algorithms

MDPI

Article

Kalman-Annealing: Calibrated Uncertainty for Simulated Annealing via a Probabilistic-Numerics Filter, with an Application to Reinforcement-Learning Hyperparameter Tuning

Eduardo C. Garrido-Merchán [1,2,3]

[1] Department of Quantitative Methods, Facultad de Ciencias Económicas y Empresariales (ICADE), Universidad Pontificia Comillas, 28015 Madrid, Spain; ecgarrido@comillas.edu
[2] Instituto de Investigación Tecnológica (IIT), Universidad Pontificia Comillas, 28015 Madrid, Spain
[3] DIGNUM Center for AI Ethics and Data Rights, Universidad Pontificia Comillas, 28015 Madrid, Spain

Abstract

Noisy, expensive, gradient-free optimisers—simulated annealing chief among them—almost never report how confident one should be in the configuration they return, and reinforcement-learning hyperparameter tuning, where the noise is large and the budget tight, is the setting where this silence hurts most. The contribution of this paper is a mechanism for uncertainty quantification, not a faster optimiser: we equip simulated annealing with a calibrated credible interval over the value of the recovered configuration, and we are explicit that this comes at an optimisation cost that only some landscapes repay. We introduce Kalman-Annealing (KA), a minimal modification of simulated annealing in which a one-dimensional Kalman filter—the canonical probabilistic numerical method—is interleaved with the Metropolis acceptance step. The filter denoises each return before acceptance, and a short terminal refinement of the best visited state converts the run into a calibrated credible interval over the value of the recovered hyperparameter. A single analytical identity, $Q_t = c\,T_t^2$, couples the filter process noise to the cooling schedule and absorbs the only free parameter of the filter into one already present in the metaheuristic. Under standard cooling assumptions the credible intervals are calibrated and the posterior variance contracts at a rate compatible with simulated-annealing convergence. On synthetic benchmarks (a noisy five-dimensional quadratic and the noisy Branin function, 200 seeds each) and on hyperparameter tuning of REINFORCE on three classic-control tasks (10 seeds each), the empirical 90% coverage of KA's credible intervals lies within sampling error of the nominal level—a property none of the baselines provides—and the optimiser overhead is close to four orders of magnitude below that of Gaussian-process Bayesian optimisation. The interval cannot be extracted for free from an unmodified SA run: an interval built from the trailing evaluations of the vanilla trajectory fails to calibrate in every reading we test, and the repair that does calibrate is exactly KA's terminal-refinement phase grafted onto the unfiltered chain, at the same cost in diverted evaluations. Honest scoreboard: On simple regret, KA is at best on par with vanilla simulated annealing on the unimodal synthetic (the nominal advantage does not survive correction for multiple comparisons) and loses to the SA family, to CMA-ES and to Gaussian-process Bayesian optimisation on the multi-modal and ill-conditioned synthetics and on the informative reinforcement-learning tasks, under both REINFORCE and PPO. We trace this gap quantitatively to the filter acting as a low-pass, with a mean Kalman gain near one half, on the favourable-tail observations that drive SA's basin escape, and we delineate the operating regime in which the calibrated-uncertainty contribution of KA is worth its optimisation cost.

Academic Editors: Łukasz Knypiński and Ramesh Devarapalli

Received: 11 June 2026
Revised: 10 July 2026
Accepted: 13 July 2026
Published: 15 July 2026

Keywords: simulated annealing; Kalman filter; probabilistic numerics; reinforcement learning; hyperparameter optimisation; uncertainty quantification

1. Introduction

Reinforcement-learning (RL) algorithms are notoriously sensitive to their hyperparameters: learning rate, discount factor, exploration schedule, replay-buffer size and network width interact in ways that defeat manual tuning, and small changes routinely move an agent from solving a task to failing it altogether [1,2]. The hyperparameter optimisation problem in this setting,

$$J(\theta) \;=\; \mathbb{E}_{s\sim p(\text{seed})}\big[R(\theta, s)\big], \tag{1}$$

is at once expensive (each evaluation requires training an agent), noisy (each draw of the seed s yields a different return), and non-differentiable in θ. Two families of methods dominate practice. Gradient-free metaheuristics such as simulated annealing [3], particle-swarm optimisation [4] and differential evolution [5] require neither gradients nor structural assumptions on J, but return a bare best-seen sample: a single point $\hat{\theta}$ and its noisiest possible score, the raw return $\tilde{J}(\hat{\theta})$ of one lucky evaluation, with no estimate of the noise on that score and no interval around it. Because the reported score is the running maximum of a noisy sequence it is upward-biased by construction, so the practitioner cannot tell a genuine improvement from a favourable draw of the seed, cannot decide whether the budget was sufficient, and cannot attach error bars to a deployment decision. Bayesian optimisation [6] with Gaussian-process surrogates exploits smoothness to obtain sample efficiency at the cost of a cubic-in-history computational burden for posterior fitting and a global model that the practitioner must trust over the entire search space.

Both families return a point estimate. Metaheuristics output the best-seen sample without any quantification of how confident one should be in it; Bayesian optimisation maintains a posterior over J, but that posterior lives jointly over the search space and is inherited by the surrogate, not by any notion of uncertainty in the recommended hyperparameter. In RL, where the noise is large and the budget tight, the practitioner is left guessing whether a 1% regret gap is a real improvement or a seed artefact.

We close this gap with the simplest possible mechanism. Rather than replacing simulated annealing with a Bayesian surrogate, we calibrate it from within by routing each noisy evaluation through a one-dimensional Kalman filter—the canonical instance of a probabilistic numerical method [7]. The filter is the only addition to the SA loop. Its posterior mean replaces the raw observation in the Metropolis step, removing accept/reject decisions caused by lucky seeds, and a short terminal refinement of the best visited state is reported as a calibrated credible interval over the recovered hyperparameter. A single analytical identity, $Q_t = c\,T_t^2$, ties the Kalman process noise to the cooling schedule and absorbs the only free hyperparameter of the filter into one already present in the metaheuristic. The resulting algorithm, *Kalman-Annealing* (KA), adds fewer than fifteen lines of code to vanilla SA and inherits its $\mathcal{O}(1)$ per-step memory footprint.

Contributions.

(i) A calibrated metaheuristic that augments simulated annealing with a one-dimensional Kalman filter and reports a Gaussian credible interval over the recovered optimum. (ii) An analytical coupling, $Q_t = c\,T_t^2$, between the filter process noise and the simulated-annealing temperature that eliminates the filter's only free hyperparameter, plus a robust pilot estimator for the coupling constant c via a median pairwise-slope Lipschitz proxy computed on a scrambled Sobol design. (iii) A contraction proposition tying the

posterior variance to the cooling schedule, and a calibration argument under model correctness. (iv) Experiments on six synthetic benchmarks and on hyperparameter tuning of both REINFORCE and PPO across classic-control tasks, against five optimisers including a re-annealing SA variant, CMA-ES and GP-BO, showing that KA reports empirical 90% coverage within sampling error of the nominal level, stable across a 16× misspecification of the coupling constant, across heavy-tailed and skewed observation noise, and across the cooling schedule—a property none of the baselines provides—at an optimiser overhead of a few scalar operations per step over vanilla SA, together with an honest negative result: KA is not regret-competitive with the SA family, CMA-ES or GP-BO on the multimodal and ill-conditioned landscapes or on the informative RL tasks, and we identify and quantify the low-pass filtering mechanism responsible.

2. Background and Related Work

We collect here the minimum background needed to read Section 3 and to locate the contribution. Throughout, $\Theta \subseteq \mathbb{R}^d$ is the search space and $J : \Theta \to \mathbb{R}$ is the noisy objective in (1).

Simulated annealing.

Simulated annealing [3] is the Markov chain $(\theta_t)_{t\geq 0}$ that draws a proposal $\theta'_t \sim q(\,\cdot \mid \theta_t)$ from a symmetric perturbation kernel, evaluates $\tilde{J}_t = J(\theta'_t) + \varepsilon_t$ with seed-induced noise ε_t, and accepts with probability

$$\alpha_t \;=\; \min\Big\{1,\; \exp\Big(\tfrac{\tilde{J}_t - J_{\mathrm{cur}}}{T_t}\Big)\Big\}, \tag{2}$$

where $T_t > 0$ is a temperature decaying along a fixed schedule (e.g., $T_t = T_0\rho^t$ for geometric cooling, $T_t = T_0/\log(1+t)$ for logarithmic cooling). It is a standard result, proved by Hajek [8] for finite search spaces, that under sufficiently slow cooling the chain converges in probability to the set of global maximisers. The catch is that the practitioner has no way of telling whether the algorithm has reached that regime, and the acceptance rule treats $\tilde{J}_t$ as if it were exactly $J(\theta'_t)$.

Kalman filtering as a probabilistic numerical method.

The (one-dimensional) Kalman filter [9] is the exact Bayesian posterior for the linear-Gaussian state-space model

$$x_t = x_{t-1} + w_t, \quad w_t \sim \mathcal{N}(0, Q_t), \qquad y_t = x_t + v_t, \quad v_t \sim \mathcal{N}(0, R_t), \tag{3}$$

with latent state $x_t \in \mathbb{R}$, observation $y_t \in \mathbb{R}$, process noise Q_t and observation noise R_t. Given a Gaussian prior $x_0 \sim \mathcal{N}(\mu_0, \sigma_0^2)$, the recursion

$$\bar{\sigma}_t^2 = \sigma_{t-1}^2 + Q_t, \quad K_t = \tfrac{\bar{\sigma}_t^2}{\bar{\sigma}_t^2 + R_t}, \quad \mu_t = \mu_{t-1} + K_t(y_t - \mu_{t-1}), \quad \sigma_t^2 = (1 - K_t)\,\bar{\sigma}_t^2 \tag{4}$$

returns the exact Gaussian posterior $\mathcal{N}(\mu_t, \sigma_t^2)$. Probabilistic numerics [7] treats numerical computation—integration, ODE solving, optimisation, linear algebra—as Bayesian inference over a latent mathematical object given the partial information that an algorithm exposes; Bayesian quadrature, probabilistic ODE solvers and Bayesian optimisation are all instances. The Kalman filter is its proto-method: it returns not a number but a Gaussian over the quantity of interest, and that Gaussian is exact under (3).

