen tasks and noise realizations with two exact-model controls. The first, exact one-step greedy, observes the gradient at every decision point but minimizes only the objective right after the update, then keeps that action for the specified block. The second, block-local scoring, removes this mismatch between scoring horizon and block length by minimizing the expected loss at the end of the current block. Exact-model receding routing instead scores continuing with the same candidate through the full remaining horizon. This analysis was designed post hoc and is descriptive; it is neither an independent confirmation nor a new draw of tasks.

Exact-model receding minus exact one-step greedy is $-4.532[-13.912, 4.848] \times 10^{-6}$, $-2.415[-2.991, -1.839] \times 10^{-5}$, and $-2.544[-3.228, -1.861] \times 10^{-5}$ at $B = 64, 128, 256$. Thus the interval contains zero at $B = 64$ and lies below zero at the two larger budgets. In contrast, receding routing is worse than block-local scoring at all three budgets, with gaps $2.235[1.638, 2.833] \times 10^{-5}$, $5.725[3.692, 7.758] \times 10^{-6}$, and $4.115[1.954, 6.276] \times 10^{-6}$. On the states actually visited by exact receding routing, its action differs from one-step scoring at $44.3\%, 28.5\%, 29.2\%$ of decision points and from block-local scoring at $25.3\%, 6.1\%, 5.0\%$. Hence the scoring horizon changes decisions, but the particular full-horizon commit score is not empirically optimal on these draws. The independent validation script regenerates all 64 tasks and frozen noise innovations, replays 9,216 exact terminal paths and 64,512 routing decisions, verifies the budget accounting and the parity of exact receding routing, and reports PASS. Outputs are under `results/pe_rcr_myopic_posthoc_v1/`.

B.4 Nonlinear problems with private optimizer state

This experiment lies outside our theory. It draws 24 tasks from each of two locally generated families. The first is a randomly rotated, rippled, and ridge-regularized six-dimensional Rosenbrock objective with bounded Rademacher noise on values and gradients. The second is a 33-parameter 2-8-1 tanh network trained on 256 randomly warped XOR examples with minibatches of size 32. The budgets are $B = 280$ and $B = 320$, respectively. Full-objective values are computed only for reporting and are hidden from all policies.

The oracle is richer than in our positive theorem, which uses only gradients: each step returns one common noisy value–gradient pair. Delayed-credit EXP3 uses the current value to score its previous arm; the other routers use the gradient and their specified state. No candidate-specific oracle response is drawn.

The portfolio contains SGD, two heavy-ball methods with momentum parameters 0.5 and 0.9, Nesterov with momentum parameter 0.9, and AdamW. Inactive private optimizer states stay frozen for the fixed methods, the phase schedule, local-secant rollout, delayed-credit EXP3, random routing, and round robin. Greedy Alignment instead shadow-updates exactly its three momentum filters before applying one update to the iterate; we do not silently add Nesterov and AdamW to this

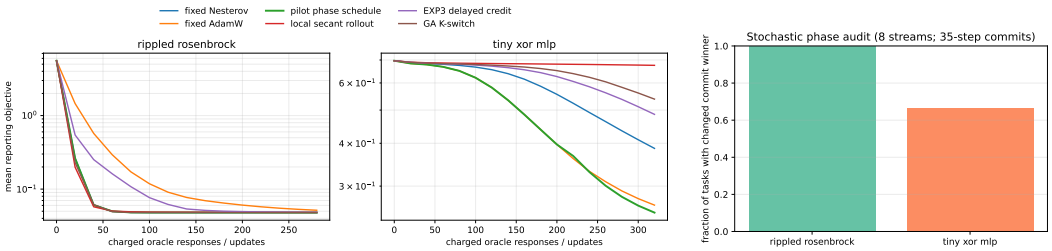

Figure 3: Diagnostics for nonlinear problems beyond the theory. Left: full-objective values, computed only for reporting, against the number of oracle calls and updates. Right: descriptive fraction of tasks whose point minimizer among 35-step commits from zero optimizer memory differs between the initial and the schedule-switch reference iterates, averaged over eight diagnostic noise realizations. The diagnostic discards the current private memory, performs no multiplicity-adjusted test for a winner, and is not used by any policy.

comparator. Every path uses B common oracle calls and B updates, with no hold, reset, model copy, candidate-point loss, or counterfactual gradient.