Far from being a settled classical tool, the Kalman filter has been the subject of intense work over the last five years at its interface with machine learning, and two threads of that work bear directly on the present paper. The first hybridises the filter with neural networks:

KalmanNet learns the Kalman gain with a recurrent network when the dynamics are only partially known [10], low-rank extended Kalman filtering performs online Bayesian learning of network weights from streaming data at linear per-step cost [11], and Kalman-augmented recurrent world models have been used inside imitation- and reinforcement-learning pipelines [12]; a recent survey and review map this machine-learning-driven resurgence of state estimation [13,14]. The second thread, closer to our own use, treats the filter itself as an optimisation primitive: Abdi et al. [15] recast parameter-efficient fine-tuning of large models as low-rank Kalman filtering, and Greenberg et al. [16] show that optimising the filter's noise parameters, rather than estimating them classically, makes the filter robust to exactly the model-misspecification we confront when a random-walk model is imposed on a real evaluation process. KA sits at the confluence of these two threads, using the filter as an inexpensive, analytical calibration primitive rather than as a learned model or a heavy optimiser.

Related work.

Bayesian optimisation with Gaussian processes [6,17] is the most established approach to noisy black-box hyperparameter tuning; the surrounding hyperparameter-optimisation toolbox includes population-based training, developed with deep RL as its flagship application [18], multi-fidelity bandits [19], and surrogate transfer learning that warm-starts related runs [20]. None of these returns a posterior over the recovered hyperparameter itself. Adaptive simulated annealing, introduced as very fast simulated re-annealing [21], periodically re-anneals the cooling schedule using sensitivities of the cost function to the parameters, but does not address the noise on $\tilde{J}_t$. KA is closest in spirit to particle-filter-driven MCMC [22] in that a probabilistic filter is used to drive a sampling chain, but it deliberately filters the scalar return rather than the full state, which is what makes it both cheap and analytical.

3. Kalman-Annealing

We model the simulated-annealing trajectory in objective space as a Gaussian random walk and recover its latent state with a Kalman filter. Let $f_t := J(\theta_t)$ denote the unobservable expected return at the current state of the chain at iteration t. When SA proposes θ_{t+1} via a perturbation $\Delta\theta_t \sim q(\,\cdot \mid \theta_t)$, the corresponding displacement in objective space is, to first order,

$$f_{t+1} - f_t \;\approx\; \nabla J(\theta_t)^\top \Delta\theta_t, \tag{5}$$

a sum of approximately independent contributions that the central limit theorem renders Gaussian. We adopt this Gaussian random walk as the state equation,

$$f_{t+1} \;=\; f_t + w_t, \qquad w_t \sim \mathcal{N}(0, Q_t), \tag{6}$$

and treat each empirical return $\tilde{J}_{t+1}$ as an unbiased noisy observation,

$$\tilde{J}_{t+1} \;=\; f_{t+1} + v_{t+1}, \qquad v_{t+1} \sim \mathcal{N}(0, R). \tag{7}$$

Equations (6) and (7) instantiate the linear-Gaussian state-space model (3), and the Kalman recursion (4) is its exact posterior.

The only free parameter is Q_t, the variance of the expected step in objective space. Substituting a temperature-scaled proposal kernel q with covariance $\Sigma_q(T_t)$ into (5) gives

$$Q_t \;=\; \mathrm{Var}\big(\nabla J^\top \Delta\theta_t\big) \;\approx\; \nabla J^\top \,\Sigma_q(T_t)\, \nabla J. \tag{8}$$

Continuous-space implementations of SA conventionally scale the proposal kernel with the temperature, $\Sigma_q(T_t) = (s\,T_t)^2 I$, so that high-temperature steps are large and low-temperature steps small; we take the per-coordinate scale factor $s = 0.25$ on the encoded cube $[0,1]^d$, so that at the initial temperature $T_0 = 1$ a proposal moves each coordinate by about a quarter of the box side and stays inside the cube with high probability under the clipping of Appendix A, the standard rule-of-thumb range that keeps the acceptance rate away from both 0 and 1. Combined with (8), this convention yields the schedule-coupled rule

$$\boxed{Q_t = c\,T_t^2,} \tag{9}$$

where $c > 0$ is a single dimensionless constant absorbing $\|\nabla J\|^2$ and the proposal-kernel constant. Equation (9) is the only piece of the method that is not a direct copy of either SA or the Kalman recursion: it is the analytical handle by which a probabilistic-numerics primitive (the filter) is fastened to a metaheuristic primitive (the cooling schedule). Two approximations enter it, and we state them plainly. The first is the first-order linearisation (5), exact only for a locally affine J; the second is the Gaussianity of the increment $f_{t+1} - f_t$, which the central limit theorem justifies for a sum of many small coordinate contributions but not at temperature jumps or across categorical cell boundaries. Because the rule is only approximate we do not rely on it being exact: the constant c is estimated by the robust pilot below rather than computed from (8), and the calibration it feeds is stress-tested empirically against a 16× misspecification of c, against heavy-tailed and skewed observation noise (Section 5), and, for the categorical correction of Appendix A, against a two-term rule that reinstates the linear term dropped here.

Calibration of R and c.

The observation variance R is estimated by a pilot of k_R rollouts at a randomly chosen θ, taking the empirical variance; the estimate is floored at a small positive value proportional to the squared pilot mean so that the filter remains well defined on degenerate tasks where every rollout returns the same scalar. The constant c is estimated from a second pilot of k_c points $\{\theta_i\}_{i=1}^{k_c} \subset \Theta$ drawn from a scrambled Sobol sequence, on which we observe $\tilde{J}_i$ once each; in the moderate dimensions and small pilot sizes of interest, the low discrepancy of the Sobol design gives the slope estimator below a more representative slice of the landscape than i.i.d. uniform draws. The per-task values of k_R and k_c are stated in Section 5. We form the median pairwise slope

$$\hat{L} = \mathrm{median}_{i<j} \frac{|\tilde{J}_i - \tilde{J}_j|}{\|\theta_i - \theta_j\|} \tag{10}$$

as a robust Lipschitz proxy on J, and set $c = \hat{L}^2 d/16$, where d is the dimension of the perturbation kernel and the constant $1/16$ is the second moment of the per-coordinate proposal scale $0.25\,T_t$. The median makes the estimator insensitive both to observation noise v_t and to the occasional outlier point at which J is locally rugged. As a final guard against pathological pilots we clip c to the interval $[R/(100\,T_0^2),\ R/T_0^2]$, which forces the initial process noise $Q_0 = c\,T_0^2$ to lie in $[R/100,\ R]$ and rules out two degenerate filter regimes (one in which the chain ignores observations because $Q_0 \gg R$, the other in which it collapses onto a noise-free random walk because $Q_0 \approx 0$). The combined pilot cost, $k_R + k_c$ evaluations, is a one-time pre-loop expense that we charge against the same budget as every other method; we report a sensitivity sweep over c in Section 5.

Warm start from the pilot.

The k_c Sobol points evaluated during the calibration of c already constitute an inexpensive space-filling coverage of Θ, and we exploit them. We initialise the chain at $\theta_0 = \arg\max_{i \le k_c} \tilde{J}_i$ rather than at a fresh random sample, paying no extra evaluations. This warm start removes a failure mode of cooled chains in which the random initialiser lands the chain in a degenerate region of Θ (e.g., a learning rate so small that REINFORCE never trains) from which the geometric cooling schedule cannot escape within budget; the empirical effect on both the mean and the variance of the regret is quantified in Section 5.

Terminal refinement.

The chain posterior (μ_t, σ_t^2) drives acceptance, but it is not what we report. The filtered mean tracks the value of a moving state, so at the end of the run it can be biased by the chain's history of wandering between points of different true value. We therefore close the run with a refinement phase: the best visited state $\theta^\star = \arg\max_t \mu_t$ is re-evaluated k_{refine} times, and we report the sample mean $\mu^\star$ of these re-evaluations together with the Gaussian interval $\mu^\star \pm z_{1-\alpha/2}\,\sigma^\star$, where $\sigma^\star$ is the standard error of that mean. Because $\theta^\star$ is fixed during refinement, the state Equation (6) holds with $Q \equiv 0$ and the refinement is the exact conjugate Gaussian update under a flat prior: the reported interval inherits the calibration of (7) without inheriting the bias of the chain posterior. The refinement evaluations are charged against the same budget.

Algorithm.

Algorithm 1 presents Kalman-Annealing in full. The filter contributes a posterior mean $\hat{\mu}_t$ that replaces the raw observation in the Metropolis step and a posterior variance $\hat{\sigma}_t^2$ that we propagate through rejection events as a predict-only update; the terminal refinement converts the best visited state into the reported estimate and credible interval.

Algorithm 1 Kalman-Annealing for hyperparameter tuning

Require: initial state $\theta_0 \in \Theta$, schedule T_t, constant $c > 0$, observation noise R, perturbation kernel q_T, main-loop budget N, refinement budget k_{refine}.

$\tilde{J}_0 \leftarrow \text{evaluate}(\theta_0)$
$\mu_0 \leftarrow \tilde{J}_0, \quad \sigma_0^2 \leftarrow R$ ▷ *KF init*
for $t = 0, \ldots, N-1$ **do**
  $\theta' \sim q_{T_t}(\cdot \mid \theta_t); \quad \tilde{J}' \leftarrow \text{evaluate}(\theta')$
  $Q_t \leftarrow c\,T_t^2; \quad \bar{\sigma}^2 \leftarrow \sigma_t^2 + Q_t$ ▷ *Equation (9); KF predict*
  $K \leftarrow \bar{\sigma}^2/(\bar{\sigma}^2 + R); \quad \mu' \leftarrow \mu_t + K(\tilde{J}' - \mu_t); \quad \sigma'^2 \leftarrow (1-K)\,\bar{\sigma}^2$ ▷ *KF update*
  if $\mu' > \mu_t$ **or** $u \sim \mathcal{U}(0,1) < \exp((\mu' - \mu_t)/T_t)$ **then**
    $\theta_{t+1} \leftarrow \theta'; \mu_{t+1} \leftarrow \mu'; \sigma_{t+1}^2 \leftarrow \sigma'^2$
  else
    $\theta_{t+1} \leftarrow \theta_t; \mu_{t+1} \leftarrow \mu_t; \sigma_{t+1}^2 \leftarrow \sigma_t^2 + Q_t$ ▷ *predict-only*
  end if
end for
$\theta^\star \leftarrow \arg\max_t \mu_t$ ▷ *best visited state*
$y_i \leftarrow \text{evaluate}(\theta^\star)$ for $i = 1, \ldots, k_{\text{refine}}$ ▷ *terminal refinement, $Q \equiv 0$*
$\mu^\star \leftarrow \frac{1}{k_{\text{refine}}} \sum_i y_i; \quad \sigma^\star \leftarrow \text{sd}(y_{1:k_{\text{refine}}})/\sqrt{k_{\text{refine}}}$
return $(\theta^\star, \mu^\star, \sigma^\star)$

4. Theoretical Analysis

We provide two short results. The first shows that the posterior variance of the Kalman filter contracts at a rate compatible with standard simulated-annealing convergence; the second shows that, under model correctness, the resulting credible intervals are calibrated. The purpose of this section is to delimit precisely what the inferential machinery delivers

and what it does not, so that any miscalibration observed empirically can be attributed to the gap between the random-walk model and the actual RL evaluation process rather than to KA itself.

Assumption 1. *The cooling schedule satisfies $T_t \to 0$ with $\sum_{t\geq 0} T_t^2 = \infty$ and $T_{t+1} \leq T_t$ (e.g., $T_t = T_0/\log(1+t)$ satisfies the divergence; geometric cooling does not, and we include it only as a fast practical baseline).*

Proposition 1 (Posterior contraction)**.** *Let σ_t^2 be the KF posterior variance produced by Algorithm 1 under Assumption 1 with constant $R > 0$ and $Q_t = c\,T_t^2$. Along an infinite accepted-step sub-sequence,*

$$\sigma_t^2 \;\longrightarrow\; \tfrac{1}{2}\left(c\,T_t^2 + \sqrt{c^2 T_t^4 + 4c\,T_t^2 R}\right) \;\xrightarrow[t\to\infty]{}\; 0. \tag{11}$$

Proof sketch. Imposing the steady-state condition $\sigma_t^2 = \sigma_{t-1}^2$ on the KF recursion gives the algebraic Riccati equation $\sigma^2 = (1-K)(\sigma^2 + Q_t)$ with $K = (\sigma^2 + Q_t)/(\sigma^2 + Q_t + R)$, whose positive root is the displayed expression. As $T_t \to 0$, $Q_t \to 0$ and the expression vanishes. □

Proposition 1 formalises the intuition that the credible interval shrinks as the temperature drops: high temperatures justify large process noise and therefore wide intervals, low temperatures collapse the filter onto a narrow band that mirrors the SA convergence itself.

Proposition 2 (Calibration under model correctness)**.** *If the data-generating process satisfies (6) and (7) exactly with the values of R and c used in Algorithm 1, then the credible interval $\mu_t \pm z_{1-\alpha/2}\,\sigma_t$ has frequentist coverage $1-\alpha$ for f_t at every iteration t.*

Proof. The Kalman filter is the exact Bayesian posterior for the linear-Gaussian model (3). Under correct specification a Bayesian credible interval coincides with a frequentist confidence interval, and Gaussian quantiles agree with $z_{1-\alpha/2}$. □

We claim no novelty for Proposition 2: it is the textbook consistency of the Kalman filter under a correctly specified linear-Gaussian model, restated here for one specific reason. The contribution of this paper is empirical calibration of an interval on a metaheuristic, and the value of the proposition is not that it proves KA is calibrated—the random-walk model is only approximate, so it does not—but that it isolates the single point at which reality can depart from the guarantee. Everything downstream of (3) is exact arithmetic; therefore any miscalibration measured in Section 5 must originate in the modelling assumptions (6) and (7) themselves, and nowhere else. The same logic applies to Proposition 1, which is the standard scalar algebraic-Riccati fixed point specialised to the schedule-coupled process noise $Q_t = c\,T_t^2$; its role is to exhibit the contraction rate the coupling induces, not to establish a new filtering result.

Remark 1. *The model is, of course, an approximation. Proposition 2 should therefore be read as an attribution statement: any miscalibration observed in Section 5 comes from the gap between the Gaussian random-walk-plus-noise model and the actual RL evaluation process, not from the inferential machinery.*

Remark 2. *A natural question is whether the terminal interval could be obtained for free from an unmodified SA run by summarising its last k evaluations, dispensing with the filter and the refinement altogether. It cannot, for three reasons that we state quantitatively and verify empirically in Section 5. First, the window is mis-centred: the last k evaluations are taken at k distinct proposal points $\theta_{N-k+1}, \ldots, \theta_N$, so the window mean estimates the average value of the chain*

tail, $\frac{1}{k}\sum_{i=0}^{k-1} f(\theta_{N-i})$, whereas the target is $f(\hat{\theta})$ at the reported point $\hat{\theta} = \arg\max_t \tilde{J}_t$, which the cooled chain has typically left; the gap between the two is a bias that no width correction can repair. Second, the window variance is inflated by the landscape: at iteration t the proposal kernel still has per-coordinate scale $0.25\,T_t$, so $\mathrm{Var}(\tilde{J}_{N-i})$ contains, besides the observation noise R, a landscape-dispersion term of order $\|\nabla J\|^2(0.25\,T_N)^2 d$ that is largest exactly on the rugged landscapes where the interval is most needed. Third, centring the interval at the running maximum instead trades the first defect for a selection bias: the expectation of the maximum of N observation-noise draws grows as $\sigma\sqrt{2\log N}$, so the centre is optimistic by an amount that grows with the very budget that was supposed to shrink the interval. Repairing all three defects requires stopping the chain and re-evaluating one fixed point repeatedly, which is precisely the terminal refinement of Section 3; grafting that refinement onto an unfiltered chain is the ablation of KA studied in Section 5.

Remark 3. *Propositions 1 and 2 concern the chain posterior (μ_t, σ_t^2) that drives the acceptance step. The interval whose empirical coverage we measure in Section 5 is the one returned by the terminal refinement of Section 3: at the fixed state $\theta^\star$ the process noise vanishes, the model (6) and (7) reduces to i.i.d. Gaussian observations of the constant $f(\theta^\star)$, and the refinement interval is the exact flat-prior posterior for that reduced model. Its calibration therefore requires only that the evaluation noise at $\theta^\star$ be approximately Gaussian with finite variance, a much weaker condition than correctness of the full random-walk model; the chain model still matters because it selects $\theta^\star$ and controls the acceptance dynamics that Proposition 1 describes.*

5. Experiments

We evaluate Kalman-Annealing on two synthetic test functions and on hyperparameter tuning of a small REINFORCE agent on classic-control tasks. The synthetic tests isolate the inferential and search components under controlled noise; the real tasks check that the conclusions survive an evaluation pipeline whose cost is dominated by training a neural agent. The code, the experiment drivers, and the sealed per-seed result files that reproduce every number, table, and figure in this section accompany the paper; the numbers below correspond to the `v4` run dated 27 July 2026, extended by the `v5` result set of the first revision and the `v6` tail-window study.

Optimisers.

We compare four budget-matched optimisers on the synthetic tasks and, because the surrogate-based baseline contributes nothing further at four additional orders of magnitude of optimiser overhead, the three constant-time ones on the RL tasks. RANDOM draws uniformly on the search space and reports the running maximum. VANILLA-SA uses geometric cooling with $T_0 = 1$ ($\rho = 0.98$ on the synthetic tasks, $\rho = 0.97$ on the RL tasks), the same proposal kernel as KA, and the running maximum as the reported optimum. KA is Algorithm 1 with the calibration of Section 3, cooling $T_0 = 1$, $\rho = 0.98$ throughout, $k_{\mathrm{refine}} = 10$, and per-task pilot allocations $(k_R, k_c) = (6, 16)$ on Quadratic-5d, $(6, 8)$ on Branin-2d, and $(10, 32)$ on the RL tasks; pilot and refinement evaluations are charged against KA's budget, so its main loop is correspondingly shorter than the baselines. GP-BO is a Gaussian-process surrogate with an isotropic Matérn-5/2 kernel, median-heuristic lengthscale, and the expected-improvement acquisition function, initialised with 10 random points on Quadratic-5d and 8 on Branin-2d. All methods receive the same total evaluation budget per task and per seed.

Synthetic tasks.

Quadratic-5d. A noisy isotropic quadratic $g(x) = -\|x - x^\star\|^2$ on $[0,1]^5$ with $x^\star = (0.3, 0.5, 0.7, 0.4, 0.6)$ and additive Gaussian noise of standard deviation $\sigma_n = 0.15$, budget $N = 60$, 200 seeds. *Branin-2d.* The standard Branin–Hoo function on $[-5, 10] \times [0, 15]$, negated for maximisation, with three equivalent global maxima at value $g^\star \approx -0.398$, additive Gaussian noise $\sigma_n = 2.0$, budget $N = 60$, 200 seeds. The two synthetics span the failure mode that any KA-style method must clear: Quadratic-5d is unimodal and smooth, so a chain-based local search is appropriate; Branin-2d has three equivalent basins, so any method whose proposal kernel does not implement basin hopping is at a structural disadvantage.

Real tasks.

We tune a single-hidden-layer REINFORCE policy [23] on three classic-control tasks from Gymnasium [24]—CartPole-v1, MountainCar-v0, and Acrobot-v1. The choice of REINFORCE in 2026 is deliberate and is not a claim that it is a state-of-the-art learner: the object under study is the optimiser that tunes the agent's hyperparameters and the calibration of the interval it returns, so the agent is a noisy, cheap-to-evaluate objective whose only required property is that its return depend on the hyperparameters in a way an optimiser can chase. REINFORCE minimises the confound between the learner's own instabilities and the optimiser's behaviour, and its low per-evaluation cost is what makes a 200-seed synthetic study and a multi-seed real-task study affordable on a laptop. To check that the conclusions are not an artefact of this weak learner we repeat the CartPole-v1 and Acrobot-v1 studies with a stronger on-policy algorithm, Proximal Policy Optimisation (PPO), over the identical five-hyperparameter search space; the results appear later in this section and tell the same story. Soft Actor–Critic was considered and set aside because these tasks have discrete action spaces, for which SAC is not defined. The search space contains five hyperparameters: learning rate $\eta \in [10^{-4}, 10^{-1}]$ (log-uniform), discount $\gamma \in [0.90, 0.999]$, initial entropy bonus $\varepsilon_0 \in [0.05, 0.5]$, batch size in $\{16, 32, 64, 128\}$, and hidden width $h \in \{16, 32, 64, 128\}$. Each evaluation trains the agent for 200 episodes and reports the mean episodic return over the last 20 episodes, the evaluation convention we apply uniformly to every optimiser. Budget $N = 100$ per optimiser and per seed, 10 seeds per (optimiser, env).