On each task, six calibration noise realizations select one fixed comparator. Three disjoint internal test noise realizations are paired across policies, averaged within each task, and summarized by task-level Student intervals. We chose a one-switch schedule using only separate pilot seeds and froze it before the full draw: Nesterov-to-SGD at $0.75B$ for Rosenbrock and AdamW-to-heavy-ball at $0.65B$ for the neural tasks. This schedule is a predictable open-loop control, not a learned router. We release the frozen schedules, pilot summary, and source hashes, but did not retain the complete candidate grid or candidate-level pilot outcomes, so the historical schedule selection cannot be replayed exactly. The reported test tasks and noise realizations are disjoint from the pilot.

Relative to the fixed method selected by calibration, the schedule has task-mean gaps -5.93×10^{-5} $[-7.90, -3.95] \times 10^{-5}$ and -0.00889 $[-0.01266, -0.00512]$ on Rosenbrock and the neural tasks, respectively. Greedy Alignment improves Rosenbrock by -4.82×10^{-5} $[-6.48, -3.16] \times 10^{-5}$ but is worse on the neural tasks. Both learned adaptations give negative results: local-secant rollout has gaps 2.44×10^{-4} and 0.415, and delayed-credit EXP3 has gaps 3.53×10^{-4} and 0.226; all four intervals lie above zero. These are unadjusted descriptive intervals over the prespecified policy–family endpoints, not multiplicity-adjusted claims of a winner.

Comparison restricted to Greedy Alignment's action set. The common five-method comparator above includes Nesterov and AdamW, which the faithful Greedy Alignment rule cannot select. We therefore ran a post hoc revision analysis on the unchanged, locked calibration and test records, but selected the fixed comparator only from Greedy Alignment's exact action set: SGD and the two heavy-ball methods. The six calibration noise realizations select heavy ball with momentum parameter 0.9 on all 24 tasks in both families. Greedy Alignment minus this fixed method is -1.08198×10^{-4} with task-level 95% interval $[-1.39596, -0.76801] \times 10^{-4}$ on Rosenbrock (24/24 task means favor Greedy Alignment), and 0.25029 $[0.21047, 0.29011]$ on the tiny network (0/24). Matching the action set thus strengthens the Rosenbrock result but leaves the neural failure unchanged. This derived comparison is not prospective and does not extend Greedy Alignment to the full portfolio, which would require a new score for Nesterov and AdamW. Source code, per-task records, and an independent reconstruction are released under `results/nonlinear_stateful_ga_fairness_v1/`.

With zero optimizer memory, the point minimizer changes on $24/24$ Rosenbrock tasks and $16/24$ neural tasks, but this count of noisy argmins does not establish a preference reversal with private optimizer state. The evaluated test paths contain $475{,}200$ paired rounds of one oracle call and one update; the complete experiment also spends $432{,}000$ fixed-policy calibration rounds and $134{,}400$ diagnostic commit rounds. The independent validation script reconstructs all 48 diagnostics and 132 stratified policy paths and finds no forbidden probes, mutations of inactive states, nonfinite values, or divergence events. On these small synthetic families, the transferred open-loop schedule, not the diagnostic with zero optimizer memory, shows that phase-dependent switching can help; the failed learned adaptations do not extend the quadratic theorem.

B.5 PRE-SPECIFIED CONFIRMATION ON RANDOM TASKS WITHIN THE THEORY

Tasks and methods. Each two-dimensional task independently draws both Hessian eigenvalues in $[0.81, 0.98] \times [1.02, 1.19]$, a rotation, an affine term with norm in $[0.08, 0.68]$, an initial state with norm in $[1, 4]$, and a rotated anisotropic Rademacher noise covariance. The known model class is

$$0.8I \preceq H \preceq 1.2I, \qquad \|b\| \le 0.75,$$

and the estimator is ridge regression with parameter 10^{-6}, followed by the exact Euclidean projection onto this class: spectral clipping of the symmetrized Hessian and radial projection of the affine term. The four task-specific symmetric positive-definite preconditioners have separated random rotations and are pairwise noncommuting. For every generated preconditioner and every Hessian in the full known class,

$$\|I - PH\|_2 \le \|I - P\|_2 + \|P\|_2\|I - H\|_2 = 1 - \lambda_{\min}(P) + 0.2\lambda_{\max}(P) \le 0.912 < \rho = 0.915.$$

For each $B \in \{64, 128, 256\}$, calibration consists of $m = \lfloor B/2 \rfloor$ predictable round-robin updates. PE–RCR then uses the exact projected ridge fit, the known task covariance, scores conditioned on the observed gradient, and blocks whose lengths halve (rounded down).

Protocol. We independently draw 64 new tasks from this family. Before any of these tasks was drawn, the released file `experiments/pe_rcr_confirmatory_v1/DESIGN_LOCK.json` fixed the task generator, all ten policies, four primary comparisons (PE–RCR minus PE–Commit, minus predictable plug-in rerouting, minus the same-prefix fixed continuation, and minus exact-model receding routing), budgets, numbers of noise realizations, budget accounting, unit of analysis, and seed ranges for future tasks and noise. A smoke test of the design used a separate seed range. The final task seeds begin at $2{,}027{,}500{,}000$ and the oracle-noise seeds at $2{,}028{,}000{,}000$; neither overlaps the seeds of an earlier internal test sample from this family, which is not reported here (its code and results are released under `experiments/pe_rcr_heldout_v1/`). This is an internal confirmation locked in advance, not an external or blinded replication.

We use budgets $B = 64, 128, 256$, 16 paired noise realizations per task and budget, and the task as the unit of analysis. The ten locked policies are the four fixed methods, the exact best fixed method, the same-prefix fixed continuation (calibration prefix, then the best fixed method), the fitted predictable PE–Commit, predictable plug-in rerouting, PE–RCR, and exact-model receding routing. The exact best fixed method is P_1, P_2, P_3, P_4 on $21, 11, 6, 26$ tasks, respectively. Every trajectory uses exactly B oracle calls and B updates, with no holds (steps without an update), resets, model copies, or shadow states.

The released summary records the task-level signed pseudo-regret estimates. For the direct difference PE–RCR minus PE–Commit, the means and 95% intervals are -1.01496×10^{-4} $[-1.60624, -0.423682] \times 10^{-4}$, -1.56861×10^{-4} $[-2.17929, -0.957933] \times 10^{-4}$, and -2.09024×10^{-4} $[-2.60120, -1.57928] \times 10^{-4}$. For PE–RCR minus predictable rerouting, the corresponding intervals are $[-1.91540, -1.08854] \times 10^{-4}$, $[-2.20660, -1.35867] \times 10^{-4}$, and $[-2.56717, -1.67853] \times 10^{-4}$. Thus both differences with predictable policies are strictly negative at every tested budget, while the exact-model receding router remains better than PE–RCR. The intervals for the same-prefix fixed continuation contain zero.

All $64/64$ independent PE outcomes designated by the code pass at every budget; the exact two-sided 95% interval for the pass probability is $[0.943991, 1]$. Descriptively, all $3{,}072/3{,}072$ fits (per task, budget, and noise realization) also pass. These observations are specific to this task generator; they prove neither almost-sure PE nor a vanishing failure probability under random PE. The mean fraction of PE–RCR actions that agree with the exact score rises from 0.757 to 0.860, while the projected parameter error falls from 0.230 to 0.172.