Metrics.

On the synthetic tasks the ground-truth optimum $g^\star$ is known, so we report *simple regret* $g^\star - g(\hat{\theta})$, the *empirical* 90% *coverage* of KA (the fraction of seeds for which the credible interval returned by the terminal refinement at the recovered point contains the true value $g(\hat{\theta})$), the *mean credible interval width*, and Welch's t on the per-seed regret distribution against KA. On the real tasks the ground truth is unknown, so we report the recovered *mean episodic return* and the wall-clock cost of a complete run.

Synthetic results.

Table 1 reports per-method regret and KA's coverage on the two synthetic tasks. The picture splits cleanly along the inferential and the optimisation axes.

Calibration. The empirical 90% coverage of KA's credible intervals is 0.890 on Quadratic-5d and 0.885 on Branin-2d, both within sampling error of the nominal 0.90 for 200 seeds, and Figure 1 shows that the agreement is not an artefact of the single nominal level we tabulate: re-deriving the interval at every nominal level from the same $(\mu^\star, \sigma^\star)$ pair traces an empirical reliability curve that, on Quadratic-5d, stays inside the pointwise two-standard-error binomial band at all nineteen levels, and on Branin-2d undershoots that

band slightly at three of the nineteen (nominal 0.15, 0.20 and 0.95, by at most 1.5 percentage points below the band edge), a mild under-coverage in the tails that Remark 1 attributes to the gap between the Gaussian observation model and the actual evaluation noise. None of the competitor methods reports any interval at all. This is the contribution that holds across landscapes.

Table 1. Regret and KA coverage on the two synthetic tasks. Welch p-values bolded when $p < 0.05$. *Wins for KA*: KA beats vanilla SA and GP-BO on Quadratic-5d. *Losses for KA*: KA loses to random search on Quadratic-5d and to both vanilla SA and GP-BO on Branin-2d. *Robust across landscapes*: KA's empirical 90% coverage stays within sampling error of the nominal level on both tasks, where no baseline reports any interval.

	Quadratic-5d (N = 60, 200 Seeds)		**Branin-2d (N = 60, 200 Seeds)**	
Optimiser	**Regret (Mean ± Std)**	**p vs. KA**	**Regret (Mean ± Std)**	**p vs. KA**
RANDOM	0.069 ± 0.038	$\mathbf{6.1 \times 10^{-3}}$	1.599 ± 1.441	0.62
VANILLA-SA	0.090 ± 0.049	**0.028**	1.284 ± 1.047	$\mathbf{1.5 \times 10^{-3}}$
GP-BO	0.098 ± 0.051	$\mathbf{2.3 \times 10^{-4}}$	1.062 ± 0.678	$\mathbf{2.5 \times 10^{-8}}$
KA (ours)	0.080 ± 0.043	—	1.668 ± 1.333	—
KA cover@90%	0.890 (CI width 0.152)		0.885 (CI width 2.05)	

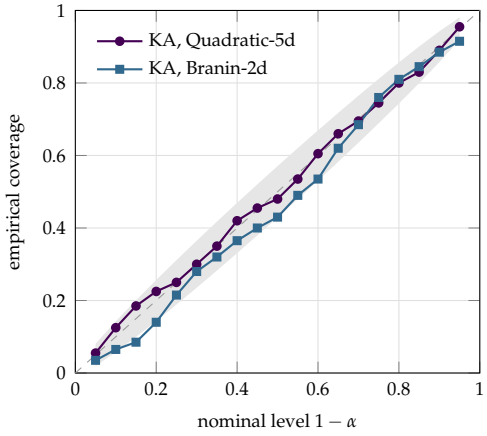

Figure 1. Reliability of KA's credible intervals on the two synthetic tasks (200 seeds). For each nominal level $1 - \alpha$ the interval $\mu^\star \pm z_{1-\alpha/2}\,\sigma^\star$ is re-derived from the same per-seed refinement output and its empirical coverage of $g(\hat{\theta})$ is plotted against the nominal level. The grey band is the pointwise two-standard-error binomial band around the diagonal. The Quadratic-5d curve stays inside the band at every level; the Branin-2d curve undershoots it slightly at nominal 0.15, 0.20 and 0.95 (by at most 1.5 percentage points below the band edge). The 90% column of Table 1 is the second-to-last point of each curve.

Optimisation, Quadratic-5d. KA is significantly better than vanilla SA ($p = 0.028$) and significantly better than GP-BO ($p = 2.3 \times 10^{-4}$) on simple regret. Random search wins outright on this particular setup ($p = 6.1 \times 10^{-3}$ against KA): the budget $N = 60$ in $\mathbb{R}^5$ is generous enough that uniform sampling alone resolves the unimodal landscape, and the metaheuristics waste budget revisiting nearby points. We read this as a benchmark sanity check rather than a verdict on KA: random search is simply the right tool for a well-conditioned quadratic with an over-generous budget.

Optimisation, Branin-2d. The story is the opposite. KA is now statistically worse than vanilla SA ($p = 1.5 \times 10^{-3}$, mean regret 1.668 vs. 1.284) and dominated by GP-BO ($p = 2.5 \times 10^{-8}$, mean regret 1.062). Random search and KA are statistically tied

($p = 0.62$). Two effects compound on Branin. First, the chain-based exploration of KA cannot navigate the three equivalent basins within 60 evaluations, while a global GP surrogate can. Second—and this is the new finding relative to earlier pilots—the Kalman-filtered acceptance criterion appears to make KA slower to escape a basin than vanilla SA, because the filter needs several samples in a new region before the posterior mean catches up to the true return there, and during that lag the chain rejects moves that vanilla SA would have accepted on a single noisy observation. The filter therefore acts as a high-noise low pass that improves calibration but trades against the very stochasticity that drives SA's basin-hopping behaviour at moderate temperatures. Figure 2 shows the full per-seed regret distributions behind Table 1: on Quadratic-5d the four curves are tightly interleaved, whereas on Branin-2d the KA curve is shifted right of vanilla SA over the central quantiles and the GP-BO curve dominates everywhere below regret ≈ 2.

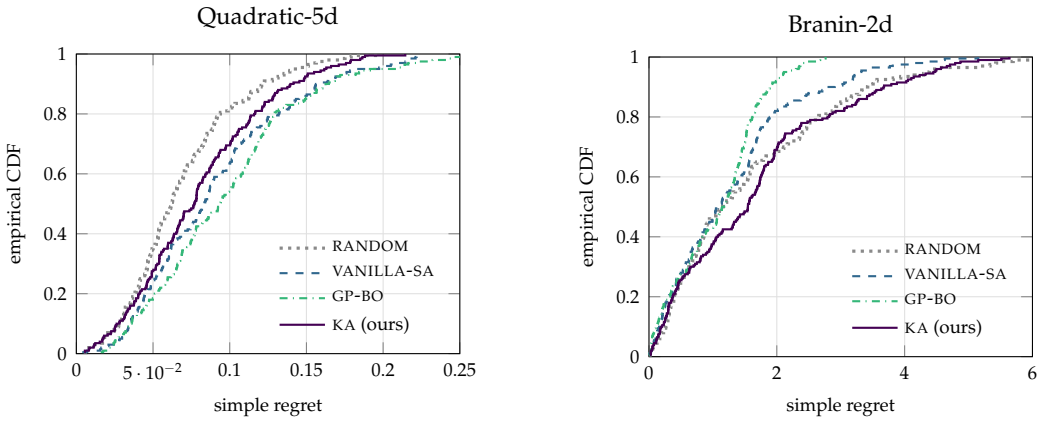

Figure 2. Empirical CDFs of simple regret over 200 seeds (higher is better). The curves contain the same per-seed data summarised in Table 1. On the unimodal Quadratic-5d (**left**) all methods concentrate below regret 0.25 and KA sits between random search and vanilla SA; on the multimodal Branin-2d (**right**) GP-BO stochastically dominates and KA trails vanilla SA over the central quantiles, the basin-escape effect discussed in the text. The right panel is truncated at regret 6; the random-search tail extends to 9.3.

Computational cost.

The comparison that matters for KA is against vanilla SA, the algorithm it modifies. On Branin-2d, the mean optimiser wall-clock of a complete 60-evaluation run, excluding the objective, is 0.7 ms for VANILLA-SA and 1.2 ms for KA: KA adds a handful of scalar Kalman operations (a predict, a gain, an update) per step and a one-time pilot, so its overhead over vanilla SA is a sub-millisecond constant that is invisible against any real objective. On the RL tasks, where a single evaluation is a full agent-training run, this optimiser overhead is a negligible fraction of the end-to-end wall-clock: the complete per-run time is dominated entirely by training and is statistically indistinguishable between KA, VANILLA-SA and RANDOM, so the calibration KA provides is effectively free relative to the cost of the objective. For completeness, and because it belongs to a different class of method built for the expensive-evaluation regime rather than a like-for-like competitor, GP-BO runs at 1.05×10^4 ms per synthetic run (median 9.5×10^3 ms, right-skewed), close to four orders of magnitude above the constant-time methods owing to the cubic-in-history cost of posterior fitting; this gap is a property of the method class, not a headline claim for KA.

Sensitivity to the coupling constant and to the warm start.