For scale only, we also report post hoc descriptive normalizations. PE–RCR's mean improvement over the best fixed method is $4.63\%, 7.26\%, 9.86\%$ of the exact best fixed method's residual above the optimal value $f^\star = \min_x f(x)$, and its direct advantage over PE–Commit is $4.82\%, 7.45\%, 9.92\%$ of that residual. Relative to the initial excess loss, the same routing gains are only $0.00268\%, 0.00420\%, 0.00570\%$. These normalizations were not primary target quantities of the lock. The explicit constants of the theorem remain vacuous: the median statistical terms range from 3.22×10^5 to 4.55×10^5, whereas the pathwise cap is about 72.2.

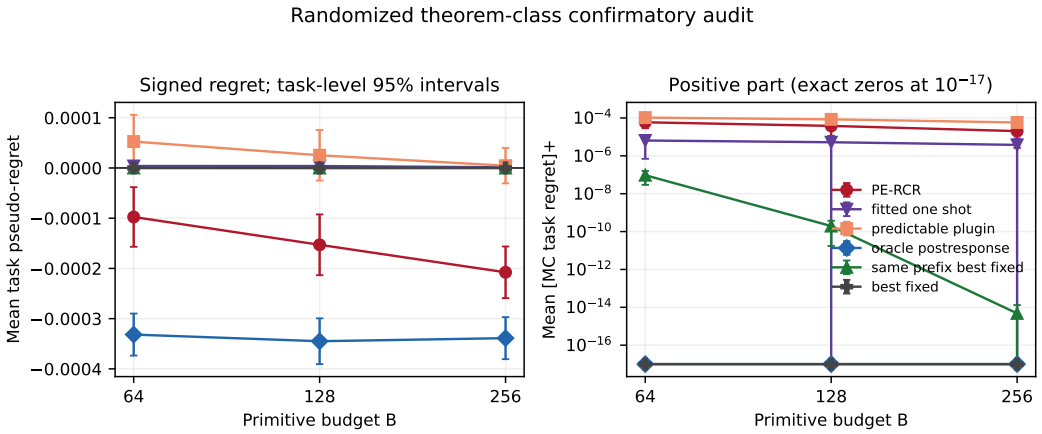

Figure 4: Pre-specified confirmation on random tasks within the theory (internal lock). Left: signed pseudo-regret averaged over tasks, with 95% intervals across tasks. Right: per-task means over finitely many noise realizations, clipped at zero (positive parts), a conservative Monte Carlo diagnostic.

An independent validation script, which does not import the experiment code, regenerates all 64 tasks and replays 3,072 fits, 30,720 terminal paths, and 67,584 routing decisions. It checks 23 output-file hashes, five source-code hashes, the internal lock, projections, scores, budget accounting, and all summaries. In total, the actual trajectories use 4,587,520 oracle calls and as many updates.

B.6 Single-task check of PE–RCR under random excitation

This experiment runs an exact finite-dimensional implementation of Algorithm 1 on one fixed task and satisfies the theorem's deterministic conditions on the model, projection, stability, budget, and routing. However, its realized Gram matrix depends on the noise, so the applicable version of the theorem is the random-PE extension in Remark 3, not the almost-sure PE version of Assumption (A3). The PE checks of the sampled designs below verify the implementation; they neither prove a distributional bound on $\delta_{\mathrm{PE}}(B)$ nor establish the asymptotic rate under random PE.

The task has $d = p = 2$ and $A_i = C = I$, and the portfolio consists of three symmetric positive-definite, pairwise noncommuting preconditioners with updates $x^+ = x - P_i g$. The known convex model class is the product of the diagonal Hessian box $[0.70, 1.30] \times [1.40, 2.20]$ and the Euclidean ball $\|b\| \le 0.75$. The true parameters are $H = \mathrm{diag}(1.15, 1.90)$, $b = (-0.40, 0.25)$, and $x_0 = (2.20, -1.70)$. Since the spectral norm is convex, it suffices to check uniform stability at the vertices of the box, where

$$\max_{i,H} \|I - P_i H\|_2 = 0.918816745 < 0.93.$$

The smallest commutator norm of two preconditioners is 0.00375, so the portfolio is not a commuting scalar-stepsize example.