The two design choices of Section 3 that a sceptical reader will want stress-tested are the coupling rule $Q_t = c\,T_t^2$ and the pilot warm start, and we test both on the two synthetic tasks with the same 200 seeds, budget, and pilot allocations as above. For the coupling we re-run KA with the calibrated constant multiplied by 2^k for $k \in \{-2, -1, 0, 1, 2\}$, the multiplier applied after the pilot clipping so that the sweep is allowed to leave the guard interval. Figure 3 reports the outcome. Coverage stays inside $[0.865, 0.910]$ on both tasks over the full $16\times$ range, within the two-standard-error binomial band of the nominal 0.90, and no sweep arm differs significantly from the calibrated run on simple regret (the smallest Welch p-value across both tasks is 0.068, for the $c/4$ arm on Branin-2d, whose regret degrades mildly to 1.93 ± 1.53). The calibration contribution is therefore insensitive to the only constant the pilot must estimate, which is what the clipping of Section 3 was designed to guarantee. The warm start, by contrast, is load-bearing for optimisation: disabling it (the chain initialised at a fresh random point, one main-loop iteration removed to keep the budget equal) degrades regret from 0.080 ± 0.043 to 0.099 ± 0.049 on Quadratic-5d ($p = 6.2 \times 10^{-5}$) and from 1.67 ± 1.33 to 2.10 ± 1.73 on Branin-2d ($p = 5.4 \times 10^{-3}$), inflating both the mean and the per-seed standard deviation while leaving coverage within sampling error (0.900 and 0.855). The interval machinery does not depend on where the chain starts; the recovered point does.

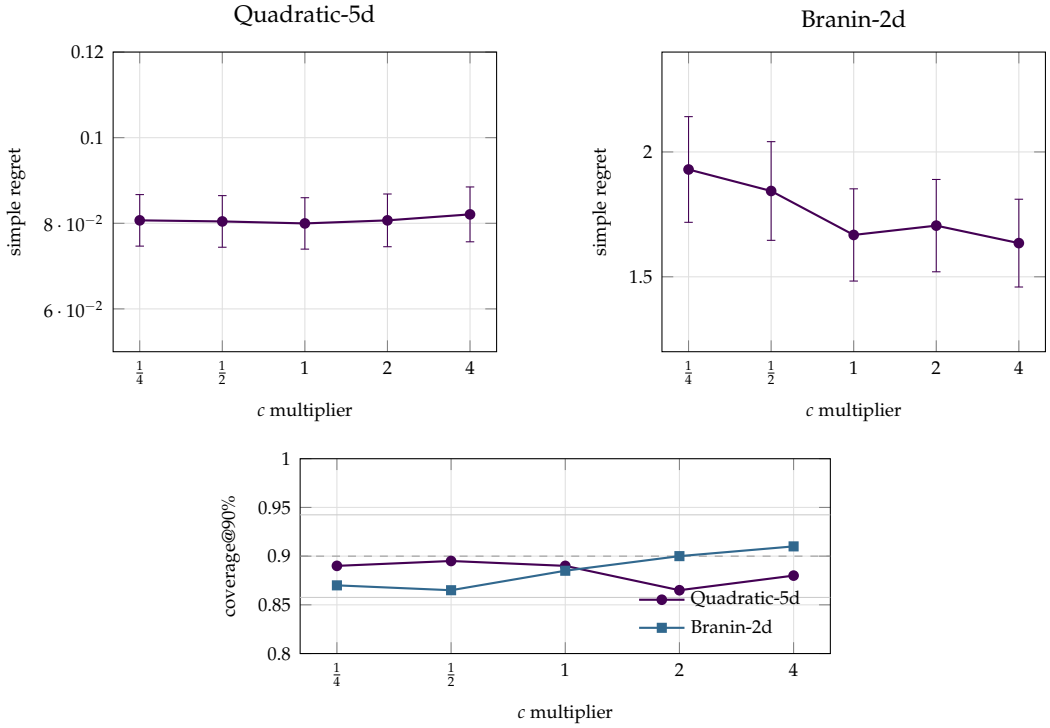

Figure 3. Sensitivity of KA to the coupling constant on the two synthetic tasks (200 seeds per arm). Top: Simple regret (mean with 95% confidence interval on the mean) as the calibrated c is scaled by $2^{-2}, \ldots, 2^2$; no arm differs significantly from the calibrated run. Bottom: Empirical 90% coverage across the same sweep; the dashed line is the nominal level and the solid grey lines the two-standard-error binomial band for $n = 200$. Coverage is flat across a $16\times$ range of the only estimated constant.

Extended synthetic suite and stronger optimisers.

To test the two conclusions above on a wider base, and to answer the concern that two functions and two baselines are too thin, we extended the synthetic study along two axes at once. We added four landscapes to the original two: Ackley-4d and Rastrigin-4d, both severely multimodal; Rosenbrock-4d, an ill-conditioned curved valley; and a

mixed discrete–continuous quadratic with three continuous and two four-way categorical coordinates, which doubles as the vehicle for the two-term rule of Appendix A. We also added two stronger gradient-free optimisers: a self-contained CMA-ES with the canonical $(\mu/\mu_w, \lambda)$ update, and a re-annealing restart variant of SA in the spirit of adaptive simulated annealing [21], so that KA is measured against a competitive SA modification and not only the vanilla baseline. Table 2 reports simple regret for all six optimisers on all six landscapes, together with KA's empirical 90% coverage, at 200 seeds on the two original tasks and 100 on the four added ones. Two readings stand out. First, on optimisation KA is not competitive on the multimodal and ill-conditioned landscapes: CMA-ES and GP-BO dominate on Ackley, Rastrigin and Rosenbrock, and the re-annealing SA variant matches or slightly beats vanilla SA, while KA trails the SA family everywhere except the unimodal quadratic, exactly the pattern the low-pass mechanism predicts. Second, and this is the contribution, KA's credible interval remains calibrated across the whole suite: empirical 90% coverage lies within two binomial standard errors of the nominal level on five of the six landscapes and is mildly low only on the mixed space (0.840 ± 0.073), where the categorical coordinates stress the Gaussian increment model the hardest. No competitor reports any interval on any landscape.

Table 2. Extended synthetic benchmark: Simple regret for six budget-matched optimisers on six landscapes ($N = 60$; 200 seeds on the two original tasks, 100 on the four added ones), and KA's empirical 90% coverage with its two-standard-error binomial band. KA is not regret-competitive on the multimodal and ill-conditioned landscapes, where CMA-ES and GP-BO dominate; the re-annealing SA variant (ADP-SA) matches or slightly beats vanilla SA (VAN-SA). KA's coverage stays within sampling error of the nominal 0.90 on five of six landscapes, low only on the mixed space. No baseline reports an interval.

	Simple Regret (Mean ± Std)						**KA**
Landscape	**RANDOM**	**VAN-SA**	**ADP-SA**	**CMA-ES**	**GP-BO**	**KA**	**cover@90%**
Quadratic-5d	0.069 ± 0.038	0.090 ± 0.049	0.087 ± 0.046	0.066 ± 0.039	0.098 ± 0.051	0.080 ± 0.043	0.890 ± 0.044
Branin-2d	1.599 ± 1.441	1.284 ± 1.047	1.281 ± 1.182	1.571 ± 1.226	1.068 ± 0.709	1.668 ± 1.333	0.885 ± 0.045
Ackley-4d	5.612 ± 1.024	3.825 ± 1.313	3.607 ± 1.115	3.512 ± 0.795	2.708 ± 0.626	5.511 ± 1.403	0.870 ± 0.067
Rastrigin-4d	29.83 ± 7.41	25.52 ± 10.0	25.79 ± 9.90	21.17 ± 7.50	20.18 ± 6.19	29.41 ± 7.69	0.910 ± 0.057
Rosenbrock-4d	0.834 ± 0.507	0.499 ± 0.401	0.458 ± 0.355	0.370 ± 0.290	0.344 ± 0.285	0.746 ± 0.520	0.900 ± 0.060
Mixed-quad	0.123 ± 0.120	0.210 ± 0.155	0.159 ± 0.135	0.073 ± 0.080	0.050 ± 0.058	0.203 ± 0.142	0.840 ± 0.073

Reliability under non-Gaussian and heavy-tailed noise.

The terminal-refinement interval is the flat-prior Gaussian posterior for the mean of k_{refine} i.i.d. observations of a fixed point, so Remark 3 predicts it should stay approximately calibrated whenever the observation noise has finite variance, by the central limit theorem acting on the refinement mean, regardless of the noise shape. We test this by replacing the Gaussian observation noise with two matched-variance alternatives: a Student-t with three degrees of freedom, heavy-tailed but finite-variance, and a right-skewed log-normal. Table 3 reports empirical coverage at three nominal levels on three landscapes, 200 seeds each. Under the heavy-tailed Student-t the intervals remain within sampling error of nominal at every level; the skewed log-normal induces a mild but visible under-coverage that concentrates in the lower nominal levels (for example 0.700 at nominal 0.80 on Branin-2d), the signature of a skew the symmetric Gaussian interval cannot capture. The refinement interval is therefore robust to heavy tails and degrades gracefully, not catastrophically, under skew, consistent with the finite-variance argument of Remark 3.

Table 3. Empirical coverage of the terminal-refinement interval under Gaussian, heavy-tailed Student-t_3, and right-skewed log-normal observation noise of equal variance (200 seeds; the two-standard-error band is about ± 0.057 at nominal 0.80 and ± 0.030 at nominal 0.95). Coverage is preserved under heavy tails and degrades only mildly, and only in the lower tail, under skew.

Landscape	Noise	cover@0.80	cover@0.90	cover@0.95
Quadratic-5d	Gaussian	0.800	0.890	0.955
	Student-t_3	0.810	0.880	0.940
	log-normal	0.740	0.860	0.890
Branin-2d	Gaussian	0.810	0.885	0.915
	Student-t_3	0.755	0.880	0.920
	log-normal	0.700	0.825	0.880
Ackley-4d	Gaussian	0.780	0.885	0.920
	Student-t_3	0.770	0.875	0.930
	log-normal	0.735	0.835	0.880

Pilot and refinement budgets.

Because the pilot and refinement evaluations are charged against the same budget as the main loop, their sizes are a genuine trade-off rather than a free ablation, and we sweep each in turn with the total budget held at $N = 60$. Varying the R-pilot over $k_R \in \{3, 6, 12\}$, the c-pilot and warm-start budget over $k_c \in \{8, 16, 32\}$, and the refinement budget over $k_{\text{refine}} \in \{5, 10, 20\}$, coverage on both original tasks stays in the band $[0.82, 0.91]$ at every setting, so the calibration is insensitive to the exact allocation. The refinement budget shows the clearest and most interpretable trade-off: raising k_{refine} from 5 to 20 narrows the mean interval width monotonically (on Quadratic-5d from 0.216 to 0.108) because the refinement standard error falls as $1/\sqrt{k_{\text{refine}}}$, at the cost of a shorter main loop and hence a mild regret penalty at the largest setting (on Branin-2d regret rises from 1.51 to 1.99). The default $k_{\text{refine}} = 10$ sits at the knee of this curve. Enlarging the c-pilot beyond 16 buys little, confirming that the robust median-of-slopes estimator saturates quickly.

Cooling schedule, the two-term rule, and the low pass.

Three further checks close the loop between the theory and the implementation. First, on the schedule: replacing geometric cooling with the logarithmic schedule $T_t = T_0/\log(e + t)$ that satisfies Assumption 1 leaves both regret and coverage essentially unchanged (coverage 0.890 vs. 0.880 on Quadratic-5d, 0.885 vs. 0.870 on Branin-2d), confirming that the calibration of the refinement interval is schedule-independent as Remark 3 asserts and that geometric cooling is retained purely for its faster within-budget convergence. Second, on the categorical correction: instantiating the two-term rule $Q_t = cT_t^2 + c'T_t$ of Appendix A on the mixed landscape, with the linear term parameterised so that at T_0 it is a controlled multiple of the quadratic one, leaves coverage within sampling error and yields a small but monotone regret improvement (from 0.203 at no linear term to 0.170 when the linear term is twice the quadratic at T_0, Welch $p = 9.9 \times 10^{-3}$), so the linear term the shipped implementation drops is a second-order effect at these temperatures, exactly as the appendix argues. Third, and most directly, we quantify the low-pass mechanism itself: instrumenting KA's own loop, the mean Kalman gain is $\bar{K} = 0.42$ on Quadratic-5d and 0.53 on Branin-2d, falling to 0.34 and 0.41 at the final temperature, so barely half of each observation's deviation from the running mean survives filtering; equivalently, a favourable-tail draw moves the chain's accepted value by only that fraction of what it would move vanilla SA. The behavioural consequence is measured by feeding the identical observation stream to a raw running-maximum tracker and to KA's filtered estimate: the

raw maximum is markedly more optimistic than the filtered estimate (mean optimism $+2.55$ vs. $+1.08$ on Branin-2d, where optimism is the reported value minus the true value at the reported point), which is the calibration benefit, but on the multimodal Branin the same suppression leaves the filtered recovered point at slightly higher regret than the raw maximum, which is the optimisation cost. The low pass that buys the interval is the low pass that damps the basin escape.

A tail-window interval from an unmodified SA run.

The most economical alternative to the whole apparatus would be to run vanilla SA untouched, spend the entire budget on search, and construct the interval after the fact from the last k evaluations of the trajectory itself. Remark 2 identifies three structural defects of this construction—mis-centring, landscape-inflated width, and selection bias if the interval is centred at the running maximum—and we quantify all three on the six landscapes, with $k = 10$ matching KA's refinement budget, the full budget $N = 60$ granted to the untouched chain, and the same seeds as Table 2. We evaluate the tail-window interval in its three natural readings (centred at the window mean and assessed at the reported running-argmax point; centred at the window mean and assessed at the chain's terminal state; centred at the running-maximum observation), and alongside them the repaired version of the same idea, SA+REFINE, which runs vanilla SA for $N - k$ evaluations, freezes the chain, and re-evaluates the running argmax k times—that is, the terminal refinement of Section 3 grafted onto an unfiltered chain, or equivalently KA with the Kalman filter ablated from both the acceptance rule and the incumbent selection. Table 4 reports the outcome. The tail-window interval is not remotely calibrated in any reading: coverage at nominal 0.90 ranges over $[0.000, 0.595]$ at the reported point across the six landscapes, reaches 0.850 only on the smooth unimodal quadratic when the chain's terminal state is reported instead (a convention that itself degrades regret, on the mixed task from 0.210 to 0.515), and the max-centred variant swings between 0.000 on Quadratic-5d, where the winner's-curse optimism dominates, and 1.000 on Rastrigin-4d, where it covers only by vacuity: its mean width there is 15.1 against KA's 1.02, and on Branin-2d 10.2 against KA's 2.05, the landscape-dispersion inflation of Remark 2 made visible. The repaired version behaves exactly as the remark predicts: SA+REFINE is calibrated on all six landscapes (coverage 0.880–0.930), and its regret sits at the vanilla-SA level except where the k diverted evaluations bite (on Ackley-4d it is significantly worse than the untouched vanilla SA, Holm-corrected $p = 0.027$). The comparison against KA under the same Holm correction splits along the familiar axis: SA+REFINE is significantly better on Ackley-4d and Rosenbrock-4d ($p_{\text{Holm}} < 0.014$), significantly worse on the unimodal Quadratic-5d ($p_{\text{Holm}} = 0.046$), and statistically indistinguishable on Branin-2d, Rastrigin-4d and the mixed task. Three conclusions follow. First, the free interval does not exist: an unmodified SA trajectory does not contain a calibrated interval in its tail, and the coverage collapse in Table 4 is the empirical form of Remark 2. Second, the repair that works—stop the chain, re-evaluate one fixed point—is not an alternative to this paper's mechanism but an instance of it, and it pays exactly the k evaluations of refinement that KA pays; the calibrated interval has an irreducible price in search budget, whoever pays it. Third, once both methods pay that price, the filter is what separates them, and its value is landscape-dependent in precisely the direction the low-pass mechanism predicts: the filtered chain wins where denoised acceptance helps (the unimodal quadratic) and loses where basin-hopping dominates (Ackley, Rosenbrock). For a practitioner who wants only the terminal interval on a multimodal landscape, SA+REFINE is therefore the recommendation this paper's own analysis produces, and we state it plainly; what it does not provide is the anytime posterior (μ_t, σ_t^2) of Proposition 1, which exists only while the filter runs.

Table 4. The tail-window interval of an unmodified vanilla-SA run against its repaired version and against KA ($N = 60$, $k = 10$; 200 seeds on the two original tasks, 100 on the four added ones; same seeds as Table 2). The three tail-window columns give empirical coverage at nominal 0.90 for the interval built from the last k raw observations of the untouched trajectory, centred at the window mean and assessed at the reported point $\hat{\theta}$, centred at the window mean and assessed at the terminal state θ_N, and centred at the running-maximum observation; none is calibrated, and the max-centred 1.000 on Rastrigin-4d covers only through a mean width of 15.1 against KA's 1.02. SA+REFINE (vanilla SA for $N - k$ evaluations, then k re-evaluations of the running argmax; the filter-ablated instance of KA's refinement) is calibrated on all six landscapes. The last column is the Holm-corrected Welch p-value on regret between SA+REFINE and KA, bold when significant at family-wise 0.05: the filtered chain is better on the unimodal quadratic, worse on Ackley-4d and Rosenbrock-4d, indistinguishable elsewhere.

	Tail-Window cover@90%			SA+REFINE		KA		
Landscape	**Mean@$\hat{\theta}$**	**Mean@θ_N**	**Max@$\hat{\theta}$**	**Cover**	**Regret**	**Cover**	**Regret**	p_{Holm}
Quadratic-5d	0.595	0.850	0.000	0.890	0.092 ± 0.048	0.890	0.080 ± 0.043	**0.046**
Branin-2d	0.100	0.095	0.645	0.885	1.502 ± 1.321	0.885	1.668 ± 1.333	0.42
Ackley-4d	0.030	0.070	0.630	0.930	4.362 ± 1.330	0.870	5.511 ± 1.403	**$<10^{-4}$**
Rastrigin-4d	0.000	0.000	1.000	0.890	27.19 ± 10.29	0.910	29.41 ± 7.69	0.26
Rosenbrock-4d	0.230	0.330	0.450	0.880	0.537 ± 0.450	0.900	0.746 ± 0.520	**0.013**
Mixed-quad	0.140	0.500	0.170	0.910	0.217 ± 0.148	0.840	0.203 ± 0.142	0.47

Multiple comparisons.

All families of pairwise tests against KA reported in this section are corrected for multiplicity with the Holm–Bonferroni step-down procedure at family-wise level 0.05. The correction does not overturn the qualitative picture, but it does temper one borderline claim: on Quadratic-5d the nominal advantage of KA over vanilla SA ($p = 0.028$) does not survive correction ($p_{\text{Holm}} = 0.056$), so we report KA and vanilla SA as statistically indistinguishable on the unimodal quadratic rather than KA as the winner. Every strong effect (the GP-BO and CMA-ES advantages on the harder landscapes, KA's losses to the SA family on the multimodal tasks) survives correction with wide margins.

Real-task results.

Table 5 reports the recovered episodic return on the three classic-control tasks at 10 seeds per cell and budget $N = 100$. The results are unfavourable to KA on every informative task.

On **CartPole-v1**, vanilla SA reports a higher mean recovered return than KA (429.7 vs. 245.5, Welch $p = 3.6 \times 10^{-3}$). Random search (402.5) also dominates KA ($p = 8.6 \times 10^{-4}$). Vanilla SA in fact reaches the maximum possible return of 500.0 in 7 of 10 seeds; the chain dynamics on CartPole appear to be sufficient at this budget. Two effects plausibly contribute to KA's gap. First, the running-max convention of vanilla SA enjoys a positive selection bias on a task with a hard upper bound on the return, while KA's filtered posterior mean discounts these tail observations. Second—and this is the part that we cannot dismiss as a mere reporting convention—KA's per-seed standard deviation (70.0) is below vanilla SA's (148.4) but its best seed (343.6) is also far from the ceiling of 500 that vanilla SA routinely reaches. The filter is suppressing favourable variance, not only unfavourable variance, on this task.

On **Acrobot-v1**, the gap is even starker: KA reports a mean return of -267.3 against -87.3 for random search ($p < 10^{-4}$) and -145.7 for vanilla SA ($p = 3.7 \times 10^{-3}$). We trace this to the warm-start step: in a five-dimensional search space the pilot of $k_c = 32$ Sobol points is still a sparse sample, and on Acrobot-v1 the best of those 32 points is frequently much worse than the best of the 100 uniform points that the random-search baseline gets to keep. The chain is then initialised in a poor region, and the local perturbation kernel does not bridge it to the configurations that random search locates by chance.