Each noise coordinate is an independent Rademacher variable with values ± 0.12; thus $\Sigma = 0.0144 I$, the noise is bounded, and its covariance is known. For $B \in \{40, 80, 160, 320\}$, the first $m = \lfloor B/2 \rfloor$ updates use predictable round robin over the three portfolio methods. The implementation then fits ridge regression with $\lambda = 10^{-6}$, applies the exact Euclidean projection onto the specified model class, computes the post-response scores, and uses blocks of length $\ell(r) = \max\{1, \lfloor r/2 \rfloor\}$. The PE threshold set in the code is 4×10^{-4}. All 1,024 sampled fits (one per budget and seed) pass, and the smallest observed PE ratio is 5.58914×10^{-4}. The mean projected model error decreases from 0.35088 at $B = 40$ to 0.20427 at $B = 320$.

We log eight policy labels and use 256 paired seeds per budget, which gives 8,192 terminal runs. The label `known_model_best_fixed` deliberately duplicates P_3, the path of the exact comparator. We select the fixed comparator by its exact expected B-step value before any noise is drawn. Hence common-random-number differences remain unbiased estimates of our pseudo-regret, whose benchmark is a minimum of expectations. At $B = 80$, PE–RCR at-

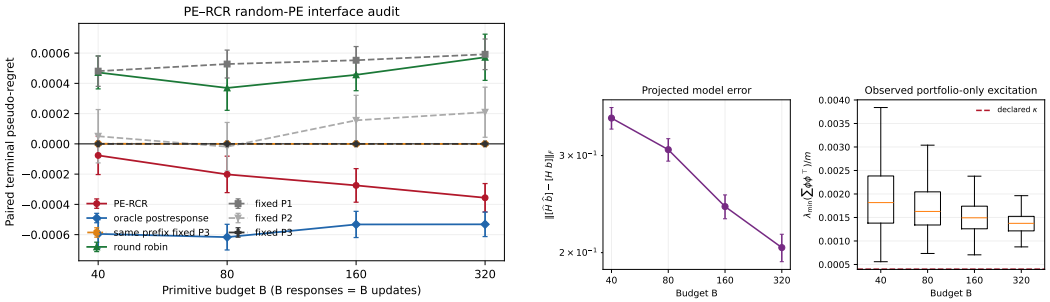

Figure 5: Single-task check of PE–RCR under random PE. Left: paired terminal pseudo-regret against the best fixed optimizer selected before any noise is drawn (bars: paired normal 95% Monte Carlo intervals). Right: projected parameter error and observed PE ratios of the calibration policy, which uses only portfolio methods. PE–RCR, exact-model receding routing, and the prefix-then-fixed control use the same calibration prefix; every curve uses B oracle calls and B updates.

tains -2.01772×10^{-4} $[-3.22255, -0.812887] \times 10^{-4}$; at $B = 320$, it attains -3.56202×10^{-4} $[-4.50054, -2.62351] \times 10^{-4}$ (mean and 95% interval). The exact-model receding router uses the same calibration prefix and attains -5.31871×10^{-4} $[-6.12827, -4.50916] \times 10^{-4}$ at $B = 320$. A second control uses the same prefix and then commits to P_3, the best fixed method selected before any noise is drawn. Its paired gaps for $B = 40, 80, 160, 320$ are, respectively,

$$1.03743 \times 10^{-6}\ [-0.603164, 2.67801] \times 10^{-6},$$
$$1.73471 \times 10^{-8}\ [-0.974499, 4.44392] \times 10^{-8},$$
$$-8.53440 \times 10^{-12}\ [-1.94202, 0.235143] \times 10^{-11},$$
$$-7.04731 \times 10^{-19}\ [-3.00368, 1.59422] \times 10^{-18}.$$

Thus the calibration prefix alone does not explain the gain of the learned router. Negative regret is possible because a routed trajectory may outperform every fixed method.