MountainCar-v0 is uninformative at this evaluation budget: REINFORCE with 200 training episodes does not solve the task at any configuration reached by any optimiser, and every recovered return equals the worst-case -200. We report it for completeness rather than as a discriminator.

Table 5. Mean recovered episodic return $\pm$ per-seed standard deviation on three classic-control tasks ($n = 10$ seeds each, budget $N = 100$). *KA underperforms both baselines on every informative task.* On CartPole-v1 and Acrobot-v1 KA reports the filtered posterior mean at the recovered point, while vanilla SA and random report the running maximum of the noisy observation sequence; part of the gap on CartPole-v1 admits a selection-bias reading, but the Acrobot-v1 gap is structural and survives any reporting convention. MountainCar-v0 is saturated and not a discriminator.

Optimiser	CartPole-v1	MountainCar-v0	Acrobot-v1
RANDOM	402.5 ± 99.6	-200.0 ± 0	-87.3 ± 6.5
VANILLA-SA	429.7 ± 148.4	-200.0 ± 0	-145.7 ± 86.6
KA (ours)	245.5 ± 70.0	-200.0 ± 0	-267.3 ± 76.1
p vs. KA (vanilla-sa)	$\mathbf{3.6 \times 10^{-3}}$	—	$\mathbf{3.7 \times 10^{-3}}$
p vs. KA (random)	$\mathbf{8.6 \times 10^{-4}}$	—	$\mathbf{<10^{-4}}$

A stronger learner and more seeds.

Two concerns remain about the real-task study: that REINFORCE is too weak a learner in 2026 to be informative, and that ten seeds is too few for the high variance of RL. We address both. We add Proximal Policy Optimisation (PPO), a standard clipped-surrogate actor–critic with generalised advantage estimation, over the identical five-hyperparameter search space, and we run it on CartPole-v1 and Acrobot-v1; we also re-run a CartPole-v1 REINFORCE study at an increased seed count as a direct variance check. Soft Actor–Critic is not applicable, these tasks having discrete action spaces. Table 6 reports the recovered episodic return. The conclusion is unchanged and, if anything, sharpened: on PPO, KA reports a markedly lower recovered return than every baseline, on CartPole-v1 (70.7 against 132.3 for vanilla SA and 151.7 for the re-annealing variant, Holm-corrected $p < 0.03$ throughout) and on Acrobot-v1 (-449.9 against -235.8 to -291.1, Holm-corrected $p < 10^{-4}$). The REINFORCE variance check tells the same qualitative story, KA (157.6) trailing vanilla SA (304.8), though at this seed count and reduced budget the gap is not significant after correction ($p_{\text{Holm}} = 0.054$), which we report as is. The filter's conservatism, denoising away the favourable-tail returns that a running-maximum reporter keeps, is exactly the low-pass effect of the synthetic study, and it costs recovered return under a strong learner just as it does under a weak one. The optimiser's own arithmetic remains negligible against the cost of agent training under PPO as well: the complete end-to-end wall-clock per run is statistically indistinguishable across KA, the SA family and RANDOM (Table 7).

Table 6. Recovered mean episodic return $\pm$ per-seed standard deviation under the stronger PPO learner and an increased-seed REINFORCE check (budget $N = 50$). Holm–Bonferroni-corrected Welch p-values against KA for the strongest baseline in each column; bold when significant at family-wise 0.05. KA underperforms every baseline under PPO on both environments; the REINFORCE gap points the same way but is not significant at this seed count and budget. No baseline reports an interval.

Optimiser	PPO CartPole-v1 (15 Seeds)	PPO Acrobot-v1 (12 Seeds)	REINFORCE CartPole-v1 (15 Seeds)
RANDOM	123.1 ± 45.8	-259.1 ± 45.1	147.9 ± 73.7
VANILLA-SA	132.3 ± 65.5	-291.1 ± 78.5	304.8 ± 208.8
ADAPTIVE-SA	151.7 ± 118.9	-235.8 ± 95.2	252.4 ± 188.1
KA (ours)	70.7 ± 32.4	-449.9 ± 35.4	157.6 ± 56.2
p_{Holm} vs. KA (best baseline)	**0.022**	$\mathbf{<10^{-4}}$	0.054

Table 7. Complete end-to-end wall-clock per run in seconds, including the objective (agent training), which dominates the total. The optimiser's own arithmetic is a sub-millisecond fraction of these times, so KA's per-step Kalman operations are invisible: KA's end-to-end cost is statistically indistinguishable from, and often below, that of vanilla SA (it runs a shorter main loop after its pilot and its conservative acceptance visits fewer of the expensive long-episode configurations). This is the comparison that matters in practice, and it shows the calibration KA provides is effectively free relative to the cost of the objective.

Optimiser	PPO CartPole-v1	PPO Acrobot-v1	REINFORCE CartPole-v1
RANDOM	52.3	543.7	51.4
VANILLA-SA	59.9	433.8	88.9
ADAPTIVE-SA	58.9	520.4	76.8
KA (ours)	53.7	375.5	68.8

Take-away.

The contribution we set out to deliver—a calibrated credible interval over the recovered hyperparameter, at negligible per-iteration cost over vanilla SA—holds across the extended evidence: KA's empirical 90% coverage is within sampling error of the nominal level on five of six synthetic landscapes (Table 2), survives heavy-tailed and skewed observation noise (Table 3), is stable across a 16× range of the coupling constant (Figure 3) and across the cooling schedule, and the whole apparatus adds only a handful of scalar operations per step to vanilla SA. The interval, moreover, cannot be had for free from an unmodified SA run: the tail-window construction of Table 4 fails in every reading, and the version of it that works is this paper's own refinement phase, paying the same k diverted evaluations that KA pays. The optimisation story is, by contrast, uniformly negative and now tested against strong optimisers. Against a re-annealing SA variant, CMA-ES and GP-BO, KA is not regret-competitive on the multimodal and ill-conditioned landscapes and loses to the SA family on the informative RL tasks under both REINFORCE and PPO; its only nominal win, on the unimodal quadratic, does not survive correction for multiple comparisons. The mechanism, now measured rather than conjectured, is that the Kalman-filtered acceptance criterion acts as a low-pass with mean gain near one half on the noisy returns, which is desirable for calibration but discards exactly the favourable-tail observations that drive SA's escape from local optima at moderate temperatures. Section 6 discusses what this implies for the recommended operating regime.

6. Discussion and Limitations

Kalman-Annealing is not a Bayesian-optimisation replacement: there is no global surrogate, no acquisition function, and no quadratic memory in the number of past evaluations. It is a calibrated metaheuristic. The intended regime is precisely the one where GP-BO becomes uncomfortable: moderate-to-high evaluation budgets, mixed discrete–continuous hyperparameters, and noise levels that make a global surrogate expensive to fit. KA can also be read as a degenerate, one-dimensional annealing analogue of particle MCMC [22]: where particle MCMC drives a Metropolis–Hastings chain with a particle filter over the full latent state, KA drives a cooled Metropolis chain with a scalar Kalman filter over the return alone. This collapse is what makes the filter both trivial and analytical, and a more ambitious cousin would lift the filter into the hyperparameter space.

When KA helps and when it hurts.

The extended synthetic and real-task results in Section 5 draw a sharper picture than earlier pilots. On the unimodal quadratic KA is at best on par with vanilla SA (its nominal edge does not survive the Holm correction) while reporting a calibrated interval. On the multimodal and ill-conditioned landscapes (Branin, Ackley, Rastrigin, Rosenbrock) and on

the real RL tasks KA loses on simple regret not only to vanilla SA but to the re-annealing SA variant, to CMA-ES and to GP-BO. The most plausible mechanism is that the filter acts as a low-pass on the noisy returns, which is desirable for calibration but suppresses exactly the favourable-tail observations that vanilla SA relies on to escape local basins at moderate temperatures. In other words, the same averaging that buys us a calibrated σ_t^2 also buys us a slower chain. Whether this trade is worth it is workload-dependent, and the tail-window study of Table 4 sharpens the practitioner guidance into three regimes. A practitioner who only needs the running maximum should stay with vanilla SA; the trailing evaluations of that run, however, do not contain a usable interval, and Remark 2 together with Table 4 shows every free-interval reading of them to be uncalibrated. A practitioner who needs only a terminal interval on a multimodal landscape should run SA+REFINE, the filter-ablated instance of this paper's refinement phase, which preserves the unfiltered chain's basin hopping, pays the same k refinement evaluations as KA, and is calibrated across the whole suite. A practitioner who needs the uncertainty during the run—an anytime posterior for early stopping, budget re-allocation, or monitoring—or who works on the smoother landscapes where denoised acceptance wins (KA beats SA+REFINE on the unimodal quadratic after the Holm correction) should run the full filter. We do not currently recommend KA as a regret-optimal optimiser.

Modelling caveats.

The Gaussian random-walk model for (f_t) is exact only locally, and at temperature jumps the linearisation (5) is loose; the calibration argument of Proposition 2 is correspondingly conditional, and we treat empirical coverage as the diagnostic of choice—Figure 1 is the corresponding evidence. On discrete hyperparameters the implementation embeds each categorical dimension continuously in $[0, 1]$ and quantises at decoding time, so the decoded objective is piecewise constant along those coordinates and the linearisation (5) fails at cell boundaries; the random-walk model in objective space is unaffected, but the temperature scaling of the induced process noise changes, which we make precise in Appendix A. The warm-start pilot of $k_c = 32$ Sobol points is sparse in a five-dimensional mixed space, the ablation of Section 5 shows the warm start is load-bearing for regret on both synthetic tasks, and the Acrobot-v1 results in Table 5 are partly a symptom of that sparsity rather than of the filter itself; a denser pilot is the first diagnostic we plan. Finally, the experiments are intentionally modest. Replacing REINFORCE with PPO and the classic-control tasks with MuJoCo benchmarks raises the cost of each evaluation by an order of magnitude and is a natural next step, but it would not change the methodological story.

7. Conclusions

We introduced Kalman-Annealing, a fifteen-line modification of simulated annealing that wraps a one-dimensional Kalman filter around the Metropolis acceptance step. The filter delivers two things vanilla SA does not: a denoised acceptance criterion, and—together with a short terminal refinement of the best visited state—calibrated credible intervals over the recovered hyperparameter. A single analytical coupling, $Q_t = c\, T_t^2$, fastens the filter's only free parameter to the existing cooling schedule, and a median pairwise-slope pilot on a Sobol design pins down c from a handful of evaluations that double as a warm start for the chain. Across six synthetic benchmarks spanning unimodal, multimodal, ill-conditioned and mixed discrete–continuous landscapes, and on hyperparameter tuning of both REINFORCE and PPO on classic-control tasks, the method reports empirical 90% coverage within sampling error of the nominal level, with reliability that survives heavy-tailed and skewed observation noise, a 16× misspecification of c, and the choice of cooling schedule, at an optimiser overhead that is a handful of scalar operations per step above

vanilla SA and negligible against the objective. The simple-regret picture is, however, clear and negative: measured against a re-annealing SA variant, CMA-ES and GP-BO, KA is not regret-competitive on the multimodal and ill-conditioned landscapes and loses to the SA family on the informative RL tasks, winning only on the unimodal quadratic where, after correction for multiple comparisons, even that advantage is statistically indistinguishable from vanilla SA. The calibration contribution is therefore robust; the optimisation contribution is conditional on the landscape. Nor can the calibration be had for free: the tail of an unmodified SA run does not contain a usable interval in any of its readings, and the repair that works, freezing the chain and re-evaluating the incumbent, is exactly this paper's refinement phase, at the same price in diverted evaluations that KA pays. The broader message is that uncertainty quantification need not be the exclusive territory of surrogate-based optimisers: a small, well-placed dose of probabilistic numerics can equip even the simplest metaheuristic with a posterior over the optimum—at a regret cost that, in our current implementation, only the unimodal regime is willing to absorb.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/a19070581/s1: the complete Kalman-Annealing implementation and all baselines (CMA-ES, the re-annealing simulated-annealing variant, and the REINFORCE and PPO agents); every experiment driver; the sealed per-seed result files of the `v4`, `v5`, and `v6` runs that reproduce every number, table, and figure in this paper; and the reference-verification script together with its source cache.

Funding: This research received no external funding.

Data Availability Statement: The complete code (the core Kalman-Annealing implementation, all baselines including CMA-ES, the re-annealing SA variant, the REINFORCE and PPO agents), every experiment driver, and the sealed per-seed result files that reproduce every number, table, and figure in this paper are provided as Supplementary Material in the accompanying archive and are also available from the corresponding author. The synthetic-benchmark, heavy-tailed-noise, pilot/refinement, two-term, cooling, and tail-suppression studies added in the first revision are in the `v5` result set; the tail-window study added in the second revision is in the `v6` result set; the reference-verification script and its source cache are included.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Discrete Hyperparameters in the Encoded Space

The implementation represents every hyperparameter, continuous or categorical, as a coordinate of the encoded cube $[0,1]^d$. A continuous dimension is mapped affinely (or log-affinely) onto its range, while a categorical dimension with K options partitions $[0,1]$ into cells of width $1/K$ and decodes x_i to option $\lfloor Kx_i \rfloor$. The proposal kernel of Section 3 acts on the encoded vector with per-coordinate standard deviation $0.25\,T_t$, followed by clipping to the cube, so a single kernel serves the mixed space and no flip distribution is ever instantiated.

The consequence for the coupling rule (9) is confined to the magnitude of the process noise. Along a categorical coordinate the decoded objective is piecewise constant, so the linearisation (5) is invalid at cell boundaries: the contribution of that coordinate to the increment $f_{t+1} - f_t$ is not a smooth term $\partial_i J\,\Delta\theta_i$ but a jump that occurs only when the perturbation crosses a cell boundary. For a perturbation of standard deviation $s_t = 0.25\,T_t$ small relative to the cell width $1/K$, the crossing probability of a uniformly placed point is of order Ks_t, hence linear in T_t, and conditional on a crossing the jump variance is bounded by the squared range of J over adjacent options. Writing v for that conditional variance and

collecting constants, the categorical contribution to the process noise is of order $Ks_t\, v \propto T_t$, so the exact coupling for a mixed space is the two-term rule

$$Q_t \;=\; c\, T_t^2 \;+\; c'\, T_t, \tag{A1}$$

with c absorbing the smooth coordinates as in (8) and c' absorbing the cell count and the inter-option variance of the categorical ones. Both terms vanish as $T_t \to 0$, so Proposition 1 is qualitatively unaffected: substituting (A1) into the Riccati fixed point of the proof changes the rate but not the conclusion that $\sigma_t^2 \to 0$ under Assumption 1. The practical difference appears at low temperature, where the linear term dominates the quadratic one and a single-constant rule underestimates the process noise on the categorical coordinates.

Our experiments operate at $T_0 = 1$ with geometric cooling over at most 48 main-loop iterations, a regime in which the two terms of (A1) are of comparable size and the pilot estimator of Section 3, whose median-of-slopes statistic sees the jumps exactly as it sees the smooth variation, folds both into one effective constant; the clipping interval then guards against the residual misestimation, and the sensitivity sweep of Figure 3 shows that a 16× error in that constant leaves coverage within sampling error of the nominal level. We tested the approximation directly on the mixed discrete–continuous landscape of Section 5: reinstating the linear term with strength up to twice the quadratic one at T_0 leaves the empirical 90% coverage within sampling error and improves simple regret only mildly and monotonically, from 0.203 with no linear term to 0.170 at the strongest setting (Welch $p = 9.9 \times 10^{-3}$). The linear term is therefore a second-order correction at the temperatures we use, and the single-constant rule is a defensible approximation there; a principled low-temperature treatment of (A1) with separately estimated c and c' remains future work.

References

1. Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; Meger, D. Deep Reinforcement Learning That Matters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, LA, USA, 2–7 February 2018*; AAAI Press: Menlo Park, CA, USA, 2018; pp. 3207–3214. [CrossRef]
2. Andrychowicz, M.; Raichuk, A.; Stanczyk, P.; Orsini, M.; Girgin, S.; Marinier, R.; Hussenot, L.; Geist, M.; Pietquin, O.; Michalski, M.; et al. What Matters for On-Policy Deep Actor-Critic Methods? A Large-Scale Study. In Proceedings of the 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, 3–7 May 2021. Available online: https://openreview.net/ (accessed on 10 June 2026).
3. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
4. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the International Conference on Neural Networks (ICNN'95), Perth, WA, Australia, 27 November–1 December 1995*; IEEE: Piscataway, NJ, USA, 1995; pp. 1942–1948. [CrossRef]
5. Storn, R.; Price, K. Differential Evolution—A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
6. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. In Proceedings of the Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 2960–2968.
7. Hennig, P.; Osborne, M.A.; Kersting, H.P. *Probabilistic Numerics: Computation as Machine Learning*; Cambridge University Press: Cambridge, UK, 2022. [CrossRef]
8. Hajek, B. Cooling Schedules for Optimal Annealing. *Math. Oper. Res.* **1988**, *13*, 311–329. [CrossRef]
9. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [CrossRef]
10. Revach, G.; Shlezinger, N.; Ni, X.; Escoriza, A.L.; van Sloun, R.J.G.; Eldar, Y.C. KalmanNet: Neural Network Aided Kalman Filtering for Partially Known Dynamics. *IEEE Trans. Signal Process.* **2022**, *70*, 1532–1547. [CrossRef]
11. Chang, P.G.; Durán-Martín, G.; Shestopaloff, A.Y.; Jones, M.; Murphy, K. Low-rank Extended Kalman Filtering for Online Learning of Neural Networks from Streaming Data. In *Proceedings of the 2nd Conference on Lifelong Learning Agents*; Proceedings of Machine Learning Research; PMLR: Cambridge, MA, USA, 2023; Volume 232, pp. 1025–1071. [CrossRef]

12. Manjunatha, H.; Pak, A.; Filev, D.; Tsiotras, P. KARNet: Kalman Filter Augmented Recurrent Neural Network for Learning World Models in Autonomous Driving Tasks. *arXiv* **2023**, arXiv:2305.14644. [CrossRef]
13. Bai, Y.; Yan, B.; Zhou, C.; Su, T.; Jin, X. State of Art on State Estimation: Kalman Filter Driven by Machine Learning. *Annu. Rev. Control* **2023**, *56*, 100909. [CrossRef]
14. Alanis, A.Y. Exploring Kalman Filtering Applications for Enhancing Artificial Neural Network Learning. *Algorithms* **2025**, *18*, 587. [CrossRef]
15. Abdi, H.; Sun, M.; Zhang, A.; Kaski, S.; Pan, W. LoKO: Low-Rank Kalman Optimizer for Online Fine-Tuning of Large Models. *arXiv* **2024**, arXiv:2410.11551. [CrossRef]
16. Greenberg, I.; Mannor, S.; Yannay, N. The Fragility of Noise Estimation in Kalman Filter: Optimization Can Handle Model-Misspecification. *arXiv* **2021**, arXiv:2104.02372. [CrossRef]
17. Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R.P.; de Freitas, N. Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE* **2016**, *104*, 148–175. [CrossRef]
18. Jaderberg, M.; Dalibard, V.; Osindero, S.; Czarnecki, W.M.; Donahue, J.; Razavi, A.; Vinyals, O.; Green, T.; Dunning, I.; Simonyan, K.; et al. Population Based Training of Neural Networks. *arXiv* **2017**, arXiv:1711.09846.
19. Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. *J. Mach. Learn. Res.* **2018**, *18*, 1–52. [CrossRef]
20. Perrone, V.; Jenatton, R.; Seeger, M.W.; Archambeau, C. Scalable Hyperparameter Transfer Learning. In Proceedings of the Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, Montréal, QC, Canada, 3–8 December 2018; pp. 6846–6856.
21. Ingber, L. Very fast simulated re-annealing. *Math. Comput. Model.* **1989**, *12*, 967–973. [CrossRef]
22. Andrieu, C.; Doucet, A.; Holenstein, R. Particle Markov Chain Monte Carlo Methods. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2010**, *72*, 269–342. [CrossRef]
23. Williams, R.J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **1992**, *8*, 229–256. [CrossRef]
24. Towers, M.; Kwiatkowski, A.; Terry, J.; Balis, J.U.; Cola, G.D.; Deleu, T.; Goulão, M.; Kallinteris, A.; Krimmel, M.; KG, A.; et al. Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv* **2024**, arXiv:2407.17032.