The independent validation script does not import the experiment code. It replays all 8,192 paths, 15,360 routing decisions, and 1,024 estimators, and rechecks the conditional scores, projections, budget accounting, summaries, matrix checks, and all recorded hashes. It reports zero failures of the PE check, budget accounting, replay, or integrity.

B.7 Small datasets with a split-sample comparator

This descriptive experiment has three tasks: logistic regression on breast cancer, ridge regression on diabetes, and logistic regression on digits 0 versus 8. Seven policies run for 500 steps on each of 30 evaluation seeds, which gives 630 runs and 315,000 paired steps. The candidate optimizers are SGD, momentum, and Adam. Candidate updates are pure previews: only the selected optimizer updates its state, inactive states remain bitwise unchanged, and Adam's bias-correction counter counts only the steps that use Adam.

As a split-sample proxy for the best fixed method, we select one fixed comparator per task on thirty disjoint calibration seeds $30, \ldots, 59$; this requires 270 additional runs of fixed policies. All differences (router minus comparator) use the evaluation seeds $0, \ldots, 29$. Calibration selects fixed Adam for both logistic regression tasks and fixed momentum for diabetes. On the calibration seeds, the unadjusted interval for the runner-up minus the selected method is $[0.00407563, 0.00654845]$ for breast cancer, $[-0.000500756, 0.00243154]$ for diabetes, and $[0.000579616, 0.000615343]$ for digits. Of these three unadjusted intervals, only the one for diabetes contains zero. The resulting gaps to the selected fixed method are split-sample proxies, not literal evaluations of the unknown population minimum over fixed methods.

By point estimate, the strongest routing baselines and their gaps to the selected fixed method are

- breast cancer, Adam-to-SGD: 0.000456589 $[-0.000114600, 0.00102778]$;
- diabetes, Adam-to-SGD: 0.000126126 $[-0.00169605, 0.00194830]$; and

- digits, round robin: 0.000178489 $[0.000157632, 0.000199346]$.

No routing interval lies strictly below zero. The separate diagnostic that takes the minimum over fixed methods on the same seed is retained only as `seedwise_oracle_gap`. It targets a different quantity; in particular, its value for Adam-to-SGD on diabetes is 0.00190603 $[0.000818073, 0.00299399]$.

B.8 Code, data, and independent validation

The experiment scripts and validation scripts of the reported experiments are in

- `experiments/pe_rcr_post_confirmatory_v1/`, with results in `results/pe_rcr_post_confirmatory_v1/` (Appendix B.2); the frozen task generator and its development history are in `experiments/pe_rcr_stateful_inclass_v1/`;
- `experiments/pe_rcr_myopic_posthoc_v1/`, with results in `results/pe_rcr_myopic_posthoc_v1/` (Appendix B.3);
- `experiments/nonlinear_stateful_v1/` and `experiments/nonlinear_stateful_ga_fairness_v1/`, with results in `results/nonlinear_stateful_v1/` and `results/nonlinear_stateful_ga_fairness_v1/` (Appendix B.4);
- `experiments/pe_rcr_confirmatory_v1/`, with results in `results/pe_rcr_confirmatory_v1/` (Appendix B.5);
- `experiments/pe_rcr_inclass_v1/`, with results in `results/pe_rcr_inclass_v1/` (Appendix B.6); and
- `experiments/shared_benchmark_v1/`, with results in `results/shared_benchmark_v1/` (Appendix B.7).

All paths are relative to `optimizer_selection_single_trajectory/`. Each main result directory contains raw records or sufficient statistics, a machine-readable configuration, summary tables, figures, and SHA-256 checksums. Each experiment has an independent validation script (it does not import the experiment script) and, where its protocol produces one, a validation report. The split-sample benchmark on small datasets also contains all independent calibration runs and the check of its comparator selection. Deterministic gzip timestamps and suppressed PDF timestamps make its regeneration from scratch reproducible byte for byte, which we checked. The supplement also contains the code of earlier development experiments that this version does not report. The source of the separate formal verification is `proofs/lean_optimizer_routing/`; the README in that directory documents its pinned toolchain, a theorem map, and the scope of what is verified.